پلتفرم اختصاصی

مهندسی کنترل

https://controlengineers.ir

https://t.me/controlengineers

https://www.instagram.com/controlengineers.ir

Large-Scale Systems

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

controlengineers.ir
@controlengineers

PRENTICE HALL SERIES ON ENVIRONMENTAL AND INTELLIGENT MANUFACTURING SYSTEMS

M. Jamshidi, Editor

Volume 1 - *Robotics and Remote Systems for Hazardous Environments*, Edited by M. Jamshidi and P. J. Eicker, 1993

Volume 2 - *Fuzzy Logic and Control: Software and Hardware Applications*, Edited by M. Jamshidi, N. Vadiee, and T. J. Ross, 1993

Volume 3 - *Artificial Intelligence in Optimal Design and Manufacturing*, Edited by Z. Dong, 1995

Volume 4 - *Soft Computing: Fuzzy Logic, Neural Networks, and Distributed Artificial Intelligence*, Edited by F. Aminzadeh and M. Jamshidi, 1994

Volume 5 - *Nonlinear Control of Robotic Systems for Environmental Waste and Restoration*, Edited by D. M. Dawson, M. M. Bridges, and Z. Qu, 1995

Volume 6 - *Design and Implementation of Intelligent Manufacturing Systems*, Edited by H. R. Parsaei and M. Jamshidi, 1995

Volume 7 - *Advances in Distributed Sensor Technology*, Edited by S. S. Iyengar, L. Prasad, and H. Min, 1995

Volume 8 - *Large-Scale Systems: Modeling, Control, and Fuzzy Logic*, M. Jamshidi, 1996

controlengineers.ir

controlengineers.ir
@controlengineers

Large-Scale Systems:

Modeling, Control and Fuzzy Logic

Mohammad Jamshidi

AT&T Professor and Director
NASA Center for Autonomous Control Engineering
The University of New Mexico
Albuquerque, NM 87131 USA

and

Director de Research Associé
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)
Centre National Recherche Scientifique (CNRS), Toulouse, France

To join a Prentice Hall PTR internet mailing list, point to
http://www.prenhall.com/register

Prentice Hall PTR
Upper Saddle River, New Jersey 07458
http://www.prenhall.com

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Library of Congress Cataloging-in-Publication Data
Jamshidi, Mohammad
Large-scale systems : modeling, control, and fuzzy logic / Mohammad Jamshidi
p. cm. -- (Prentice Hall series on environmental and intelligent manufacturing systems)
Includes bibliographical references and index.
ISBN 0-13-125683-1
1. Large-scale systems--Mathematical models. 2. Intelligent control systems.
3. Fuzzy systems. I. Title. II. Series.
TA168.J35 1996
003'71--dc20 96-26613
CIP

Editorial/production supervision: *Joanne Anzalone*

Manufacturing manager: *Alexis R. Heydt*

Acquisitions editor: *Bernard Goodwin*

Editorial assistant: *Diane Spina*

Cover design: *Anthony Gemmellaro*

Cover design director: *Jerry Votta*

© 1997 by Prentice Hall PTR

Prentice-Hall, Inc.
A Simon & Schuster Company
Upper Saddle River, New Jersey 07458

The publisher offers discounts on this book when ordered in bulk quantities.

For more information, contact:

Corporate Sales Department
Prentice Hall PTR
1 Lake Street
Upper Saddle River, NJ 07458

Phone: 800-382-3419, Fax: 201-236-7141

E-mail: corpsales@prenhall.com

All product names mentioned herein are the trademarks of their respective owners.

All rights reserved. No part of this book may be
reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-125683-1

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

To four of my dearest and best teachers—those who taught me control systems theory and applications

Solon A. Stone (Oregon State University, 1966–67)

José B. Cruz, Jr., Petar V. Kokotović, and William R. Perkins (University of Illinois, 1967–71)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Contents

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

controlengineers.ir
@controlengineers

PREFACE

This book is, in part, a revised version of a previous book by the author: *Large-Scale Systems: Modeling and Control* (Elsevier Publishers, New York, 1983). The main revisions in this volume are as follows:

1) Model reduction methods of frequency domain have been eliminated.
2) Stochastic decentralized control and decentralized estimation have been eliminated.
3) Some near-optimum control schemes of Chapter 6 have been eliminated.
4) Balanced aggregation of model reduction has been added.
5) Fuzzy set, fuzzy logic, and fuzzy control theories have been discussed in about one-third of this text, so much so that the title has been modified to reflect that addition.
6) Fuzzy control architectures, stability criteria, adaptation, and hierarchy have been added.
7) Computer-aided fuzzy logic control and systems simulation have been added
8) An effort has been made to fuse theories of fuzzy systems and large-scale control methods of hierarchy and decentralization, leading to such topics as sensory fusion, rule-based reduction, and fuzzy control of complex systems.

Many real-life problems facing nations of the world are brought forth by present-day technology and by societal and environmental processes which are highly complex, "large" in dimension, and stochastic in nature. The notion of "large-scale" is a subjective one in that one may ask: How large is large? Many viewpoints have been presented on this issue. One viewpoint has been that a system is considered large in scale if it can be decoupled or partitioned into a number of interconnected subsystems for either computational or practical reasons. Another viewpoint considers "large-scale systems" to be simply those whose dimensions are so large that conventional techniques of modeling, analysis, control design, and

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

optimization fail to give reasonable solutions with reasonable computational efforts.

Needless to say, many real problems are considered to be "large-scale" by nature and not by choice. Two important attributes of large-scale systems are (i) they often represent complex, real-life systems and (ii) their hierarchical (multilevel) and decentralized information structures depict systems dealing with societal, business, and management organizations, the economy, the environment, data networks, electric power, transportation, information systems, aerospace (including space structures), water resources, and, last but not least, energy. Such systems which are used in support of human life are complex in nature. As a result of these important properties and potential applications, several researchers have paid a great deal of attention to various facets of large-scale systems such as modeling, model reduction, control, stability, controllability, observability, optimization, stabilization, and the role of artificial intelligence. These concepts have been applied to various problems and have helped with the creation of different notions of systems analysis, design, control, and optimization.

Recent advances of *soft computing* such as fuzzy logic, neural networks, and genetic (evolutionary) algorithms have also led us to new approaches for control of complex systems. These techniques, which form building blocks of what is commonly termed *intelligent control*, as in this text, are becoming very appropriate for the control of complex systems. The notions of fuzzy set, logic, and control are covered in some detail. However, due to a shortage of space, we are not able to cover neural networks and genetic algorithms in much detail.

The purpose of this book is to present a balanced treatment of the large-scale systems by featuring past, present, and potential trends of the subject as well as the role that fuzzy logic can play in such systems. An attempt is made to introduce the fundamental and more-or-less settled issues on the subject. The general theme throughout the book is the algorithmic and computer-assisted approach. Most of the theoretical concepts are stated with proofs, followed by an algorithm showing how to use the results. One or more numerical examples then illustrate the theoretical concept. A great majority of the numerical examples in the book were solved on personal computers using computer-aided design packages which were either developed by the author's team of researchers or by other colleagues. Because most problems of the book (denoted by a computer terminal in the problem sections) require the use of a computer, the interested reader may use the postcards at the end of the book or write directly to the author for information on these. A notable exception is Chapter 6, where more research-oriented topics are presented.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The book will address three main issues: (1) modeling and model reduction (2) fundamental concepts in optimum, near-optimum control, and system properties such as stability, controllability, observability, pole assignment, and hierarchical and decentralized control, and finally (3) the role of fuzzy systems in the model identification and control of large-scale systems.

Chapter 1 presents an introduction to large-scale systems. Important classes such as hierarchical or multilevel control systems and decentralized control systems are presented. An introduction to artificial intelligence notions of fuzzy logic and neural networks are also given here.

Models, model reduction, and model identification of large-scale systems are treated in Chapter 2. Time-domain modeling schemes, such as aggregation, perturbation, balance aggregation, modeling via system identification methods and via fuzzy logic are described here.

The structural properties (stability, controllability, and observability) of large-scale systems are considered in Chapter 3. Both the Lyapunov and Input-Output stability approaches have been treated on an equal basis. In addition, the related notion of "connective" stability is introduced, and a comparative discussion on the main stability approaches is given.

Chapter 4 considers both open- and closed-loop hierarchical control systems in continuous and discrete forms. In Chapter 5, decentralized control represents the main theme. Decentralized stabilization and adaptive decentralized control are treated. Chapter 6 is concerned with the applications of optimum control theory to large-scale systems methods that have been developed in Chapters 2 to 5. A detailed discussion on the degradation of the optimal performance measure is presented at the end of this chapter.

Chapters 7, 8, and Appendices A and B are dedicated to fuzzy logic in the broad sense. Fuzzy logic, in this context, refers to fuzzy set theory and fuzzy sets which include fuzzy mathematics, fuzzy operations research, fuzzy control, etc. Chapter 7 introduces the reader to fuzzy control structures and fuzzy control stability. Chapter 8 details the latest advances of fuzzy control systems, adaptation, and hierarchy. Important subjects such as fuzzy rule-base reduction through sensory fusion and rule hierarchy are discussed. This chapter and the book end with a set of new open problems for the infusion of fuzzy logic in large-scale complex systems. Here such problems as decentralized fuzzy control, hierarchical fuzzy control, etc. have been introduced. Some of these control paradigms are left as open research problems for the readers.

Finally, Appendix A presents a brief introduction into fuzzy sets and fuzzy logic as a logical system. The fuzzy logic software environment FULDEK™ (FUzzy Logic DEvelopment Kit) is described in Appendix B.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The author is indebted to many people for their various contributions. Foremost, I would like to thank Solon Stone (Oregon State University), Petar Kokotović (University of California at Santa Barbara), Joe Cruz (Ohio State University), and Bill Perkins (University of Illinois at Champaign-Urbana), who have been instrumental in my education in control and systems engineering. I would like to thank my dear friend and colleague Peter Dorato at the University of New Mexico and my personal role models Lotfi Zadeh (University of California, Berkeley) and Faz Reza (Concordia University, Canada) who have always inspired me and have always been inspiring to me in my career. I am especially grateful to the many researchers and workers in large-scale systems and fuzzy logic without whose work this book could not have become a reality. I thank, in particular, Professor Andy Sage (George Mason University) and Madan Singh (University of Manchester Institute of Technology, U.K.) who have been supportive of my large-scale systems research throughout years. Many individuals at the University of New Mexico have been very supportive of my work. I thank former EECE Department Chair Professor Russell Seacat and former EECE Department Chair and Associate Professor for Research Nasir Ahmed for continuous support of the author's activities. I thank Dr. Yvonne Freeman, former Associate Administrator for NASA, and Dr. Darwish Al-Gobaisi of the Encyclopedia of Life Support Systems for believing in my approach to research and educational programs, and Mark Dreier, the author of the FULDEK fuzzy logic software environment, for his always delightful approach to any problem or relationship.

A major part of this book was revised and written while the author was at LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) of CNRS (Centre National de Recherche Scientifique) in Toulouse, France. While at LAAS, I benefited immensely from professional interactions with many colleagues. I thank Professor André Titli, Dr. Jacques Bernussou, Dr. Francois Dufour, and Professor Allen Costes, Director of LAAS, for stimulating interactions.

I am particularly indebted to many of my present and former students, especially Dr. Nader Vadiee, Kishan Kumbla, Mohammad Akbarzadeh, Eddie Tunstel, Jr., Ali Asgharzadeh, Dr. John Santiago (U.S. Air Force), and Roger Salters (University of Denver), for many inspiring discussions and solid research throughout the years. I wish to express special thanks to my former doctoral student and present colleague Young-Tae Kim (Professor of Dongguk University, Korea, and visiting Professor at the University of New Mexico) for his careful proofreading of the manuscript.

I would like to acknowledge the typing and editorial assistance of many who helped prepare the manuscript. In particular, I would like to thank Virginia and Bill Henning for their excellent typesetting of the manuscript

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and patience with all its revisions and alterations and Bernard Goodwin, publisher, and the production staff of Prentice Hall PTR for their patience and cooperation. Last, but by no means least, I would like to thank my loving and sacrificing family—my wife Jila, my daughter Ava, and my son Nima for putting up with me for so many years and providing me with love and energy to move on.

M. Jamshidi
Toulouse, France and Albuquerque, NM, USA

controlengineers.ir

controlengineers.ir

controlengineers.ir
@controlengineers

Chapter 1

Introduction to Large-Scale Systems

1.1 Historical Background

A great number of today's problems are brought about by present-day technology and societal and environmental processes which are highly complex, "large" in dimension, and uncertain by nature. The notion of "large-scale" is a very subjective one in that one may ask: How large is *large*? There has been no accepted definition for what constitutes a "large-scale system." Many viewpoints have been presented on this issue. One viewpoint has been that a system is considered large-scale if it can be decoupled or partitioned into a number of interconnected subsystems or "small-scale" systems for either computational or practical reasons (Ho and Mitter, 1976; Jamshidi, 1983). Figure 1.1 depicts this viewpoint. Another viewpoint is that a system is large-scale when its dimensions are so large that conventional techniques of modeling, analysis, control, design, and computation fail to give reasonable solutions with reasonable computational efforts. In other words, a system is large when it requires more than one controller (Mahmoud, 1977).

Since the early 1950s, when classical control theory was being established, engineers have devised several procedures, both within the classical and modern control contexts, which analyze or design a given system. These procedures can be summarized as follows:

1. Modeling procedures which consist of differential equations, input-output transfer functions, and state-space formulations.
2. Behavioral procedures of systems such as controllability, observability, and stability tests, and the application of such criteria as Routh-Hurwitz, Nyquist, Lyapunov's second method, circle criterion, etc.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

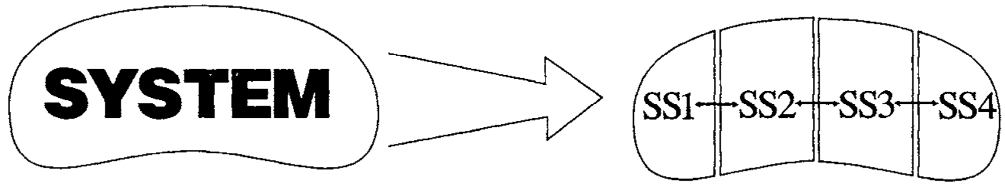


Figure 1.1 A definition of a large-scale system based on notion of decomposition.

3. Control procedures such as series compensation, pole placement, optimal control, robust control, intelligent control, etc.

The underlying assumption for all such control and system procedures has been "centrality" (Sandell *et al.*, 1978; Jamshidi, 1983), i.e., all the calculations and measurements based upon system information (be it *a priori* or sensor information) and the information itself are localized at a given center, very often a geographical position. A prime example of a centralized system is a computer-controlled experimental testbed physically located in a laboratory setting.

A notable characteristic of most large-scale systems is that centrality fails to hold due to either the lack of centralized computing capability or centralized information. Needless to say, many real problems are considered large-scale by nature and not by choice. The important points regarding large-scale systems are that they often model real-life systems and that their hierarchical (multilevel) and decentralized structures depict systems dealing with society, business, management, the economy, the environment, energy, data networks, aeronautical systems, power networks, space structures, transportation, aerospace, water resources, ecology, robotic systems, and flexible manufacturing systems, to name a few. Some of these systems are often separated geographically, and their treatment requires consideration of not only economic costs, as is common in centralized systems, but also such important issues as reliability of communication links, value of information, environmental consciousness, machine intelligence quotient (MIQ), etc. It is for the decentralized and hierarchical control properties and potential applications of such exciting areas as intelligent large-scale systems in which *fuzzy logic* and *neural networks* are incorporated within the control architecture that many researchers have devoted a great deal of effort to large-scale intelligent systems in recent years.

controlengineers.ir

controlengineers.ir
@controlengineers

1.2 Hierarchical Structures

One of the earlier attempts in dealing with large-scale systems was to "decompose" a given system into a number of subsystems for computational efficiency and design simplification. The idea of "decomposition" was first treated theoretically in mathematical programming by Dantzig and Wolfe (1960) by treating large linear programming problems possessing special structures. The coefficient matrices of such large linear programs often have relatively few nonzero elements, i.e., they are sparse matrices. There are two basic approaches for dealing with such problems: "coupled" and "decoupled." The coupled approach keeps the problem's structure intact and takes advantage of the structure to perform efficient computations. The "compact basis triangularization" and "generalized upper bounding" are two such methods (Ho and Mitter, 1976). The "decoupled" approach divides the original system into a number of subsystems involving certain values of parameters. Each subsystem is solved independently for a fixed value of the so-called "decoupling" parameter, whose value is subsequently adjusted by a coordinator in an appropriate fashion so that the subsystems resolve their problems and the solution to the original system is obtained.

This approach, sometimes termed as the "multilevel" or "hierarchical" approach, is shown in Figure 1.2. At the first level, N subsystems of the original large-scale system are shown. At the second level a coordinator receives the local solutions of the N subsystems, s_i, $i = 1, 2, \dots, N$, and then provides a new set of " interaction" parameters, a_i, $i = 1, 2, \dots, N$. The goal of the coordinator is to arrange the activities of the subsystems to provide a feasible solution to the overall system. This exchange of solution (by the substystems) and coordination (interaction) vector (by the coordinator) will continue until convergence has been achieved. Such a solution,

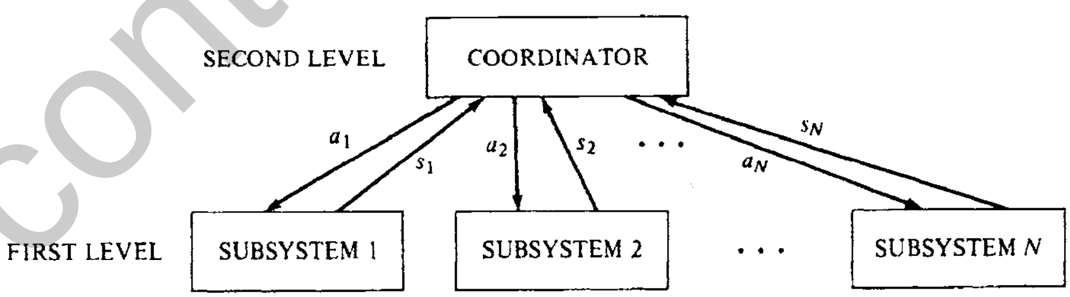


Figure 1.2 Schematic of a two-level hierarchical system.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

as it will be seen in Chapter 4, is both feasible and optimum, in the sense of minimizing a cost function of the overall system.

The original success of the hierarchical multilevel approach has been primarily in social systems (Mayne, 1976) and water resources systems (Haimes, 1977). Today, hierarchical control has been used for many engineering problems such as robotics, manufacturing, and the like (Jamshidi *et al.*, 1992, 1994). The multilevel structure, according to Mesarovic *et al.* (1970), has five advantages: (i) the decomposition of systems with fixed designs at one level and coordination at another is often the only alternative available; (ii) systems are commonly described only on a stratified basis; (iii) available decision units have limited capabilities, hence the problem is formulated in a multilayer hierarchy of subproblems; (iv) the overall system resources are better utilized through this structure; and (v) there will be an increase in system reliability and flexibility. There has been some disagreement among system and control specialists regarding these points. For example, Varaiya (1972) has mentioned that the first three advantages are a matter of opinion, and there is no evidence in justifying the other two. One shortcoming of most multilevel structures is that they are inherently open-loop structures, although closed-loop structures have been proposed (Singh 1980). Detailed discussion on the hierarchical (multilevel) method will be given in Chapter 4.

1.3 Decentralized Control

Most large-scale systems are characterized by a great multiplicity of measured outputs and inputs. For example, an electric power system has several control substations, each being responsible for the operation of a portion of the overall system. This situation arising in a control system design is often referred to as *decentralization*. The designer for such systems determines a structure for control which assigns system inputs to a given set of local controllers (stations), which observe only local system outputs. In other words, this approach, called decentralized control, attempts to avoid difficulties in data gathering, storage requirements, computer program debuggings, and geographical separation of system components. A preliminary comparison between decentralized and hierarchical control can be given here. In hierarchical control, a decomposition in system structure will lead to computational efficiency. In decentralized control, on the other hand, a decomposition takes place with the system's output information leading to simpler controller structures and computational efficiency. Figure 1.3 shows a two-controller decentralized system. The basic characteristic of any decentralized system is that the transfer of information from

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

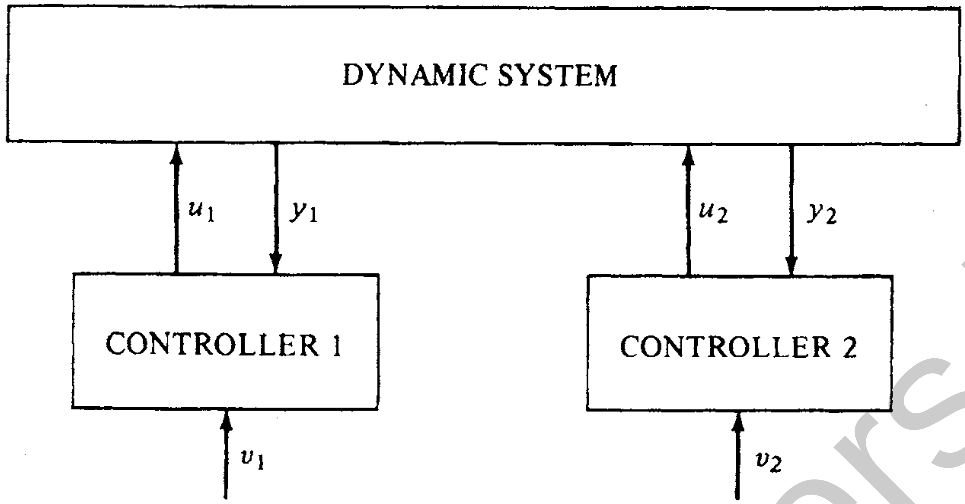


Figure 1.3 A two-controller decentralized system.

one group of sensors or actuators to others is quite restricted. For example, in the system of Figure 1.3, only the output y_1, and external input v_1, are used to find the control u_1, and likewise the control u_2 is obtained through only the output y_2 and external input v_2.

The determination of control signals u_1, and u_2 based on the output signals y_1, and y_2, respectively, is nothing but two independent output feedback problems which can be used for stabilization or pole placement purposes. It is therefore clear that the decentralized control scheme is of feedback form, indicating that this method is very useful for large-scale linear systems. This is a clear distinction from the hierarchical control scheme, which was mainly intended to be an open-loop structure. Further discussion of decentralized control and its applications for stabilization, robust controllers, etc., will be considered in Chapters 5 and 6.

In this and the previous two sections the concept of a large-scale system and two basic hierarchical and decentralized control structures were briefly introduced. Although there is no universal definition of a large-scale system, it is commonly accepted that such systems possess the following characteristics (Jamshidi, 1983):

1. Large-scale systems are often controlled by more than one controller or decision maker involving "decentralized" computations.
2. The controllers have different but correlated "information" available to them, possibly at different times.
3. Large-scale systems can also be controlled by local controllers at one level whose control actions are being coordinated at another level in a "hierarchical" (multilevel) structure.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

4. Large-scale systems are usually represented by imprecise "aggregate" models.
5. Controllers may operate in a group as a "team" or in a "conflicting" manner with single- or multiple-objective or even conflicting-objective functions.
6. Large-scale systems may be satisfactorily optimized by means of suboptimal or near-optimum controls, sometimes termed a "satisfying" strategy.

1.4 Artificial Intelligence

Today's complex large-scale systems' difficulties begin with modeling and propagate on the simulation, analysis, and design. In one school of thought, a system is considered large in scale if traditional tools of modeling, analysis, design, etc., would not be able to handle such systems in reasonable time or even with the most powerful computers.

In recent years, attempts have been made to introduce a new tool into the modeling and control of systems—large or small in scale. This new tool is artificial intelligence (AI), which can be defined as the science of automating intelligent behavior (Luger and Stubblefield, 1989). Within the field of AI, three important approaches are worth mentioning. These are expert systems, neural networks, and fuzzy logic. An *expert system* is constructed (often in the form of a computer code) by obtaining the domain-specific knowledge (e.g., knowledge of medicine from a medical doctor) and coding it into a form that a computer may apply to similar problems. In traditional expert systems, uncertainty is handled by probability theory and reasoning is often done by probabilistic methods—*probabilistic reasoning* (PR). *Neural networks*, on the other hand, represent models of the human brain. In neural networks the emphasis is on the brain's learning process, while traditional AI expert systems try to model the physical aspects of the brain (Newell and Simon, 1976). *Fuzzy logic*, which stems from fuzzy set theory (Zadeh, 1965), deals with the vagueness, imprecision, and linguistic approach to human reasoning. Fuzzy logic deals with a measure of vagueness as predicate logic deals with a measure of randomness.

1.4.1 Neural Networks

Neural networks (NN) are developed to mimic the flexibility and power of the human brain by artificial means. A NN typically consists of a mesh of

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

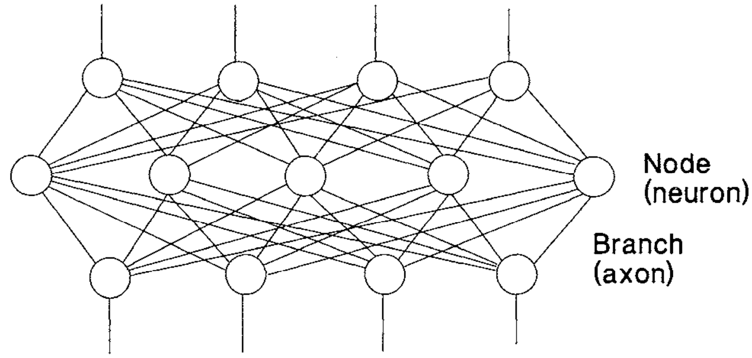


Figure 1.4 A typical neural network.

nodes and branches connected together, as shown in Figure 1.4. The main processing element of every NN is an artificial neuron, or simply a neuron, or a node as shown in Figure 1.4. One early model of a simple NN was called a *perceptron* by Rosenblatt (1958). Figure 1.5 shows a multilayer perceptron along with a single neuron. Associated with each perceptron are $n + 1$ input branches with n unknown weights (sometimes called synaptic weights), a threshold (or bias weight) input w_0, and an *activation* (nonlinear) function such as a hard limiter or a so-called sigmoid function. Thus, the output of the perceptron represents a nonlinear transformation of the inner product of a weighting vector $w = (w_0, w_1, w_2, \ldots, w_n)$ and the input vector $x = (x_1, x_2, \ldots, x_n)$, i.e.,

$$u = f(y),\ y = w^T x \tag{1.1}$$

In (1.1), the threshold input has been incorporated as a known input of value one (Hush and Horne, 1990).

One of the more celebrated applications of neural networks has been classification and pattern recognition of data.

1.4.2 Fuzzy Logic

Among the many new technologies based on AI, fuzzy logic is now perhaps the most popular area, judging by billions of dollars worth of sales and close to 2,000 patents issued in Japan alone since the announcement of the first fuzzy chips in 1987. Fuzzy logic, as discussed earlier, stems from the notion of fuzzy sets. A fuzzy set is one in which the transition

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

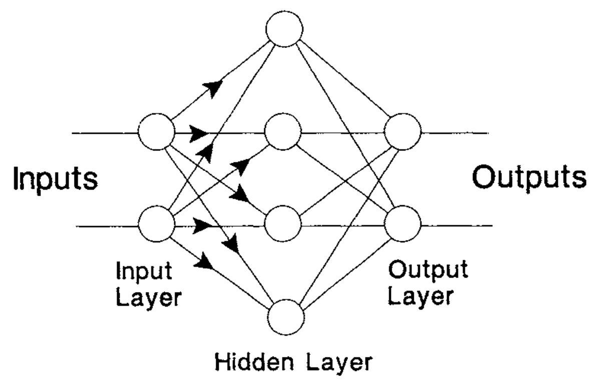


Figure 1.5a A three-layer perceptron.

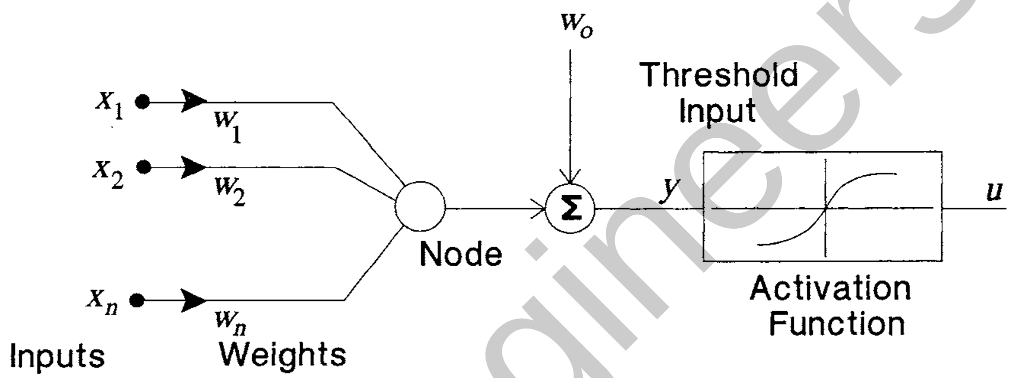


Figure 1.5b A perceptron model.

between membership and nonmembership is gradual and not abrupt as in a classical set. A fuzzy set, on the other hand, constitutes a mathematical expression for the lack of precision in a quantitative fashion by introducing a set membership function that can take on real values between 0 and 1. Assume that a definition of "tall" persons is measured at 6′ 0″ or higher. Figure 1.6 shows the set "tall" from the point of view of a "classical" and "fuzzy" set. The membership function gives a degree or grade of membership within the set. This membership function, denoted by $\mu_A(x)$, maps the elements x of the universe X into a numerical value {0 or 1} for a classical set or the closed interval [0,1], i.e.,

$$\begin{aligned} &\mu_A(x):\ X \to \{0,1\} \ldots \textit{classical} \\ &\mu_A(x):\ X \to [0,1] \ldots \textit{fuzzy} \end{aligned} \tag{1.2}$$

One of the most dominant applications of fuzzy logic has been fuzzy control. In a fuzzy controller, the desired behavior of a system's response

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

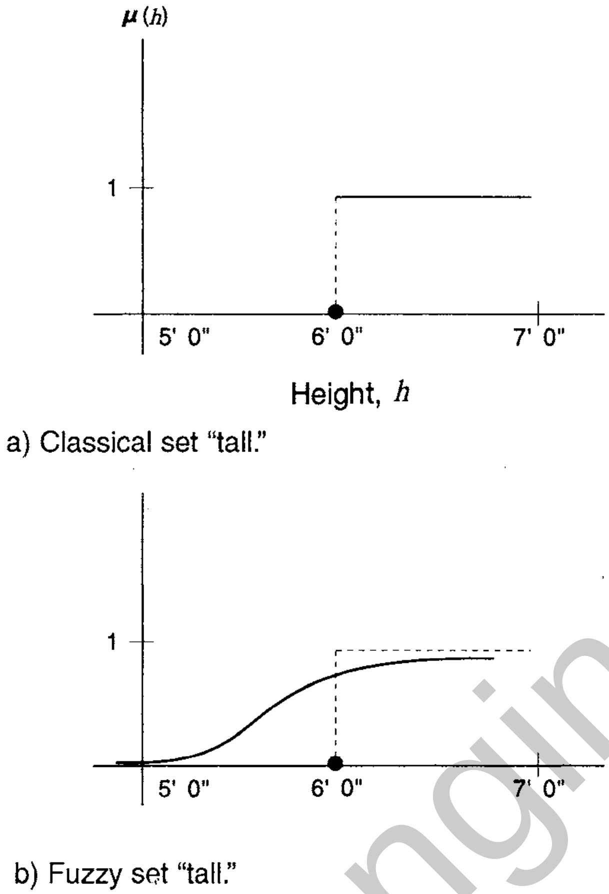


Figure 1.6 Set membership functions for linguistic variable "tall."

controlengineers.ir

is depicted in a number of behavioral implications through fuzzy IF-THEN rules. A typical fuzzy rule has one or more *antecedents* (IF parts) and one or more *consequents* (THEN parts) which are membership functions. Consequents from different rules are numerically combined (typically unioned via a MAX operation) and are than collapsed (typically taking the weighted average or *anteroid* of the combined distribution) to yield a single real number (binary) output. This process is called *defuzzification* in fuzzy control terminology (Jamshidi *et al.*, 1993). Within the framework of a fuzzy expert system, like standard expert systems, typical rules can be the result of, among other schemes, a human expert's knowledge, e.g.,

> IF the temperature is "Hot" and the pressure is "Low"
> THEN increase value position to "Large" amount.

In this rule, *Hot*, *Low*, and *Large* are fuzzy variables. These fuzzy variables or sets are the results of compressing or reducing a partition of nu-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

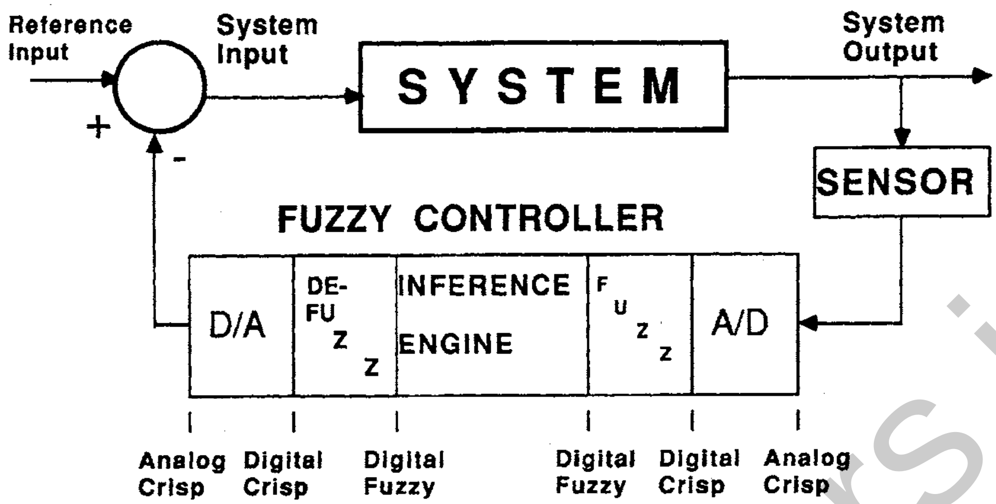


Figure 1.7 The architecture of a fuzzy controller.

merical values of physical variables like temperature or pressure into a linguistic value. Such a process is sometimes called "defuzzification." Figure 1.7 shows a typical fuzzy control architecture.

Appendix A provides a brief overview of fuzzy sets and fuzzy logic while fuzzy control is described in Chapters 7 and 8.

1.5 Computer-Aided Approach

Another important tool to handle systems of any scale is the digital computer. It was only a few years ago that control system designers did not have any significant computer-aided control system design (CACSD) software. In the past decade or so over 20 CACSD packages have been created either at academic institutions or commercial establishments (Jamshidi *et al.*, 1992). As computers have become an indispensable part of the design process, new control philosophies, such as hierarchical and decentralized control within large-scale systems, have provided new challenges for CACSD programs.

Some of the few CAD (Computer Aided Design) programs which are solely dedicated to large-scale systems have been developed by the author's research team. These packages are: LSSPAK©, a Large Scale Systems PAcKage, and LSSTB©, a MATLAB™ -based large-scale systems toolbox. These two packages are available to the reader, who can order a copy by sending in the self-addressed card at the back of the book. These programs

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

will be used extensively by the author in almost all chapters.

1.6 Scope

Since the subject of large-scale systems is still growing, it is difficult and pointless to attempt to cite every reference. On the other hand, if we were to confine the discussion to one or two subtopics and use only immediate references, it would hardly reflect the importance of the subject. In this text an attempt is made to primarily consider modeling, control, and fuzzy logic applications of large-scale systems. Important topics, such as stability, controllability, and observability, are discussed briefly. Also, an attempt is made in this text to introduce an important area of artificial intelligence, fuzzy logic, in the theory of large-scale systems. Most of our discussions are focused on large-scale linear, continuous-time, stationary, and deterministic systems. However, other classes of systems, such as discrete-time, time-delay, and nonlinear systems, are also considered. Among control strategies, the main focus has been on hierarchical (multilevel), decentralized, and intelligent controls. The term *intelligent control* in this text refers to the integration of artificial intelligence (expert systems, neural networks, fuzzy logic, genetic algorithms, probabilistic reasoning, etc.), operations research, and control theory. *Fuzzy control* refers to an integration of fuzzy logic and control theory. On the modeling side, aggregation and perturbation (regular and singular) are among the primary topics discussed. Other topics such as identification and estimation as well as large-scale systems control and modeling schemes, such as the Stackelberg approach (Cruz, 1978), component connection model (Saeks and DeCarlo, 1981), multilayer and multiechelon structures, and Nash games, are either considered very briefly or have not been discussed. The emphasis has been on the use of the subject matter in the classroom for students of large-scale systems in a simple and understandable language. Most important theorems are proved, and many easily implementable algorithms support the theory; ample numerical and CAD examples demonstrate their use.

Problems

1.1. Develop a multilevel (hierarchical) structure for a business organization with a board of directors, a chairman of the board, a president, three vice presidents (marketing-sales, research, technology), etc.

1.2. Explain whether the concept of "centrality" holds for each of the

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

following systems. State your reasons in a sentence.

a. An autopilot aircraft control system.
b. A three-synchronous machine power system.
c. A computer system involving a host computer and five terminals.
d. A home heating system.
e. A radar control system.

1.3. The allocation of water resources in any state is commonly the responsibility of the state engineer's office which checks for overall system feasibility by overseeing municipalities and conservancy districts, which work independently and report their programs to the state engineer's office. Consider a two-municipality and three-district state and draw a block diagram representing the water resources system.

1.4. A system has two inputs and one output. Draw a 3-layer perceptron which would identify the model of the system.

1.5. Draw a triangular fuzzy set membership function for each of the following linguistic variables: (a) short, (b) not tall, (c) almost 5 volts DC, and (d) approximately 5 p.m.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 2

Large-Scale Systems Modeling

2.1 Introduction

Scientists and engineers are often confronted with the analysis, design, and synthesis of real-life problems. The first step in such studies is the development of a *mathematical model* which may be a substitute for the real problem.

In any modeling task, two often conflicting factors prevail—"simplicity" and "accuracy." On one hand, if a system model is oversimplified, presumably for computational effectiveness, incorrect conclusions may be drawn from it in representing an actual system. On the other hand, a highly detailed model would lead to many unnecessary complications and should a feasible solution be attainable, the extent of resulting details may become so vast that further investigations on the system behavior would become impossible with questionable practical values (Sage, 1977; Šiljak, 1978; Jamshidi, 1983). Clearly a mechanism by which a compromise can be made between a complex, more accurate model and a simple, less accurate model is needed. Such a mechanism is not a simple undertaking. The key to a valid modeling philosophy is to set forth the following outline (Brogan, 1991):

1. The *purpose* of the model must be clearly defined; no single model can be appropriate for all purposes.
2. The *system's boundary* separating the system and the outside world must be defined.
3. A *structural relationship* among different system components which would best represent desired or observed effects must be defined.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

4. Based on the physical structure of the model, a set of *system variables* of interest must be defined. If a quantity of important significance cannot be labeled, step 3 must be modified accordingly.
5. Mathematical descriptions of each system component, sometimes called *elemental equations*, should be written down.
6. After the mathematical description of each system component is complete, they are related through a set of physical laws of *conservation* (or *continuity*) and compatibility, such as Newton's, Kirchhoff's, or D'Alembert's.
7. Elemental, continuity, and compatibility equations should be *manipulated*, and the mathematical *format* of the *model* should be *finalized*.
8. The last step to a successful modeling is the analysis of the model and its comparison with real situations.

Should there be any significant discrepancies, steps 1–8 must be reexamined and modified accordingly.

The above steps for a system model development emphasize the fact that a great deal of experience is needed for a sound compromise between accuracy and simplicity. The common practice has been to work with simple and less accurate models. There are two different motivations for this practice: (i) the reduction of computational burden for system simulation, analysis, and design; and (ii) the simplification of control structures resulting from a simplified model. It should be emphasized that these motivations are distinct in the sense that one does not necessarily imply the other. This distinction has been demonstrated by Gelb (1974) for the reduced-order Kalman filter design. It has been shown that the determination of a Kalman filter's error covariance matrix requires the solution of n^2 equations, n being the system order. The error covariance matrix for a reduced-order model of lth dimension would require the solution of $(n + l)^2$ equations instead. This example leads one to conclude that simplified structure and computational reduction are two separate issues in system modeling which are not necessarily compatible (Sandell *et al.*, 1978; Jamshidi, 1983).

Thus far we have outlined the necessary steps for a system model with "centralized" structure and have indicated that "reduced computation" and "simplified structures" are two characteristics every system analyst would wish to attribute to models he or she would be dealing with. These desirable properties are of even more concern for the decentralized control of large-scale systems, introduced briefly in Chapter 1. This concern may very well be more subjective for large-scale systems than regular systems, mainly due to the fact that the state of the art in large-scale systems calls for more desirable structures.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Until recently there have been only two schemes for modeling large-scale systems, and they have been around for quite sometime—*aggregation* and *perturbation*. These schemes have been carried on from economic theory and mathematics, respectively, to systems modeling, analysis, and control. Other large-scale systems structures and strategies, such as the hierarchical (Mesarovic *et al.*, 1970), decentralized control (Sandell *et al.*, 1978), and Stackelberg (leader-follower) approaches (Cruz, 1976), are not considered modeling schemes for the sake of discussion here. They are treated as system control strategies in Chapters 4 and 5.

An *aggregate model* of a system is described by a "coarser" set of state variables. The underlying reason for aggregating a system model is to be able to retain the key qualitative properties of the system, such as stability, which is viewed by Siljak (1978) as a natural process through the second method of Lyapunov. In other words, the stability of a system described by several state variables is entirely represented by a single variable—the Lyapunov function. Aggregated models are common processes for many econometric studies. For example, behind an aggregated variable, say, the consumer price index, numerous economic variables and parameters may be involved.

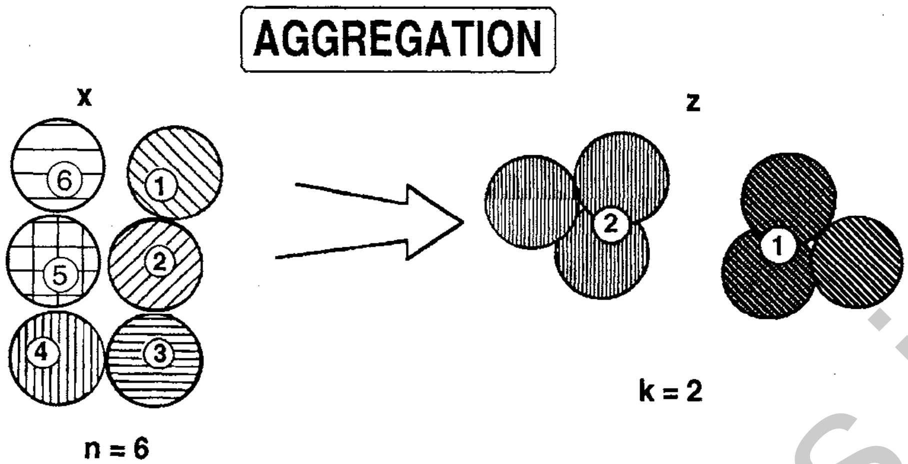
Figure 2.1 presents a pictorial presentation of the aggregation process. The system on the left is described by six variables (circles), and the system on the right represents an aggregated model for it, where two variables now describe the system. Variable No. 1, called z_1, is an average of the full model's first three variables (x_1, x_2, x_3), while the second aggregated variable z_2 is an average of the second three variables (x_4, x_5, x_6).

The other scheme for large-scale systems modeling has been perturbation, which is based on ignoring certain interactions of the dynamic or structural nature in a system. Here again the benefits received from reduced computations must not be at the expense of key system properties. Although both perturbation and aggregation schemes tend to provide reduction in computations and perhaps simplification in structure, there has been no hard evidence that they are the most desirable for large-scale systems.

Figure 2.2 provides a pictorial presentation of perturbation of a singular type. The fast variable of the system can be approximated by an auxiliary or quasi steady-state variable, and through a concept called "boundary layer" (see Section 2.3.2) the missing initial information of this variable can be accounted for. This would lead to a reduced model based on the slow variable.

An effort along these lines is perhaps the *descriptive variable* scheme for large-scale systems modeling due to Larson, Luenberger, and their associates (Luenberger, 1977, 1978; Stengel *et al.*, 1979). The fundamental

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



$$z = Cx = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} x$$

Figure 2.1 A pictorial presentation of the aggregation process.

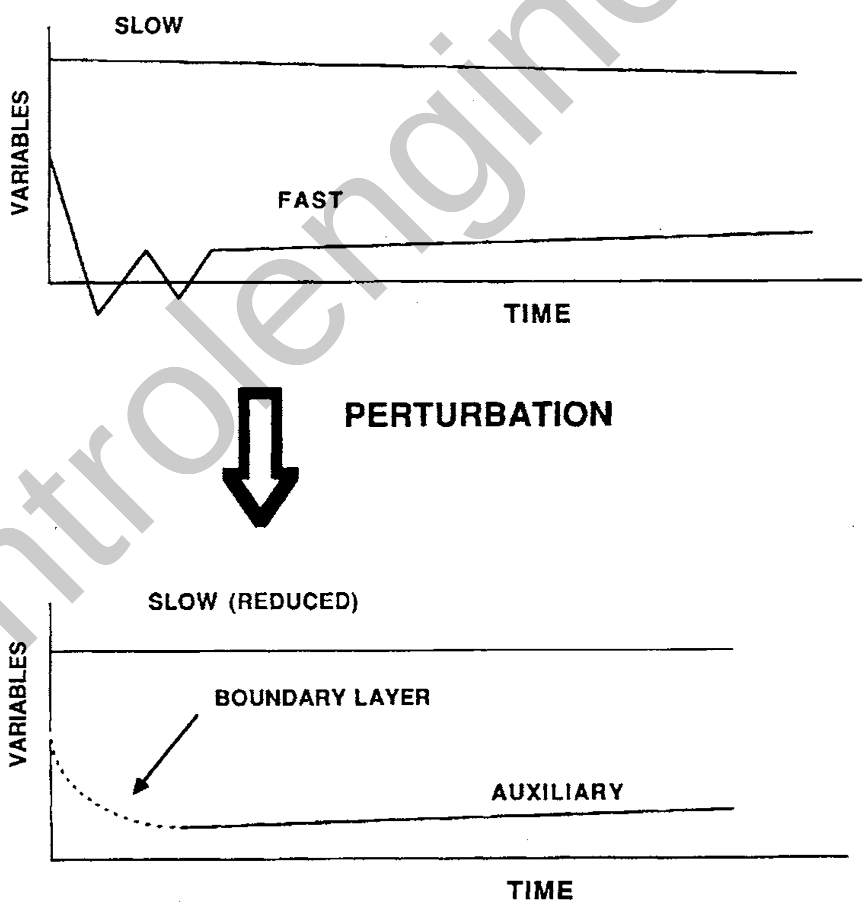


Figure 2.2. A pictorial presentation of the perturbation process.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

issue in this modeling philosophy is that the accuracy of a given large-scale system model is most likely preserved if the system is represented by the actual physical or economical *variables* which *describe* the operation of the system; hence the name "descriptive variable." Due to the lack of space and extensive need for the coverage of this topic, the descriptor variable approach is not covered here. A brief introduction to this approach can be found in Jamshidi (1983).

This chapter is devoted to detailed examinations of "aggregation" and "perturbation," methods viewed as modeling alternatives for large-scale systems. Portions of this chapter and the next are based on other works by the author (Jamshidi, 1989, Jamshidi *et al.*, 1992). Figure 2.3 provides a summary of all time-domain and frequency-domain methods of large-scale systems model simplification schemes.

In the quest for models of systems—large or small—other approaches and tools are still available to find models numerically. These are based on the notion of *system identification*. System identification refers to the following problem: given input and output data of the operation of a system with an unknown model, what would be the best mathematical model which can fit the given data? Toward this goal, computational software such as MATLAB's System Identification toolbox (Ljung, 1988) as well as the use of AI approaches such as neural networks and fuzzy logic also will be used in this chapter and textbook.

2.2 Aggregation Methods

Aggregation has long been a technique for analyzing static economic models. The treatment of aggregation in modern time is probably due to Malinvaud (1956), whose formulation is shown in Figure 2.4a (Aoki, 1978). In this diagram, X, Y, Z, and V are topological (or vector) spaces and f represents a linear continuous map between the exogenous variable $x \in X$ and endogenous variable $y \in Y$. The aggregation procedures h: $X \rightarrow Z$, and g: $Y \rightarrow V$, lead to aggregated variables $z \in Z$ and $v \in V$. The map k: $Z \rightarrow V$ is to represent a simplified or an aggregated model. The aggregation is said to be "perfect" when k is chosen such that the relation

$$gf(x) = kh(x) \tag{2.2.1}$$

holds for all $x \in X$. The notion of perfect aggregation is an idealization at best, and in practice it is approximated through two alternative procedures according to econometricians (Chipman, 1976). These are (i) to impose some restrictions on f, g, and h while leaving $\mathbf{X}$ unrestricted and (ii) to require (2.2.1) to hold on X_0 some subset of X.

controlengineers.ir

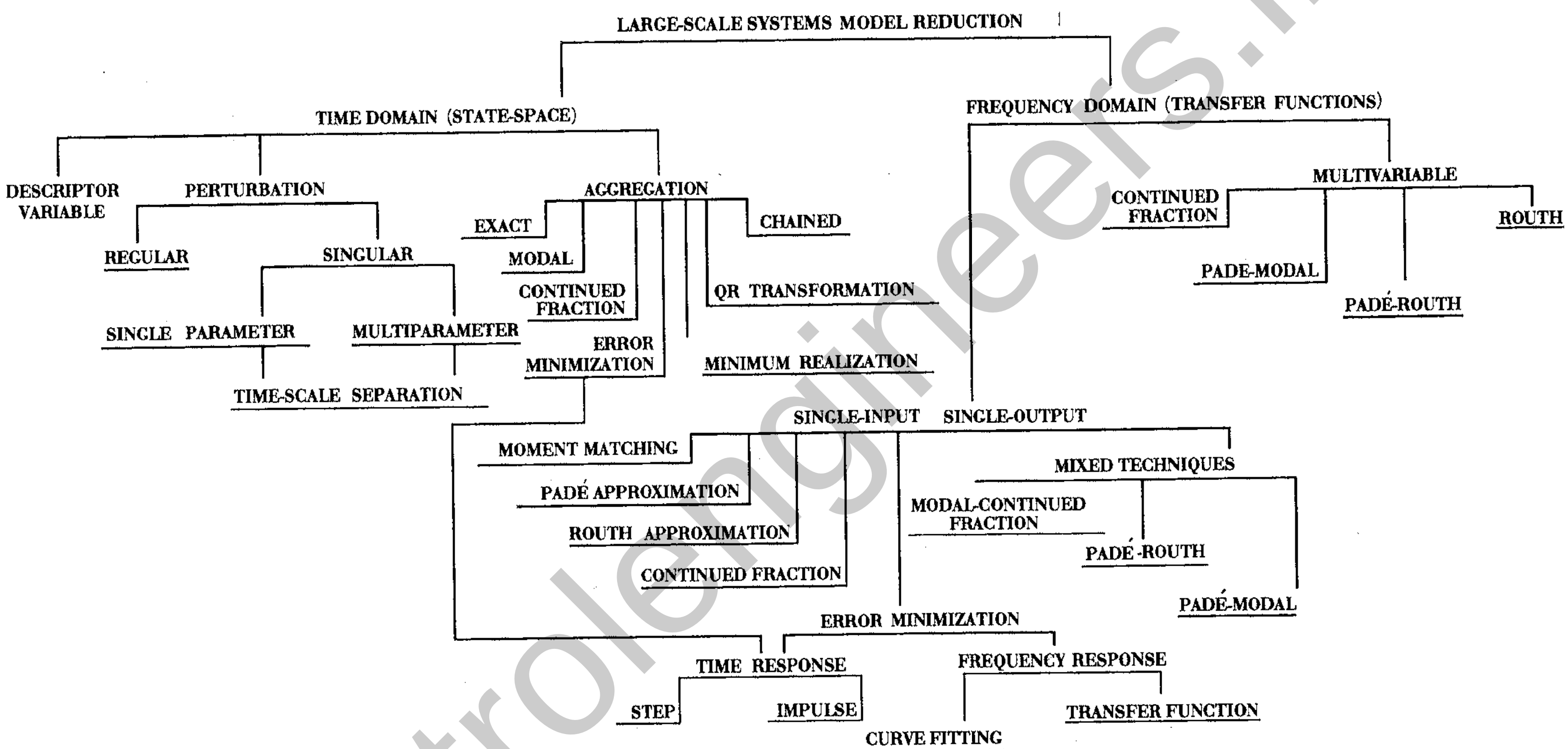


Figure 2.3 Large-scale systems model reduction methods.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$
\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "h"'] & Y \arrow[d, "g"] \\
Z \arrow[r, "k"'] & V
\end{tikzcd}
$$

(a)

$$
\begin{tikzcd}
 & \boldsymbol{R}^n \arrow[r, "A"] \arrow[dd, "C"'] & \boldsymbol{R}^n \arrow[dr, "D"] \arrow[dd, "C"] & \\
\boldsymbol{R}^m \arrow[ur, "B"] \arrow[dr, "G"'] & & & \boldsymbol{R}^r \\
 & \boldsymbol{R}^l \arrow[r, "F"'] & \boldsymbol{R}^l \arrow[ur, "K"'] &
\end{tikzcd}
$$

(b)

Figure 2.4 A pictorial representation of aggregation: **(a)** a static system, and **(b)** a dynamic linear system.

In this section, aggregation of large-scale linear time-invariant systems is introduced and it is shown that it is not merely a model reduction scheme but, more importantly, a conceptual basis for other approximation techniques, including the modal aggregation (Davison, 1966, 1968) which retains the dominant modes of the original system.

2.2.1 General Aggregation

The procedures of aggregation can be similarly applied to a large-scale linear time-invariant system; i.e., consider, a linear controllable system

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \tag{2.2.2a}$$

$$y(t) = Dx(t) \tag{2.2.2b}$$

where $x(t)$, $u(t)$, and $y(t)$ are $n \times 1$, $m \times 1$, and $r \times 1$ state, control, and output vectors, respectively, and A, B, and D are $n \times n$, $n \times m$, and $r \times n$ matrices. It is desired to describe the time behavior of

$$z(t) = Cx(t),\ z(0) = z_0 = Cx_0 \tag{2.2.3}$$

where C is an $l \times n$ $(l < n)$ constant aggregation matrix and $l \times 1$ vector z is called the aggregation of x. On the other hand, assuming that x is available and starting from $z_0 = Cx_0$, it is desired to maintain the relation (2.2.3). Without loss of generality, it is assumed that rank$(C) = l$. Then the aggregated system is described by

$$\dot{z}(t) = Fz(t) + Gu(t),\ z(0) = z_0 \tag{2.2.4a}$$

$$\hat{y}(t) = Kz(t) \tag{2.2.4b}$$

where the pair (F,G) satisfy the following, so-called *dynamic exactness* (perfect aggregation) conditions:

$$FC = CA \tag{2.2.5}$$
$$G = CB \tag{2.2.6a}$$
$$KC \cong D \tag{2.2.6b}$$

The aggregation of the dynamic system (2.2.2) is illustrated in Figure 2.4b. The vector $\hat{y}$ is an $r \times 1$ approximate output. It is noted that (2.2.5)–(2.2.6) cannot hold simultaneously if $l < n$ and (2.2.2) is assumed to be irreducible, hence the condition (2.2.6b) is an approximation at best (Hickin and Sinha, 1980).

If an error vector is defined as $e(t) = z(t)\text{-}Cx(t)$, then its dynamic behavior is given by $e(t) = Fe(t)+(FC\text{-}CA)x(t)+(G - CB)u(t)$, which reduces to $\dot{e}(t) = Fe(t)$ if conditions (2.2.5)–(2.2.6a) hold. Hence, if $e(0) = 0$, then $e(t) = 0$ for all $t \geq 0$. Should $e(0) \neq 0$ but F be a stable matrix, then $\lim_{t \to \infty} e(t) = 0$; i.e., dynamic exactness condition (2.2.5)–(2.2.6a) is asymptotically satisfied.

In order to determine the aggregation matrices (F,G), two procedures can be followed. The first results from the Penrose solvability condition (Penrose, 1955), i.e.,

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$F = CAC^T (CC^T)^{-1} \tag{2.2.7}$$

Thus, once matrix C is known, the aggregated matrix F is obtained by (2.2.7) and the aggregated control matrix G is determined from (2.2.6a). The analysis of identities (2.2.5)–(2.2.7) gives some insight in the choice of the "aggregation matrix" C. Aoki (1968, 1971) has shown that the analysis of (2.2.5) will lead to a description of the aggregated state vector $z(t)$, which is a linear combination of certain modes of $x(t)$. It must be noted, however, that the aggregated matrix F is obtained from (2.2.7) only if the conditions (2.2.5)–(2.2.6) are satisfied. Under these circumstances, the eigenvalues of F constitute a subset of eigenvalues of A. As mentioned earlier for the static models of econometrics, the dynamic exactness (perfect aggregation) is an idealized situation. The use of (2.2.7), as in Michailesco and Siret (1980), is an approximation which in fact minimizes the square of the norm $\|FC - CA\|$ unless a consistency relation

$$CAC^+C = CA \tag{2.2.8}$$

is satisfied. In (2.2.8), the matrix $C^+ \triangleq C^T(CC^T)^{-1}$ is the generalized inverse of C.

The linear system aggregation procedure described thus far requires the knowledge of all eigenvalues of A. This requirement would make the method rather impractical for large-scale systems. A second approach which does not require the knowledge of the eigenvalues of A has also been proposed by Aoki (1968). Consider the controllability matrix*:

$$W_A \triangleq [B, AB, \ldots, A^{n-1} B] \tag{2.2.9}$$

and a modified controllability matrix of (2.2.4),

$$W_F \triangleq [G, FG, \ldots F^{n-1} G] \tag{2.2.10}$$

it can be seen from (2.2.5)–(2.2.6b) that these matrices are related by

$$W_F = CW_A \tag{2.2.11}$$

Thus using the generalized (pseudo-) inverse, matrix C can be obtained by

$$C = W_F W_A^+ = W_F W_A^T \left(W_A W_A^T\right)^{-1} \tag{2.2.12}$$

*For a discussion on controllability, see Chapter 3.

controlengineers.ir
@controlengineers

since by initial controllability assumption rank $(W_A) = n$. Therefore, if F is specified, say $F = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_l)$, and G is chosen to make (2.2.4) completely controllable, i.e., rank$(W_F) = l$, then C is obtained by (2.2.12). It is noted here that this procedure would, in effect, forego the dynamic exactness conditions (2.2.5)–(2.2.6). Before it is demonstrated that the modal (dominate pole) aggregation (Davison, 1966, 1968; Chidambara, 1969) is a special case of the above, a numerical example is presented.

Example 2.2.1. Consider a third-order unaggregated system described by

$$\dot{x} = \begin{bmatrix} -0.1 & 1 & 2 \\ 1 & -4 & 0 \\ 2 & 0 & -6 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u \tag{2.2.13}$$

It is desired to find a second-order aggregated model for this system.

SOLUTION: This example is solved using the two methods described above. The first solution is obtained by the use of the eigenvalues of A in (2.2.13), which are $\lambda\{A\} = \{-0.70862, -6.6482, -4.1604\}$. From the relative magnitudes of $\lambda_i\{A\}$, it is clear that the first mode is the slowest of all three. A possible choice for aggregation matrix C can be

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix} \tag{2.2.14}$$

which implies that the first aggregated state is chosen to be approximately the slowest mode while an average of the two faster modes constitutes the second aggregated state. From (2.2.7) and (2.2.6a), the aggregated model becomes

$$\dot{z}(t) = Fz(t) + Gu(t) = \begin{bmatrix} -0.1 & 3 \\ 1.5 & -5 \end{bmatrix} z(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \tag{2.2.15}$$

It is clear from (2.2.13)–(2.2.15) that for this choice of aggregation matrix, the condition (2.2.5) for dynamic exactness is not satisfied. The resulting aggregated system, as mentioned earlier, is at best an approximation. The resulting error vector $e(t)$ satisfies

$$\dot{e}(t) = Fe(t) + \begin{bmatrix} 0 & 0.5 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix} x(t) \tag{2.2.16}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

An alternative choice of C,

$$C=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix} \tag{2.2.17}$$

results in an aggregated system

$$\dot{z}(t)=\begin{bmatrix}-0.1 & 1\\ 1 & -4\end{bmatrix}z(t)+\begin{bmatrix}1\\ 1\end{bmatrix}u(t) \tag{2.2.18}$$

with an error system

$$\dot{e}(t)=Fe(t)+\begin{bmatrix}0 & 0 & -2\\ 0 & 0 & 0\end{bmatrix}x(t) \tag{2.2.19}$$

This latter choice of aggregation matrix provided better results as evident from (2.2.19). However, this case is not exact either. In later discussions it will be demonstrated how dynamic exactness can be achieved. In some problems F may even turn out to be a null matrix regardless of what C one may try.

The second solution is obtained by evaluating controllability matrices. Following the discussions made earlier, let F = diag (–0.70862,–4.1604); then by trial and error a $G = (1\ 1)^T$ column vector can be found so that the pair (F,G) is controllable. The controllability matrix W_A and matrix W_F defined by (2.2.9)–(2.2.10) are given,

$$W_A=\begin{bmatrix}1 & 2.9 & -11.29\\ 1 & -3 & 14.9\\ 1 & -4 & 29.8\end{bmatrix},\quad W_F=\begin{bmatrix}1 & -0.70862 & 0.502\\ 1 & -4.1604 & 17.309\end{bmatrix} \tag{2.2.20}$$

and hence a possible aggregation matrix is obtained from (2.2.12),

$$C=\begin{bmatrix}0.32 & 1.08 & -0.4\\ -0.24 & 1.5 & -0.26\end{bmatrix} \tag{2.2.21}$$

This scheme leads to dynamically inexact aggregation also. In fact, its approximate nature is somewhat more difficult to estimate than the first procedure.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Example 2.2.2. For the third order unaggregated system described by

$$\dot{x} = \begin{bmatrix} -0.1 & 1 & 2 \\ 0 & -1.2 & 1 \\ 0 & 0 & -3 \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} u$$

find an aggregated system.

SOLUTION: The first solution is obtained by the use of eigenvalues of A matrix of the above system which are $\lambda_1 = -0.1$, $\lambda_2 = -1.2$, and $\lambda_3 = -3$. From the relative magnitudes of λ_i it is clear that the slowest modes of the system are the first two, hence a structure for aggregation matrix C can be

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

which implies that the two potential aggregated states $z_1(t)$ and $z_2(t)$ are chosen to be the slowest mode of the original system and an average of the next two modes of the full model, respectively. The aggregated matrices F and G are obtained from Eqs. (2.2.7) and (2.2.6a), that is,

$$\dot{z}(t) = Fz(t) + Gu(t) = \begin{bmatrix} -0.1 & 3 \\ 0 & -1.5 \end{bmatrix} z(t) + \begin{bmatrix} 1 & 1 \\ 0.5 & 1 \end{bmatrix} u(t)$$

It is noted that for this aggregation, the aggregability measure is given by

$$FC - CA = \begin{bmatrix} 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & \frac{-1}{4} & \frac{1}{4} \end{bmatrix} \neq 0$$

which indicates that aggregability condition Eq. (2.2.5) is not satisfied. This would mean that this choice of C needs to be changed. One can, through trial and error, come up with an aggregation matrix such that some desired goal would be satisfied. Among desirable goals are close agreements between time (or frequency) responses of the full- and reduced-order models, or full aggregability.

CAD Example 2.1. In this example, a sixth-order system representing a modified form of a flexible booster (Jamshidi, 1983) is reduced to a sec-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ond- and a fourth-order system; step responses for the full-order and reduced-order models are obtained for comparison purposes. In this example we use LSSRB (see Appendix C).

```
>> a = [-.42 -.2 -.008 0 .95 ld-5;1 -.053 -3d-4 10d-4 3.5d-4 0;0 0 0 1 . . .
00;00 -688 -5.900;000001;0000 -4880 -18.51;
>> b=[-9.5;-5.2d-2;10d-2;899;10d-3;-488.5]
>> c=[1 0 0 8.85d-4 0 -9.82d-3;0 1 0 0 0 0];
>>r1 =[1 0 0 0 0 0;0 1 0 0 0 0];f1 =r1*a*pinv(r1) %pinv is pseudo-inverse
        command

f1 =
        - 0.4200          -0.2000
          1.0000          -0.0530

>> g1=r1*b,h1=c*pinv(r1)
g1 =
        -9.5000
        -0.0520
h1 =
        1          0
        0          1

>> d = [0;0];time = [0:0.1: 20.]; % define D matrix and time vector for %
        simulation

% Now, a 4th order reduced model
>> r2 = [eye (4), 0 * ones (4,2)];
>> f2 = r2*a*pinv(r2),g2 = r2*b,h2 = c*pinv(r2)

f2 =
         -0.4200        -0.2000        -0.0080              0
          1.0000        -0.0530        -0.0003         0.0010
               0              0              0         1.0000
               0              0       -688.000        -5.9000
g2 =
         -9.5000
         -0.0520
          0.1000
        899.0000

h2 =
        1.0000              0              0         0.0009
             0         1.0000              0              0
```

controlengineers.ir @controlengineers

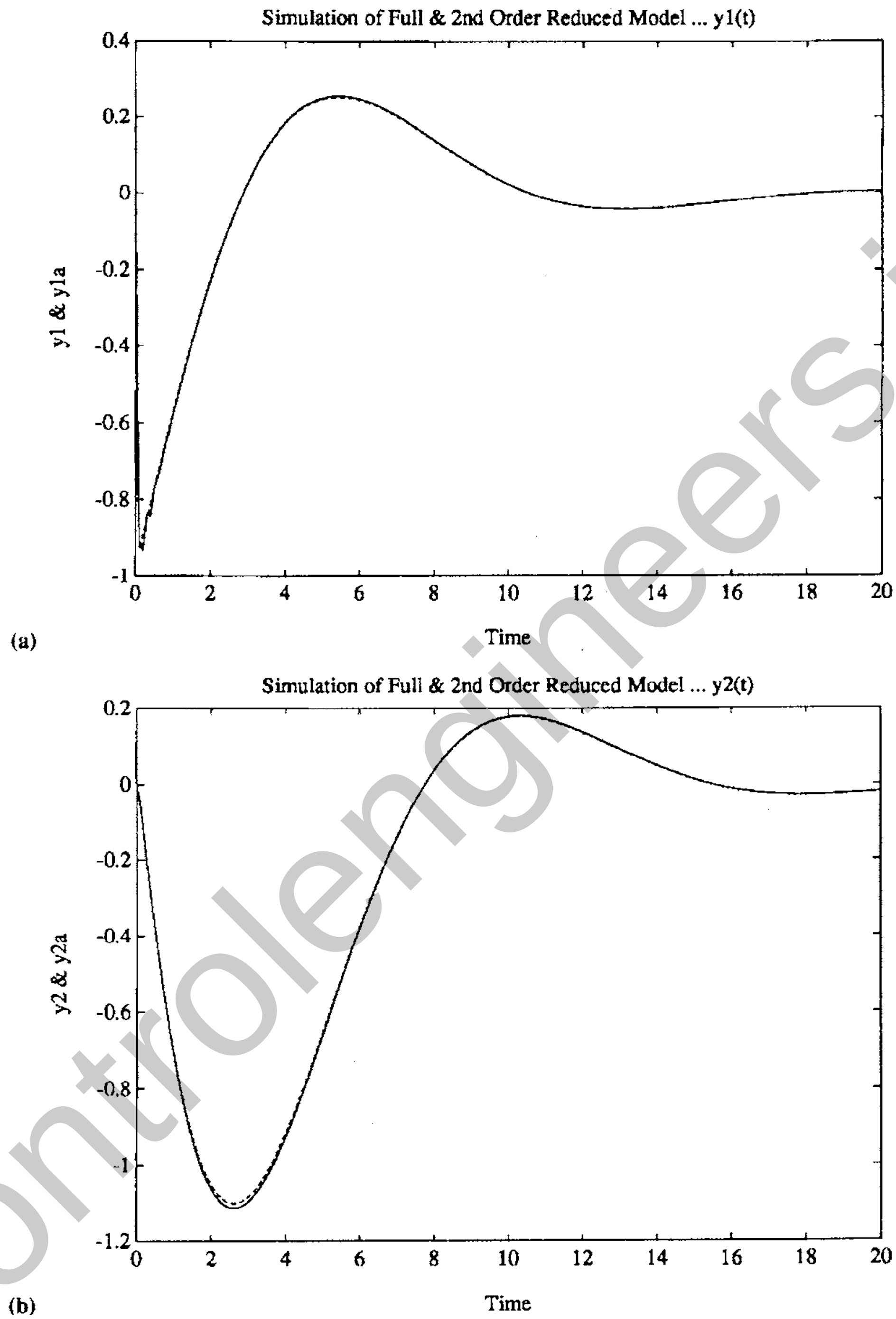


Figure 2.5 Impulse responses for full-order and three reduced-order models for CAD Example 2.1.

(a) Second-order reduced model—output y_1.
(b) Second-order reduced model—output y_2.
(c) Fourth-order reduced model—output y_1.
(d) Fourth-order reduced model—output y_2.

controlengineers.ir
@controlengineers

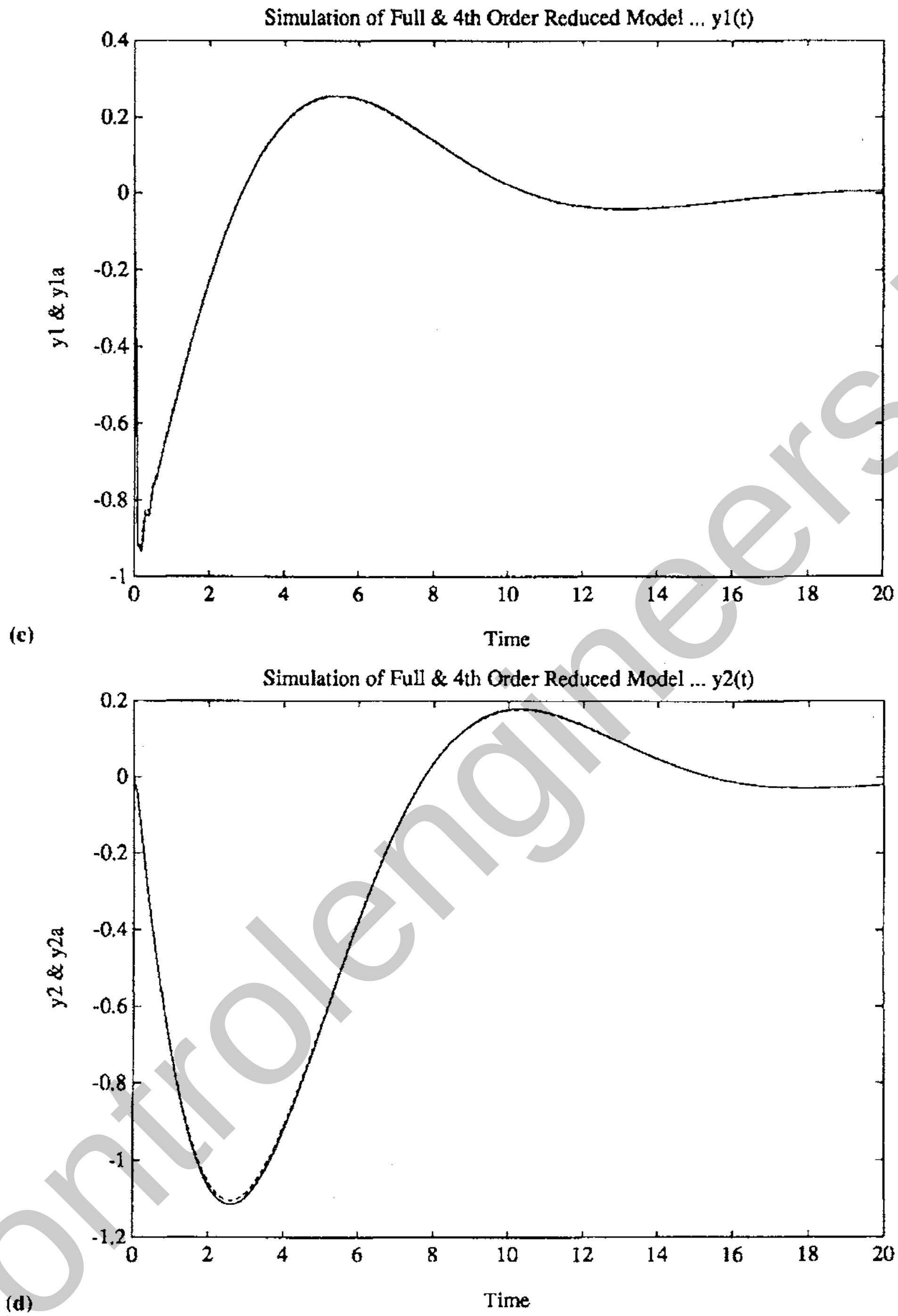


Figure 2.5 (*Continued*)

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

```
>> % Impulse responses for full and reduced-order models
>> uin = 0 * time; uin (1,1) = 1;
>> yf = Isim (a,b,c,d,uin, time);
>> y1 = Isim (f1,g1,h1,d,uin, time);
>> y2 = Isim (f2,g2,h2,d,uin, time);
>> Plot (time, yf(:1), time, y1(:1))
>> P!ot !time, yf !:2!, time, y2 (:2))
```

The resulting impulse responses for the full-order system and typical output responses of the second- and fourth-order reduced models are shown in Figure 2.5.

```
ans =
        -0.5800
        -1 .0000
        -5.0000

>> % Fifth order reduced model
>> r5=[eye(5h0*ones(5,2)];
>> f5=r5*a*pinv(r5),g5=r5*b;h5=c*pinv(r5);

f5 =
  -0.5800     0          0        -0.2690      0
   0         -1.0000     0         0           0
   0          0         -5.0000    2.1200      0
   0          0          0         0         377.0000
  -0.1410     0          0.1410   -0.2000     -0.2800

>> eig(f5)

ans =
        -0.8839 + 8.40851
        -0.8839 + 8.40851
        -3.7903
        -0.3019
        -1.0000
```

Note that in the first-order aggregation, all seven eigenvalues of the system (three complex conjugate pairs and a real one) have been approximated by only one real eigenvalue at –0.58. For a third-order aggregation however, the system's eigenvalues are approximated by three real ones at –0.58, –1.0, and –5.00. One should not expect good approximation of the full-order models, because none of the dominant complex conjugate eigenvalues, that is, those close to the $j\omega$-axis, have been retained. The fifth-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

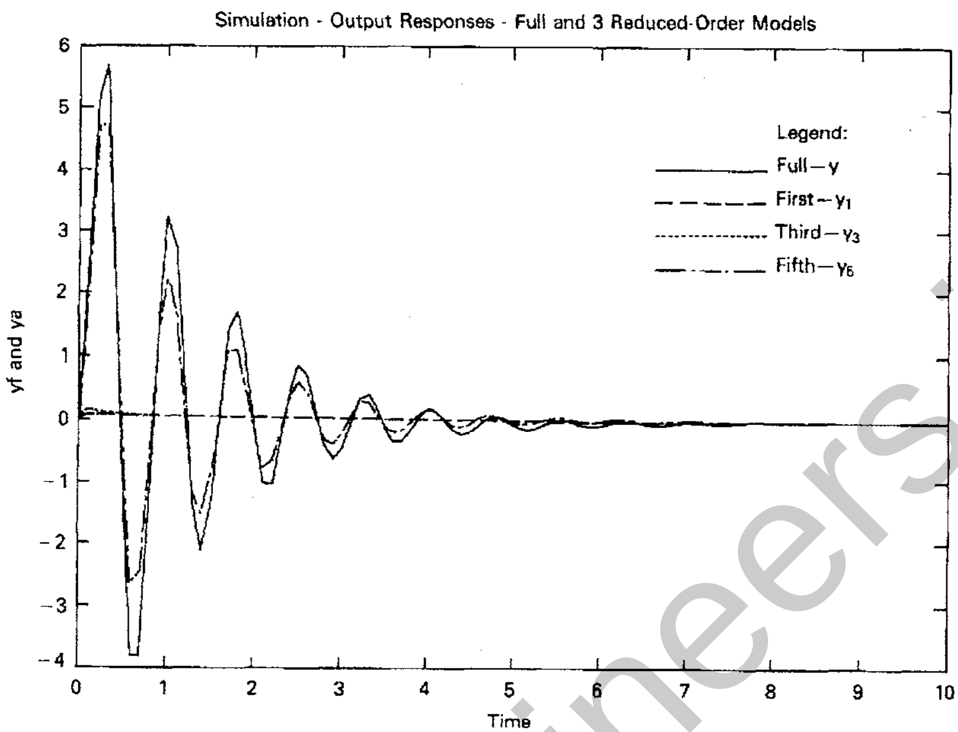


Figure 2.6 Step responses for CAD Example 2.1.

order aggregated model, on the other hand, has been approximated by three real and a pair of complex conjugate eigenvalues. Following are the step response calculations and a plot of the output of the full-order model and three reduced-order models in Figure 2.6.

```
>> time= [0:.1 :10];
>> yf=step(a,b,c,d,l,time);
>> y1=step(f1,g1,h1,d,1,time);
>> y3=step(f3,g3,h3,d,1,time);
>> y5=step(f5,g5,h5,d,1,time);
>> plot(time,yf,time,yl,time,y3,time,y5)
>> xlabel('Time'),ylabel('yf & y1,3,5')
>> title('Step Responses Full and Three Reduced-Order Models')
```

The time response verifies the points brought up earlier regarding the eigenvalues of the full-order and reduced-order models. These plots make it clear that as the number of retained modes of the system increases, the step response of the reduced model gets closer to that of the full model. On

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

the other hand, with only a few modes (1 or 2) retained, there are great discrepancies between the two models.

2.2.2 *Modal Aggregation*

Next, we concentrate on a special case of the general aggregation considered in Sec. 2.2.1. The method is based on the early works of Davison (1968) and Chidambara (1969). It is essentially a mapping based on a submatrix of the full-order model's modal matrix. The columns of this matrix are its eigenvectors, and can be ordered in accordance with their dominancy, that is, the eigenvector with eigenvalues closest to the jw-axis.

Consider the linear time-invariant system Eq. (2.2.2), repeated here for convenience,

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(0) = x_0 \tag{2.2.2a}$$

$$y(t) = Dx(t) \tag{2.2.2b}$$

Then the aggregated matrix pair (F, G) is given by

$$F = M_l P \Lambda P^T M_l^{-1} \tag{2.2.22}$$

$$G = M_l P M^{-1} B \tag{2.2.23}$$

where M is the modal matrix of (2.2.2) consisting of the eigenvectors of A arranged in descending* order of the $\mathrm{Re}[\lambda_i\{A\}]$. M_l is an $l \times l$ matrix which includes the l dominant eigenvectors of A corresponding to the retained modes of the original system, assumed to be nonsingular, and

$$P{:}\left[I_l \,\vdots\, 0\right] \tag{2.2.24}$$

is an $l \times n$ transformation matrix. Modal matrix M can be represented by

$$M = \begin{array}{c} \\ l\{ \\ \\ \end{array} \underbrace{\overbrace{\begin{bmatrix} M_l & \vdots & M_{12} \\ \cdots & & \cdots \\ M_{21} & \vdots & M_{22} \end{bmatrix}}^{l}}_{n} \}n \tag{2.2.25}$$

---

*The ordering can also be in accordance to any number of the eigenvalues which are to be retained.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The aggregation matrix C in $z = Cx$ is given by (Lamba and Rao, 1978),

$$C = M_l P M^{-1} \tag{2.2.26}$$

It is noted that this scheme works for the case where A has complex or repeated eigenvalues as well. Under those conditions, the columns of M can be real and imaginary parts of the complex eigenvectors or generalized eigenvectors in addition to regular eigenvectors (see Problem 2.6).

Example 2.2.3. Consider a third-order system

$$\dot{x}(t) = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0.833 & -2.1667 & -0.333 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} u(t) \tag{2.2.27}$$

It is desired to find a reduced-order model.

SOLUTION: This example will be solved by using both the modal aggregation described by (2.2.22)–(2.2.23) and the general aggregation under conditions (2.2.5)–(2.2.7).

a. *Modal*. The eigenvalues of A are

$$\lambda\{A\}=\{0.5,\ 1,\ -0.333\} \tag{2.2.28}$$

which indicates that the system is unstable. The resulting aggregated model based on (2.2.22)–(2.2.23) is given by

$$\dot{z}(t) = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} z(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \tag{2.2.29}$$

which has retained the two eigenvalues with positive real parts. It is noted that the aggregated system is also unstable.

b. *General.* This solution is demonstrated by three different choices of matrix C and each time the dynamic exactness is checked. Consider

$$C^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix} \tag{2.2.30}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which by virtue of (2.2.7) and (2.2.6a) leads to

$$\dot{z}^1(t) = \begin{bmatrix} 0.5 & 0.5 \\ 0.416 & -0.75 \end{bmatrix} z^1(t) + \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} u(t) \tag{2.2 31}$$

This aggregation leads to an error system equation,

$$\dot{e}^1(t) = F^1 e^1(t) + \begin{bmatrix} 0 & -0.25 & 0.25 \\ 0 & 0.208 & -0.208 \end{bmatrix} x(t) \tag{2.2.32}$$

which indicates that this choice of C is far from a perfect aggregation. Next, a second C matrix,

$$C^2 = \begin{bmatrix} 0 & 1 & 0.5 \\ 0.5 & 0 & 0.5 \end{bmatrix} \tag{2.2.33}$$

is tried and results in the following aggregated and error systems:

$$\dot{z}^2(t) = \begin{bmatrix} 1 & 0 \\ -0.8335 & 0.5 \end{bmatrix} z^2(t) + \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} u(t) \tag{2.2.34}$$

$$\dot{e}^2(t) = F^2 e^2(t) + \begin{bmatrix} 0 & 0 & 0 \\ 0.417 & 0 & 0.417 \end{bmatrix} x(t) \tag{2.2.35}$$

The second choice of C matrix, although still not exact, has resulted in a reduced-order model closer to perfect aggregation as is evident from (2.2.35) as compared to (2.2.32). The third choice of C matrix is

$$C^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.2.36}$$

which provides the following aggregated and error systems:

$$\dot{z}^3(t) = \begin{bmatrix} 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} z^3(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \tag{2.2.37}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{e}^3(t) = F^3 e^3(t) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x(t) \tag{2.2.38}$$

This choice of aggregation matrix has resulted in a perfect aggregation in which the dynamic exactness conditions (2.2.5)–(2.2.6a) are satisfied. Comparison of this third aggregated system (2.2.37) and the aggregated system through modal aggregation given by (2.2.29) reveals that the two aggregated systems are identical. This implies that the modal approach is in fact a special case of the exact one, as expected.

Next, let us discuss another modal aggregation, originally due to Chidambara (1969) but also presented by Rao and Lamba (1974). Let the large-scale linear time-invariant system (2.2.2a) be rewritten as

$$\dot{x}(t) = \begin{bmatrix} \dot{z}(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_1 & A_{12} \\ A_{21} & A_2 \end{bmatrix} \begin{bmatrix} z(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t) \tag{2.2.39}$$

where z is the aggregated state. System (2.2.39) can be reduced to its modal form,

$$\begin{bmatrix} \dot{w}(t) \\ \dot{v}_2(t) \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} w(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix} u(t) \tag{2.2.40}$$

where w is the vector of retained dominant (aggregated) variables,

$$x = Mv = M\left[w \mid v_2\right]^T, \quad \Lambda = \text{Block-diag}\ (\Lambda_1, \Lambda_2) = M^{-1}AM$$

and

$$\Gamma = \left[\Gamma_1 \mid \Gamma_2\right]^T = M^{-1}B$$

and M is the modal matrix corresponding to matrix A. For simplicity, it is assumed that all the eigenvalues of A are real and distinct. The approach can be similarly extended for complex and/or repeated eigenvalues (see Problem 2.7). Referring back to systems (2.2.39)–(2.2.40), it is desired to retain $l(l < n)$ dominant modes (vector w) of (2.2.40), i.e.,

$$\dot{w}(t) = P\Lambda P^T w(t) + P\Gamma u(t) \tag{2.2.41}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where P is given by (2.2.24) and $w = Pv$. In (2.2.40) let the dimension of Λ_1, be $l \times l$ and take the Laplace transform of the v_2-equation to yield

$$V_2(s) = \left(sI - \Lambda_2\right)^{-1} \Gamma_2 U(s) \tag{2.2.42}$$

If only the DC transmission between $u(t)$ and $v_2(t)$ is of interest, and since Λ_2 represents nondominant (fast) modes, (2.2.42) can be approximated by

$$v_2(t) \cong -\Lambda_2^{-1}\Gamma_2 u(t) \triangleq Lu(t) \tag{2.2.43}$$

The partitioned forms of x and v lead to

$$\begin{bmatrix} z \\ x_2 \end{bmatrix} = \begin{bmatrix} M_1 & M_{12} \\ M_{21} & M_2 \end{bmatrix} \cdot \begin{bmatrix} w \\ v_2 \end{bmatrix} \tag{2.2.44}$$

or

$$z = M_1 w + M_{12} v_2, \quad x_2 = M_{21} w + M_2 v_2 \tag{2.2.45}$$

Solving for w in the first equation of (2.2.45) and substituting it in the second while using (2.2.43) leads to

$$x_2 = M_{21} M_1^{-1} z + \left(M_2 - M_{21} M_1^{-1} M_{12}\right) Lu \triangleq Nz + Eu \tag{2.2.46}$$

Eliminating x_2 in the $\dot{z}$-equation (2.2.39) by virtue of (2.2.46) leads to the aggregated model

$$\dot{z}(t) = Fz(t) + Gu(t) \tag{2.2.47}$$

$$y = Kz + Hu \tag{2.2.48}$$

where

$$F = A_1 + A_{12} N \tag{2.2.49a}$$

$$G = B_1 + A_{12} E \tag{2.2.49b}$$

$$K = D_1 + D_2 N \tag{2.2.49c}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$H = D_2 E \tag{2.2.49d}$$

In this method, the effects of the nondominant modes have been neglected to result in the reduced model.

In the case when the steady-state response is not of great importance, one can use the original modal reduction scheme suggested by Davison (1966,1968). In an abbreviated fashion, the aggregated system matrices are given by,

$$F = M_l P \Lambda P^T M_l^{-1} \tag{2.2.50}$$

$$G = M_l P M^{-1} B \tag{2.2.51}$$

$$C = M_l P M^{-1} \tag{2.2.52}$$

$$K \cong DC^+ \tag{2.2.53}$$

where C is the aggregation matrix and C^+ is the pseudo inverse of C.

Example 2.2.4. Consider a fourth-order system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.6 & -9.22 & -33.32 & -11.3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} u \tag{2.2.54}$$

It is desired to find an aggregated system for it.

SOLUTION: The eigenvalues of A are -0.1, -0.2, -5.0, and -6.0, which indicates that the first two eigenvalues are dominant and hence a second-order reduced model will be sought. The modal matrix whose columns are ordered in ascending order of eigenvalues real parts is

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -0.1 & -0.2 & -5 & -6 \\ 0.01 & 0.04 & 25 & 36 \\ -0.001 & -0.008 & -125 & -216 \end{bmatrix}$$

and diagonal matrix $\Lambda = \text{diag}\{-0.1, -0.2, -5, -6\}$ and modal input matrix

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$\Gamma^T = (11.48,-11.5,0.12,-0.074)$. The aggregated matrices are

$$F = \begin{bmatrix} 0 & 1 \\ -0.02 & -0.3 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1.144 \end{bmatrix}$$

which results in a set of retained eigenvalues $\lambda\{F\} = \{-0.1,-0.2\}$ corresponding to the dominant ones of (2.2.54). This example was also solved using the first modal aggregation and the results were identical.

Before we leave our initial discussions on aggregation, it is useful to see the correlation between the exact and modal aggregations. Let ξ be the right eigenvector of A corresponding to eigenvalue λ, i.e., $A\xi = A\xi$. Premultiplying both sides of this equality by C leads to

$$CA\xi = \lambda C\xi \tag{2.2.55}$$

Now denoting $\gamma = C\xi$ and remembering the condition (2.2.5), (2.2.55) can be rewritten as

$$FC\xi = F\gamma = \lambda\gamma \tag{2.2.56}$$

which indicates that γ is a right eigenvector of matrix F under perfect aggregation conditions, provided that $C\xi \neq 0$. Therefore F inherits a set of eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_l\}$ of A corresponding to those eigenvectors of A with $C\xi \neq 0$. Now let the left eigenvectors of A, i.e., the set of all row vectors v_i, such that $v_iA = \lambda_i v_i$, be the rows of the aggregation matrix, i.e.,

$$C = \begin{bmatrix} v_1 \\ -- \\ v_2 \\ -- \\ \vdots \\ -- \\ v_l \end{bmatrix} \tag{2.2.57}$$

Then the aggregated system (2.2.4) using (2.2.7) reduces to

$$\dot{z} = \Lambda_l z + Gu$$

when $\Lambda_l = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_l)$ and $G = CB$. Moreover, it is easy to see that this aggregation does satisfy the dynamic exactness condition (2.2.5). Assume that the modal representation of (2.2.2a) is $\dot{q} = \Lambda q + \Gamma u$ with

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$\Lambda = M^{-1}AM =$ Block-diag$\{\Lambda_1, \Lambda_2\}$ and $\Gamma = M^{-1}B = \left[\Gamma_1^T \vdots \Gamma_2^T\right]^T$. Thus, the modal representation of the reduced model is $\dot{q}_l = \Lambda_l q_l + \Gamma_1 u$ or $q_l = \left[I_l \vdots 0\right] M^{-1} x = C_0 x$, hence the aggregation Equation (2.2.3) is redefined by $z = Cx = Mq_l = MC_0 x$ or $C = MC_0$. Note that to compute C_0 it is not necessary to obtain M^{-1}. The l rows of C_0 are the eigenvectors of A^T associated with the l retained eigenvalues.

The above development indicates that if the aggregation matrix is properly chosen, one can use exact aggregation to retain the dominant modes of the system, i.e., modal aggregation. This explains the correlation between general and modal aggregation as demonstrated in Example 2.2.3 and further in Problem 2.5.

Example 2.2.5. Consider the following seventh-order system

$$\dot{x} = \begin{bmatrix} -0.58 & 0 & 0 & -0.269 & 0 & 0.2 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -5 & 2.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 377 & 0 & 0 \\ -0.141 & 0 & 0.141 & -0.2 & -0.28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0838 & 2 \\ -173 & 66.7 & -116 & 40.9 & 0 & -66.7 & -16.7 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [1\ \text{-}1\ 1\ 1\ 1\ 0\ 1\ 0]x$$

whose unforced form represents a single-machine infinite bus power system (Jamshidi, 1983). It is desired to find some modally aggregated models.

SOLUTION: The eigenvalues of the system matrix are $-0.362 \pm j0.556$, $-0.858 \pm j8.38$, -3.94, and $-8.55 \pm j8.2$ which indicates that the system has as few as 2 or as many as 4 or 5 dominant modes. The modal matrix, set in the order of dominancy of the eigenvalues is given by

$$M = \begin{bmatrix} 0.1692 & 0.2416j & 0.0080 & 0.0087j & 0.0400 & 0.0001 & + & 0.0029j \\ 1.0000 & 0.0000j & -0.0282 & 0.0027j & 0.1692 & -0.0002 & + & 0.0151j \\ -0.0592 & 0.1256j & -0.0583 & 0.0942j & 0.2593 & -0.0002 & + & 0.0001j \\ -0.1624 & 0.2593j & 0.2584 & 0.4144j & 0.1298 & 0.0008 & + & 0.0006j \\ -0.0002 & 0.0005j & 0.0086 & 0.0067j & -0.0014 & -0.0000 & + & 0.0000j \\ 0.6377 & 0.5564j & -0.0265 & 0.2357j & -0.4972 & -0.1217 & - & 0.1157j \\ -0.2970 & 0.0533j & 1.0000 & 0.0000j & 1.0000 & 1.0000 & - & -0.0000j \end{bmatrix}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and the block-diagonal Jordan matrix J is given by

$$J = \begin{bmatrix} -0.362 & 0.556 & 0 & 0 & 0 & 0 & 0 \\ -0.556 & -0.362 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.858 & 8.38 & 0 & 0 & 0 \\ 0 & 0 & -8.38 & -0.858 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3.94 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -8.55 & 8.2 \\ 0 & 0 & 0 & 0 & 0 & -8.2 & -8.55 \end{bmatrix}$$

Then, three of the aggregated models using the modal approach, that is, Eq. (2.2.49) will be,

$k = 2$

$$(F,G,H,K) = \left(\begin{bmatrix} -0.752 & 0.200 \\ -2.303 & 0.027 \end{bmatrix}, \begin{bmatrix} -0.76 \\ -0.112 \end{bmatrix}, \begin{bmatrix} -2.90 & 0.075 \end{bmatrix}, \begin{bmatrix} -1.04 \end{bmatrix} \right)$$

$k = 4$

$$(F,G,H,K) = \left(\begin{bmatrix} -.86 & 0.2 & 0.22 & -0.21 \\ -1.41 & -0.01 & 1.1 & 0.29 \\ 0 & 0 & -5.0 & 2.1 \\ -18.18 & 0.973 & -43.5 & 3.4 \end{bmatrix}, \begin{bmatrix} 1.4 \\ 2.1 \\ 1 \\ -124 \end{bmatrix}, \begin{bmatrix} -0.41 & -0.01 & 2.1 & 1.3 \end{bmatrix}, \begin{bmatrix} 7.22 \end{bmatrix} \right)$$

$k = 5$

$$(F,G,H,K) = \left(\begin{bmatrix} -1.1 & 0.21 & -0.52 & -0.15 & -6.45 \\ -3.0 & 0.07 & -2.6 & 0.59 & -3.22 \\ 0 & 0 & -5.0 & 2.12 & 0 \\ 0 & 0 & 0 & 0 & 377 \\ -0.14 & 0 & 0.14 & -0.20 & -0.28 \end{bmatrix}, \begin{bmatrix} 1.55 \\ 2.8 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1.97 & 0.07 & -1.62 & 1.6 & -32.24 \end{bmatrix}, \begin{bmatrix} 0 \end{bmatrix} \right)$$

CAD Example 2.2. In this CAD example, the single-machine system of Example 2.1 will be used with LSSTB and MATLAB to illustrate both

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

modal aggregation cases.

```
>>% modal aggregation No. 1
>>a = [ –.58 0 0  –.269 0 .2 0; 0 –1 0 0 0 1 0;0 0 –52.12 0 0 0; 0 0 0 0 377 0 0; . . .  –
.141 0 .141 –.2 –.28 0 0; 0 0 0 0 0 .0838 2; –173 66.7, –116 40.9 0 –66.7 –16.71;
>>b=[1 0 1 0 1 0 1]'; c=[1 -1 1 1 0 1 0];
>>% modal + Jordan matrices
>>Im,j] = eig (a);
>> % eigenvalues are:
>>diag (j)

ans =
        –8.5478 + 8.2044i
        –8.5478 – 8.2044i
        –0.8586 + 8.3793i
        –0.8586 – 8.3793i
        –0.3623 + 0.5564i
        –0.3623 – 0.5564i
        –3.9388 + 0.0000i

>>% order eigenvectors in modal matrix
>>m_ord = [m(:,5),m(:,6),m(:,3),m(:,4),m(:,7), m(:,1),m(:,2)];
>>%extract 2nd order modal matrix
>>m1 = m_ord (1 :2, 1 :2), m12 = m_ord (1 :2, 3:7); m21 = m_ord (3:7,1:2);

        >>m2 - m_ord(3:7,3:7);
        m1 =
                Columns 1 through 2
                0.1692 – 0.2416i            0.1692 + 0.2416i
                1.0000 + 0.0000i            1.0000 – 0.0000i

>>p1 = leYe(2),0*ones (2,5)]; gama = inv(m_ord)*b;
>>gama 2 = gama (3:7); d = c * m_ord; js = j(3:7,3:7);
>>L = –inv(j2) * gama2; n = m21 * inv(m1); e = (m2 + m21 * inv(ml) * ml2) * 1;
                al = a(1 :2, 1 :2);
>>a12 = a(1:2,3:7); d1 = d (1:2); b1 = b(1:2);
>>% Second-order aggregated model
>>f2=a1 + a12 * n, g2=b1 + a12 * e, h2=d1 * inv~m),
                k2 = d1 * inv(m1) * m12 * L

f2 =
        –0.7520 + 0.0000i    0.2003 – 0.0000i
        –2.3036 + 0.0000i    0.0275 – 0.0000i

g2 =
        –0.7508 – 0.0000i
        –0.1125 – 0.0000i
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

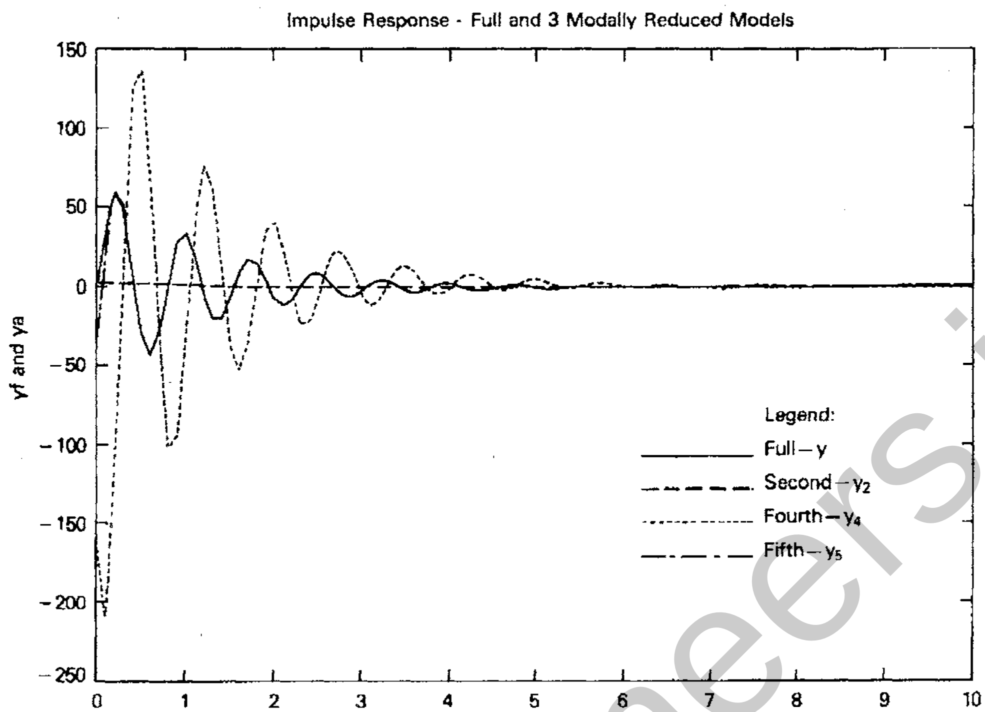


Figure 2.7 Impulse responses for CAD Example 2.2 using first method of modal aggregation.

h2 =

−2.897i + 0.0000i 0.0754 − 0.0000i

k2 =

−1.0407 − 0.0000i

⋮

>> % Fourth-order aggregated model

f4 =

−0.8630 + 0.0000i	0.1980 − 0.0000i	0.2193 + 0.0000i	0.2097 − 0.0000i
−1.4151 + 0.0000i	−0.0098 − 0.0000i	1.0967 + 0.0000i	0.2964 − 0.0000
0 0	−5.0000	2.1200	
−18.1811 − 0.0000i	0.9728 − 0.0000i	−43.5067 − 0.0000i	3.4310 + 0.0000i

g4 =

1.0E + 002 *
0.0142 + 0.0000i
0.0211 + 0.0000i
0.0100

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers


```
        -1.2399 - 0.0000i
h4 =
        -0.4151 + 0.0000i   -0.0098 -0.0000i   2.0967 + 0.0000i   1.2964 - 0.0000i

k4 =
        7.2197 + 0.0000i

>> % impulse response comparison >>time = [0:0.1: 10];
>> imf = impulse (a,b,c,d,1,time); >> ial = impulse (f,g,h,k,1, time);
>> Plot (time, imf, time, ial, time, . . ., ia5)
>> xlabel ('Time'), ylabel ('yf & ya')
>> title ('Impulse Response - Full & 3 Modally Reduced Models')
```

The impulse responses of the full-order and three modally reduced models using the relations Eq. (2.2.49) are shown in Figure 2.7.

In order to implement the second modal aggregation, given by Eqs. (2.2.50) to (2.2.53), a *.m file in PC-MATLAB was written which is created using any screen editor and saved on the same directory as PC-MATLAB itself. The file mod.m is shown.

```
function [f,g,h,r]=mod(a,b,c,d,m,k)
% Modal aggregation Case 2 (Eqs. 2.2.50 - 2.2.53)
% M is modal matrix whose columns are put in order of dominancy
% r is k´n aggregation matrix
[n,mx] = size(b);
m1 = m(1 :k,1 :k);
jord = inv(m)*a*m;p = [eye(k),O*ones(k,n-k)];
f=m1*p*jord*p'*inv(m1);g=m1*p*inv(m)*b;r=m1*p*inv(m);h=c*pinv(r); end
```

Once mod.m has been created and debugged within LSSTB and MATLAB, the following statements would provide three reduced-order models.

```
>> [f2,g2,h2,r2]=mod(a,b,c,df,m_ord,2)

f2 =
        -0.7520 - 0.0000i    0.2003 + 0.0000i
        -2.3036 - 0.0000i    0.0274 + 0.0000i
g2 =
        -1.1999 - 0.0000i
        -2.1071 + 0.0000i

h2
        1.0358 - 0.0000i   -1.2322 - 0.0000i
```

controlengineers.ir
@controlengineers

```
r2
Columns 1 through 4
0.9956 + 0.0000i  -0.0200   -0.1197 + 0.0000i   0.0017 - 0.0000i
0.0630 + 0.0000i   0.8940   -0.4375 + 0.0000i   0.0113 - 0.0000i

Columns 5 through 7
-2.0786 - 0.0000i     0.0223 - 0.0000i     0.0026 + 0.0000i
-1.7471 + 0.0000i     0.1179 + 0.0000i     0.0145 + 0.0000i

>>imz2 = impulse (f2,g2,h2,r2,1,t);
>>[f4,g4,h4,r4] = mod(a,b,c,df,m_ord,4)

f4 =
-0.8630 + 0.0000i   0.1980 - 0.0000i    0.2193 + 0.0000i  -0.2097 - 0.0000i
-1.4151 + 0.0000i  -0.0098 - 0.0000i    1.0967 + 0.0000i   0.2964 - 0.0000i
 0.0000 - 0.0000i  -0.0000 + 0.0000i   -5.0000 - 0.0000i   2.1200 + 0.0000i
 8.1811 - 0.0000i   0.9728 - 0.0000i  -43.5067 - 0.0000i  3.4310 + 0.0000i

g4 =
-0.0633 - 0.0000i
-4.0446 + 0.0000i
-9.6787 + 0.0000i
-5.3362 + 0.0000i

h4 -
1.1791 + 0.0000i -1.3725 - 0.0000i 0.1763 + 0.0000i 0.5019 - 0.0000i

r4 =
 Columns 1 through 4
  0.9934 + 0.0000i -0.0198 + 0.0000i  -0.1107 + 0.0000i  -0.0032 - 0.0000i
  0.0277 + 0.0000i  0.8944 + 0.0000i  -0.4238 + 0.0000i  -0.0293 - 0.0000i
 -0.4599 + 0.0000i  0.0337 - 0.0000i  -0.0578 - 0.0000i   0.0890 + 0.0000i
 -0.2389 + 0.0000i  0.0209 - 0.0000i  -0.5350 + 0.0000i   1.0485 - 0.0000i

Columns 5 through 7
-0.9487 - 0.0000i    0.0226 - 0.0000i    0.0027 + 0.0000i
-3.6628 + 0.0000i    0.1168 + 0.0000i    0.0143 + 0.0000i
-9.1596 + 0.0000i   -0.0102 - 0.0000i   -0.0014 - 0.0000i
-4.5601 + 0.0000i   -0.0093 - 0.0000i   -0.0021 - 0.0000i

>> imz4 = impulse (f4,g4,h4,r4,1,t);
>> [f5,g5,h5,r5]=mod(a,b,c,df,m_ord,5)

f5 =
1.OE + 002 *
```

controlengineers.ir
@controlengineers

```
Columns 1 through 4
-0.0117 - 0.0000i    0.0021 + 0.0000i  -0.0052 - 0.0000i  -0.0015 + 0.0000i
-0.0297 - 0.0000i    0.0007 + 0.0000i  -0.0262 - 0.0000i   0.0059 - 0.0000i
 0.0000 + 0.0000i  -0.0000 - 0.0000i  -0.0500 + 0.0000i   0.0212 + 0.0000i
-0.0000 - 0.0000i  -0.0000 - 0.0000i   0.0000 + 0.0000i  -0.0000 - 0.0000i
-0.0014 - 0.0000i    0.0000 - 0.0000i   0.0014 - 0.0000i  -0.0020 - 0.0000i

Column 5
-0.0645 - 0.0000i
-0.3224 - 0.0000i
 0.0000 + 0.0000i
 3.7700 + 0.0000i
 0.0028 + 0.0000i

g5 =
        1.5767 - 0.0000i
        2.8918 + 0.0000i
        0.9509 - 0.0000i
       -0.0155 + 0.0000i
        1.0054 + 0.0000i

h5 =
    Columns 1 through 4
    1.2694 - 0.0000i -0.9522 + 0.0000i 1.1939 - 0.0000i 0.9288 - 0.0000i
    Column 5
    1.6373 + 0.0000i

r5 =
    Columns 1 through 4
     1.0639  + 0.0000i  -0.0248 - 0.0000i    0.0520 +  0.0000i   -0.0169 - 0.0000i
      0.3256 + 0.0000i    0.8732 - 0.0000i    0.2645 + 0.0000i   -0.0870 - 0.0000i
    -0.0034 - 0.0000i    0.0012 - 0.0000i    0.9969 - 0.0000i    0.0006 - 0.0000i
    -0.0104 + 0.0000i    0.0046 - 0.0000i  -0.0071 + 0.0000i    1.0042 - 0.0000i
      0.0007 - 0.0000i  -0.0003 + 0.0000i    0.0005 + 0.0000i  -0.0002 + 0.0000i

    Columns 5 through 7
     0.4579 - 0.0000i       0.0210 - 0.0000i   0.0029 + 0.0000i
     2.2865 - 0.0000i       0.1227 + 0.0000i   0.0153 + 0.0000i
    -0.0427 - 0.0000i      -0.0011 + 0.0000i   0.0001 + 0.0000i
     0.0034 + 0.0000i      -0.0047 + 0.0000i  -0.0014 + 0.0000i
     1.0042 + 0.0000i       0.0003 - 0.0000i   0.0000 + 0.0000i

>> imz5 = impulse (f5,g5,h5,r5,1,t);
>> impz=impulse(a,b,c,d,1,t);
>> plot (t,imps,t,imz2,t,imz4,t,imz5),xlabel ('Time),ylabel ('yf & ya')
>> title('Impulse Responses - Full & 3 Modally Reduced Models # 2')
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

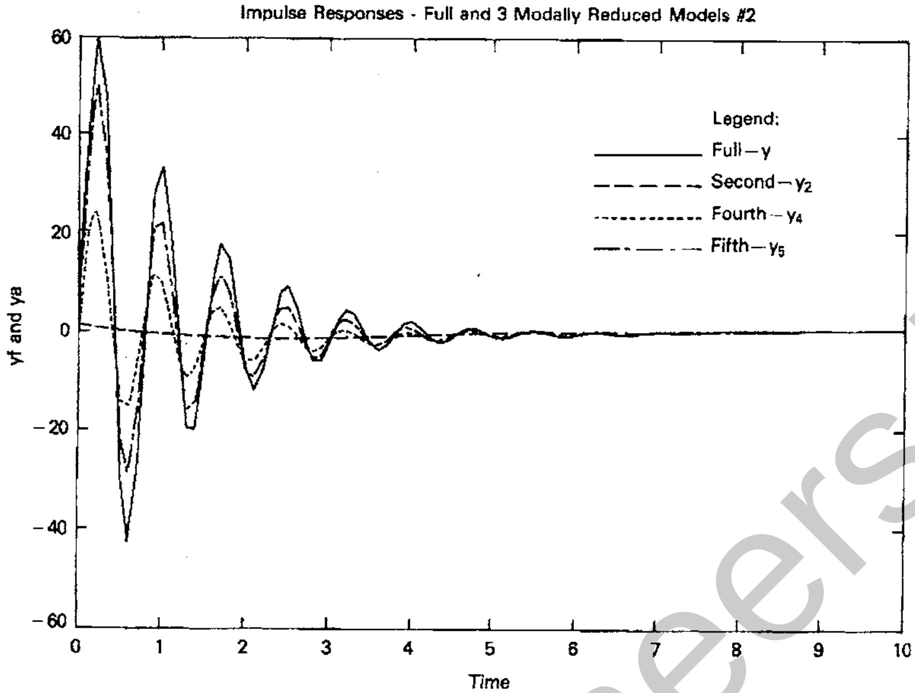


Figure 2.8 Step responses of a full-order model and three balanced reduced-order models.

The impulse responses of the full-order and three reduced models are shown in Figure 2.8. Note that the matrices are the same as before, but as expected, g and h matrices are not. Here again, the responses of reduced models and the full model match better when there are a sufficient number of modes retained. The best order of reduced models would be when a criterion such as minimum error response can be satisfied.

2.2.3 Balanced Aggregation

One of the main shortcomings of model reduction methods is the lack of a strong numerical tool to go with the well-developed theory. For example, the minimal realization theory of Kalman (Jamshidi *et al.*, 1992), offers a clear understanding of the internal structure of linear systems. The associated discussions on controllability, observability, and minimal realization often illustrate the points, but numerical algorithms are adequate only for low-order textbook examples. Furthermore, there has been little connection made between minimal realization, controllability and observability, and model reduction on the other hand. Moore (1981) proposed to use the *Principle Component Analysis* of statistics along with some algorithms for the computation of "singular value decomposition" of matrices to develop a model reduction scheme which makes the most controllable

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and observable modes of the system transparent. Under a certain matrix transformation, the system is said to be "balanced" and the most controllable and observable modes would become prime candidates for reduced-order model states.

Consider an asymptotically stable, controllable and observable linear time invariant system (A, B, C) defined by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.2.58a}$$

$$y(t) = Cx(t) \tag{2.2.58b}$$

where $x, u, y, A, B,$ and C are defined as before. The balanced matrix method is based on the simultaneous diagonalization of the positive definite controllability and observability Gramians of Eq. (2.2.58), (see also Sec. 3.4), that is,

$$G_c = \int_0^\infty e^{At} BB^T e^{A^T t} d \tag{2.2.59}$$

and

$$G_0 = \int_0^\infty e^{A^T t} C^T C e^{At} dt \tag{2.2.60}$$

The Gramian matrices G_c and G_o satisfy the following Lyapunov-type equations:

$$G_c A^T + AG_c + BB^T = 0 \tag{2.2.61}$$

$$G_0 A + A^T G_0 + C^T C = 0 \tag{2.2.62}$$

The balanced approach of model reduction is essentially the computation of a similar transformation matrix S such that both G_c and G_0 become equal and diagonal, that is, balanced. This transformation matrix is given by (Moore, 1981),

$$S = V\ D\ P\ \Sigma^{-1/2} \tag{2.2.63}$$

where orthogonal matrices V and P satisfy the following symmetric eigenvalue/eigenvector problems.

$$V^T G_c V = D^2 \tag{2.2.64}$$

and

$$P^T\left[(VD)^T G_0 (VD)\right]P = \Sigma^2 \tag{2.2.65}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\Sigma = S^T G_0 S = S^{-1} G_c \left(S^{-1}\right)^T$$
$$= diag\left(\sigma_1, \sigma_2, .., \sigma_n\right) \tag{2.2.66}$$

Here D is a diagonal matrix like Σ. The diagonal elements of Σ have the property that $\sigma_1 \geq \sigma_2 ,..\geq \sigma_n > 0$ and are called *second-order modes* of the system by Moore (1981). Using the transformation $\hat{x} = S^{-1}x$, one obtains the following full-order equivalent system,

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \tag{2.2.67}$$

$$y = \hat{C}\hat{x} \tag{2.2.68}$$

where

$$\hat{A} = S^{-1}A\,S, \quad \hat{B} = S^{-1}B, \quad \hat{C} = C\,S \tag{2.2.69}$$

Now, if $\sigma_r >> \sigma_r+1$ for a given r, and internally dominant reduced-order model of order r can be obtained from Eqs. (2.2.67) and (2.2.68) by

$$\dot{z} = Fz + Gu \tag{2.2.70}$$

$$y = Hz \tag{2.2.71}$$

where (F, G, K) matrices are represented by the following partitioned matrices

$$\hat{A} = \left[\begin{array}{c|c} F & \hat{A}_{12} \\ \hline \hat{A}_{21} & \hat{A}_{22} \end{array}\right], \quad \hat{B} = \left[\begin{array}{c} G \\ \hline \hat{B}_2 \end{array}\right], \quad \hat{C} = \left[H | \hat{C}_2\right] \tag{2.2.72}$$

Although this partitioning of second-order models leading to a reduced and a residual model are somewhat arbitrary, grouping the most controllable and observable modes together does, as it is seen later, provide a reasonable criterion for model reduction.

Laub (1980) has proposed a more efficient method to compute the balancing transformation matrix given by

$$S = L_c U \Sigma^{-1/2} \tag{2.2.73}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where L_c is a lower triangular of the Cholesky decomposition (Moler, 1980) of the controllability Gramian G_c, U is an orthogonal modal matrix, and Σ is the diagonal matrix to the symmetric eigenvalue/eigenvector problem of

$$U^T\left(L_c^T G_o L_c\right)U = \Sigma^2 \tag{2.2.74}$$

To show that G_o and G_c are diagonalized and equal, we note that

$$\hat{G}_0 = S^T G_0 S \tag{2.2.75}$$

$$\hat{G}_c = S^{-1} G_c S \tag{2.2.76}$$

It can be easily verified (see 2.2.26), that $\hat{G}_c = \hat{G}_0 = \Sigma$.

Laub *et al.* (1980) have presented an algorithm for computing a state-space balancing transformation directly from a state-space realization. This algorithm is computationally much more efficient than the one reported earlier by Laub (1980). One difference is that it avoids the so-called "squaring up" problem, that is, neither BB^T nor C^TC in Equations (2.2.61) and (2.2.62), respectively, need to be formed explicitly. The new algorithm is given below.

Algorithm 2.1

1. Compute the Cholesky factors of the Gramians. Let L_c and L_o denote the lower triangular Cholesky factors of G_c and G_o, that is,

$$G_c = L_c L_c^T \qquad G_0 = L_0 L_0^T \tag{2.2.77}$$

This step is accomplished using an algorithm by Hammarling (1982) which would avoid the formation of G_c or G_o themselves.

2. Compute the singular value decomposition of the product of the Cholesky factors; that is,

$$L_0^T L_c = V \Sigma U^T \tag{2.2.78}$$

3. Form the balancing transformation

$$S = L_c U \Sigma^{-1/2} \tag{2.2.79}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

it is noted that

$$S^{-1} = \Sigma^{-1/2} V^T L_0^T \tag{2.2.80}$$

4. Form the balanced state-space matrices

$$\hat{A} = S^{-1} A S = \Sigma^{-1/2} V^T L_0^T A L_c U \Sigma^{-1/2} \tag{2.2.81}$$

$$\hat{B} = S^{-1} B = \Sigma^{-1/2} V^T L_0^T B \tag{2.2.82}$$

$$\hat{C} = C S = C L_c U \Sigma^{-1/2} \tag{2.2.83}$$

There are two major differences between this algorithm and Laub's (1980) earlier one. One is, as mentioned before, the elimination of the "squaring up" problem in finding G_c and G_o in Eqs. (2.2.61) and (2.2.62). The other is the difference between singular-value decomposition algorithm of step 2, that is, Eq. (2.2.78) versus the solution of the symmetric eigenvalue/eigenvector problem of Eq. (2.2.74). For further details on this scheme of model reduction, refer to Laub *et al.* (1980).

Example 2.2.6. Consider a fourth-order SISO system,

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 & -150 \\ 1 & 0 & 0 & -245 \\ 0 & 1 & 0 & -113 \\ 0 & 0 & 1 & -19 \end{bmatrix} x + \begin{bmatrix} 4 \\ 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x$$

Find a reduced-order model using the balanced method.

SOLUTION: The controllability and observability Gramians are obtained by solving Eqs. (2.2.61) and (2.2.62), that is,

$$G_c = \begin{bmatrix} 13.5 & 6.8 & 1.05 & 0.05 \\ 6.8 & 3.6 & 0.58 & 0.03 \\ 1.05 & 0.58 & 0.098 & 0.005 \\ 0.05 & 0.03 & 0.005 & 0.0003 \end{bmatrix}$$

$$G_o = \begin{bmatrix} 6.7 & 10.9 & 4.5 & 0 \\ 10.9 & 18.4 & 8.2 & 0.045 \\ 4.5 & 8.2 & 4.75 & 0.073 \\ 0 & 0.045 & 0.073 & 0.03 \end{bmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The transformation matrix S is given by

$$S = \begin{bmatrix} 29.1 & -4.0 & 0.55 & -0.31 \\ 14.8 & 5.4 & -0.55 & 0.42 \\ 2.3 & 2.1 & -0.03 & -0.12 \\ 0.12 & 0.13 & 0.05 & 0.007 \end{bmatrix}$$

The transformed state-space matrices are given by

$$\hat{A} = \begin{bmatrix} -0.44 & -1.17 & -0.41 & -0.05 \\ 1.17 & -3.13 & -2.83 & -0.33 \\ -0.41 & 2.83 & -12.5 & -3.2 \\ 0.05 & -0.33 & 3.2 & -2.95 \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} 0.12 \\ -0.13 \\ 0.05 \\ -0.007 \end{bmatrix}, \hat{C} = \begin{bmatrix} 0.12 & 0.13 & 0.05 & 0.007 \end{bmatrix}$$

One can now extract a third-, second-, or a first-order reduced model. The best reduced-order model is subject to other research (Jamshidi, 1984), but through the relative magnitudes of second-order modes $\Sigma^2 = \text{diag}\,(0.016, 0.003, 0.001, 0.0)$, one may often find an appropriate order for the reduced model. For example, here it is anticipated that a second-order reduced model is fairly close to the full model. This is also evident by a comparison of the step responses of the full-order model and the first, second, and third reduced models, shown in Figure 2.9.

The balanced realization and its reduced-order modeling have been programmed into a command in most CAD packages, including MATLAB and LSSTB.

However, one can use commands such as LYAP or GRAM, CHOL, SVD, and so on, to write one's own set of commands to find a balanced state-space formulation of a linear time-invariant system. Moreover, the extension of the balanced realization to unstable systems as well as to the discrete-time case can be easily accomplished. For the former, refer to Santiago and Jamshidi (1986) and Problems 2.27 and 2.28; for the latter, refer to the work of Laub *et al.* (1980).

CAD Example 2.3. Using MATLAB, the fourth-order system of Example 2.2.6 is reduced to a lower order (1, 2 or 3) model. The impulse responses of both full- and reduced-order models will be compared.

```
>> a = [0 0 0 –150; 1 0 0 –245; 0 1 0 –113; 0 0 1 –191;
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

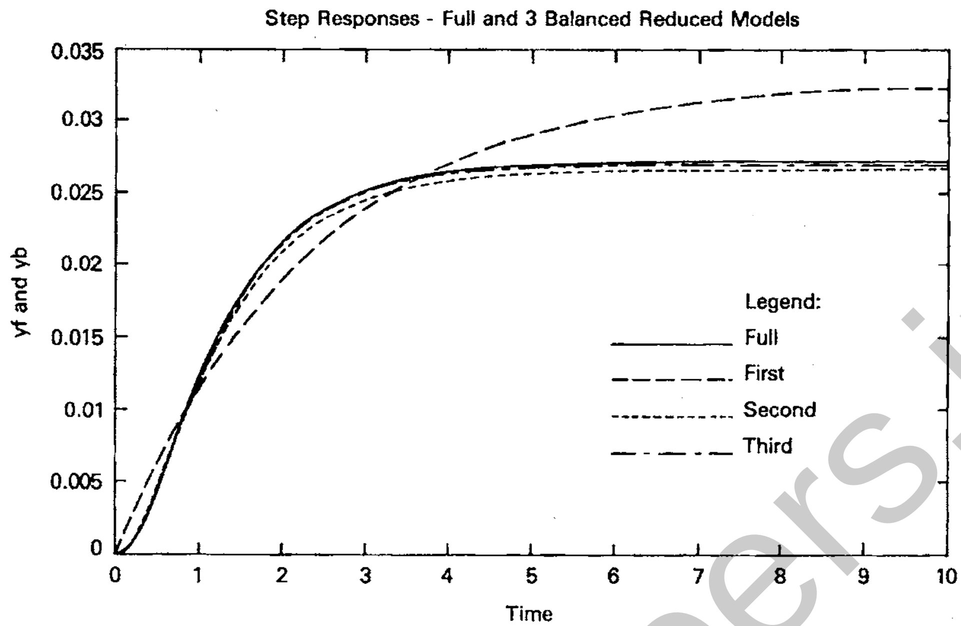
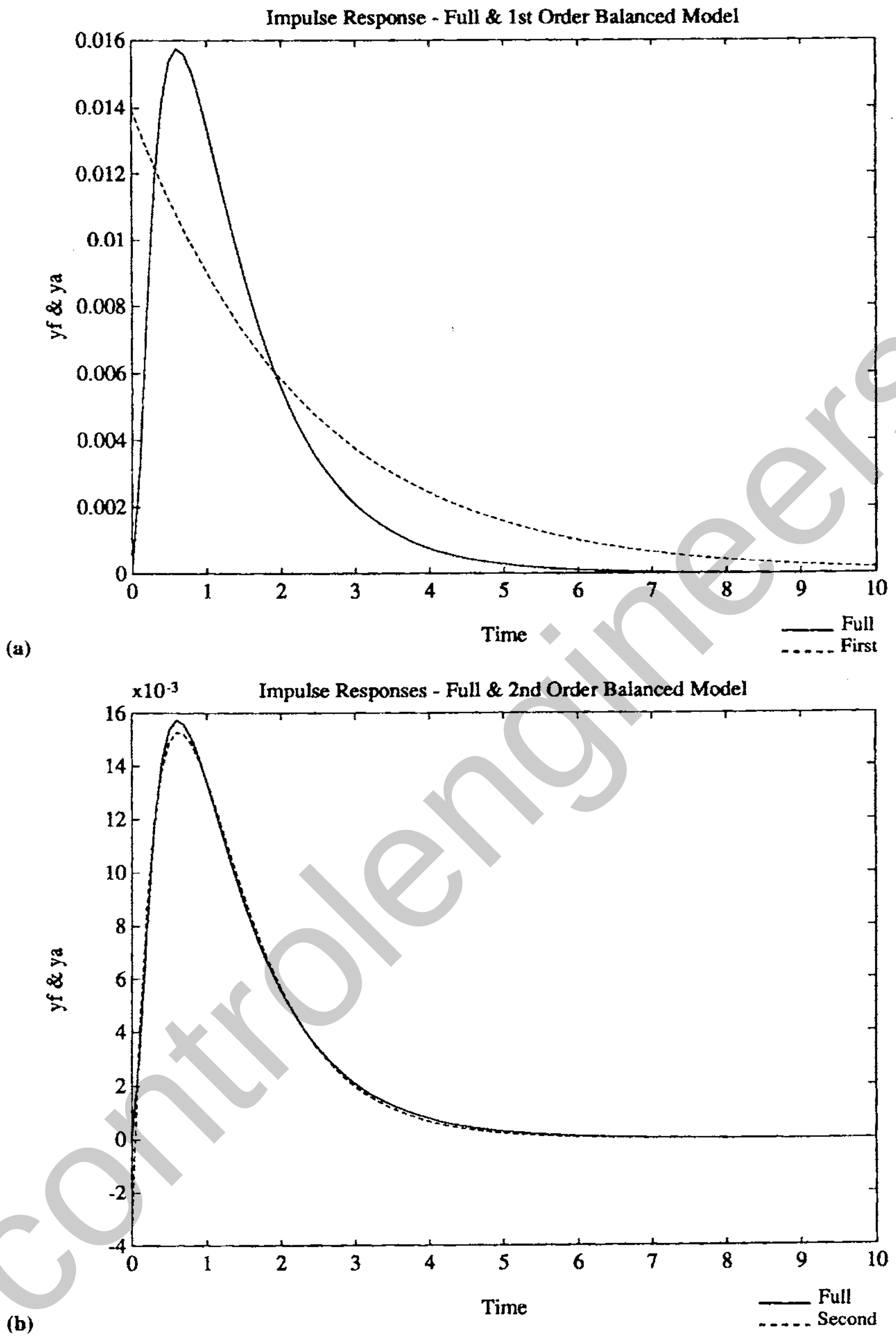


Figure 2.9 Step responses of a full-order model and three balanced reduced-order models for Example 2.2.6.

```
>> b = [4; 1; 0; 0]; c = [0 0 0 1]; d = 0;
>> [ab,bb,cb,sig,tr] = balreal (a,b,c);
>> % second-order reduced model
>> f1 = ab(1:1, 1:1); g1 = bb(1:1,:); h1 = cb(:,1:1);
>> f2 = ab(1 :2, 1 :2), g2 = bb(1 :2,:), h2 = cb(:,1 :2)

f2 =
         -0.4378   -1.1681
          1.1681   -3.1353

g2 =
          0.1179
         -0.1304

h2 =
         0.1183    0.1310

>> t = [0: 0.1: 10];
>> % impulse response comparisons
>> im = impulse (a,b,c,d,1.t);
>> ial = impulse (f1,g1,h1,d,1,t);
>> ia2 = impulse (f2,g2,h2,d,1,t);
```

controlengineers.ir

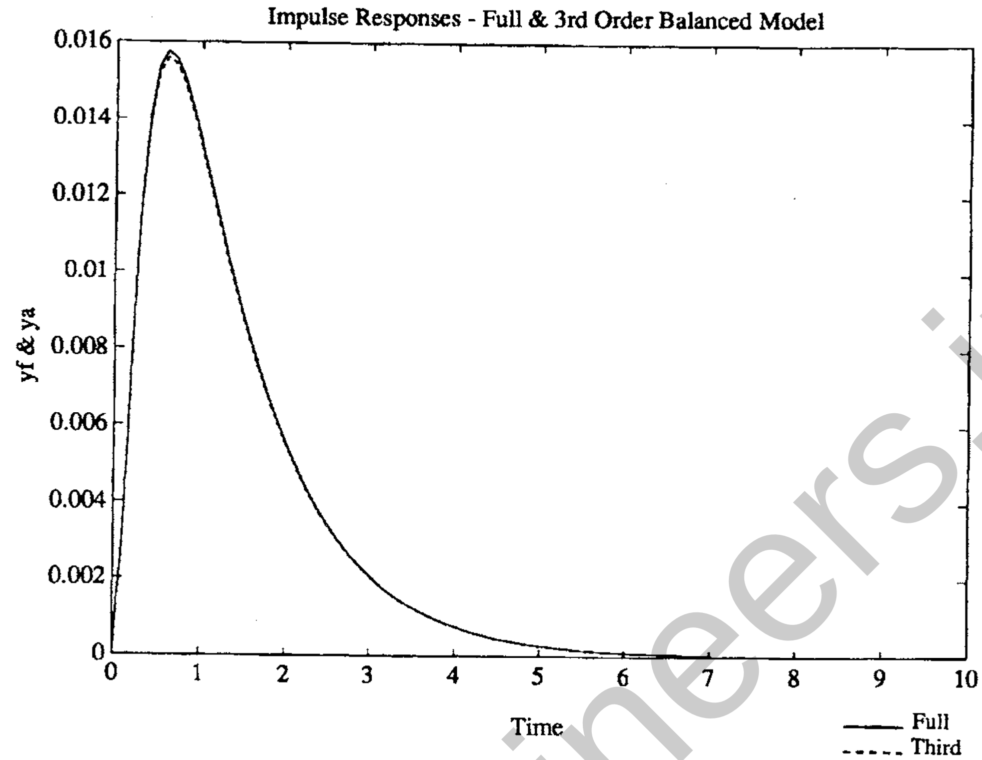
اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



controlengineers.ir

Figure 2.10 Impulse responses of the full model and the balanced models for:
(a) First-order balanced model, CAD Example 2.3 (LSSTB);
(b) Second-order; and
(c) Third-order balanced model.

controlengineers.ir
@controlengineers



(c)

Firgure 2.10 (Continued)

```
>> ia3 = impulse (f3,g3,h3,d,1,t);
>> Plot (t, im, t, ia1)
>> Plot (t, im, t, ia2)
>> Plot (t, im, t, ia3)
```

The impulse responses of the full-order model with three successive reduced-order models are shown in Figure 2.10. It is noted that the impulse response of the first-order reduced model is somewhat far apart from the fourth-order full model. However, the second-order response is fairly close, while the third-order reduced model has a very close and indistinguishable response. This characteristic is also clear from the relative magnitude of Moore's third and fourth second-order modes, that is, d_{33} and d_{44} in matrix D^2.

Before we turn to the other important approach for large-scale systems modeling, i.e., "perturbation," a few comments on the works of authors in regard to aggregation through another method based on error minimization are due. One of the earliest attempts in aggregating linear time-invariant systems via this criterion is due to Meier and Luenberger (1967), who have synthesized reduced-order transfer functions for SISO (Single Input Single Output) systems by a minimization of the mean-square error between outputs of full and aggregated systems.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The necessary conditions for this minimization process, similar to another one by Wilson (1970), lead to a set of nonlinear equations for pole-zero locations for aggregated system transfer functions. These equations are solved by an iterative scheme such as the Newton or Gradient scheme. Another approach, due to Anderson (1967), fits the output data from a large-scale discrete-time system to a reduced-order model. Still another effort is due to Sinha and Berezani (1971), who use a "pattern-search algorithm" to minimize the sum of output error norms at different time instances raised to a prespecified power. Galiana (1973) extended the fitting methods of Anderson (1967) to multivariable systems. The criterion is a weighted quadratic error, and the necessary conditions are nonlinear matrix equations which are solved by iterative methods.

A possible formulation of these schemes is to obtain the matrices (F, G, D) of

$$\dot{z}(t) = Fz(t) + Gu(t) \tag{2.2.84a}$$

$$\hat{y}(t) = Dz(t) \tag{2.2.84b}$$

by minimizing a quadratic function of the reduction error

$$J = \int_0^{\alpha} e^T(t) Q e(t) dt \tag{2.2.85}$$

where Q is an $r \times r$ symmetric positive-definite matrix (Mahmoud and Singh, 1981). Although all the error minimizations are technically sound, the extraction of an aggregated model requires the solution of a set of rather complicated nonlinear matrix equations which cannot always be used for large-scale systems.

Further readings on this topic may be obtained from Wilson (1974), Wilson and Mishra (1979), Siret *et al.* (1977b), Riggs and Edgar (1974), Edgar (1975), Hirzinger and Kreisselmeier (1975), Nagarajian (1971), Mishra and Wilson (1980), Mahmoud and Singh (1981). Discussion and comments on the aggregation methods are given in Section 2.7.

2.3 Perturbation Methods

The basic concept behind perturbation methods is the approximation of a system's structure through neglecting certain interactions within the model which leads to lower order. From a large-scale system modeling point of view, perturbation methods can be considered as approximate aggregation techniques.

There are two basic classes of perturbations applicable for large-scale

controlengineers.ir

controlengineers.ir
@controlengineers

systems modeling purposes: *weakly coupled* models and *strongly coupled* models. This classification is not universally accepted, but a great number of authors have adapted it; others refer to them as *nonsingular* and *singular* perturbations.

2.3.1 Weakly Coupled Models

In many industrial control systems certain dynamic interactions are neglected to reduce computational burdens for system analysis, design, or both. Examples of such practice are in chemical process control and space guidance (Kokotović, 1972), where different subsystems are designed for flow, pressure, and temperature control in an otherwise coupled process or for each axis of a three-axis attitude control system. However, the computational advantages obtained by neglecting weakly coupled subsystems are offset by a loss in overall system performance. In this section the weakly coupled models for large-scale linear systems are introduced. Nonlinear large-scale systems are considered, in part, in Chapter 6, where near-optimum control of these systems is discussed. Consider the following largescale system split into k linear subsystems,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} A_1 & \varepsilon A_{12} & \cdots & \varepsilon A_{1k} \\ \varepsilon A_{21} & A_2 & \varepsilon A_{23} & \varepsilon A_{2k} \\ \vdots & & & \vdots \\ & & & A_k \end{bmatrix}$$

$$+ \begin{bmatrix} B_1 & \varepsilon B_{12} & \cdots & \\ \varepsilon B_{21} & B_2 & & \\ \vdots & & & \vdots \\ & & & B_k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \tag{2.3.1}$$

where ε is a small positive coupling parameter, x_i and u_i are ith subsystem state and control vectors, respectively, and all A and B matrices are assumed to be constant. A special case of (2.3.1), when $k = 2$, has been called the ε-*coupled* system by Kokotović *et al.* (1969, 1972), i.e.,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & \varepsilon A_{12} \\ \varepsilon A_{21} & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 & \varepsilon B_{12} \\ \varepsilon B_{21} & B_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.3.2}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

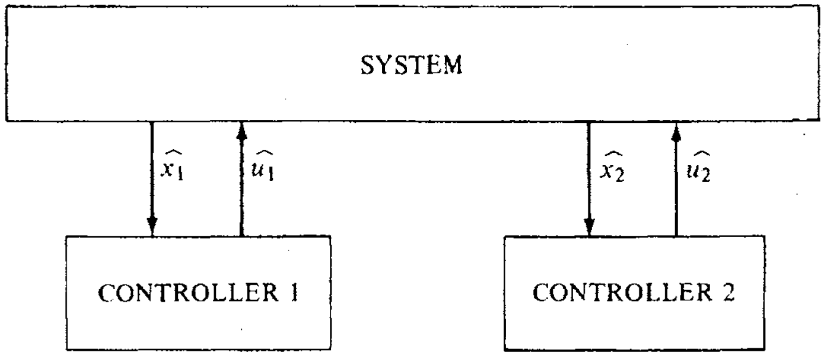


Figure 2.11 A decentralized control structure for two weakly coupled subsystems.

It is clear that when $\varepsilon = 0$ the above system decouples into two subsystems,

$$\begin{aligned}\dot{\hat{x}}_1 &= A_1\hat{x}_1 + B_1\hat{u}_1\\ \dot{\hat{x}}_2 &= A_2\hat{x}_2 + B_2\hat{u}_2\end{aligned} \tag{2.3.3}$$

which correspond to two approximate aggregated models (Jamshidi, 1983), one for each subsystem. In this way the computation associated with simulation, design, etc., will be reduced drastically, especially for large-scale system order n and k larger than two subsystems. In view of the decentralized structure of large-scale systems briefly introduced in Chapter 1, these two subsystems can be designed separately in a decentralized fashion shown in Figure 2.11.

Example 2.3.1. In this example a 17th-order linear system representing a simplified model of a three-stand cold rolling mill considered by Jamshidi (1972, 1983) is considered. See Figure 2.12.

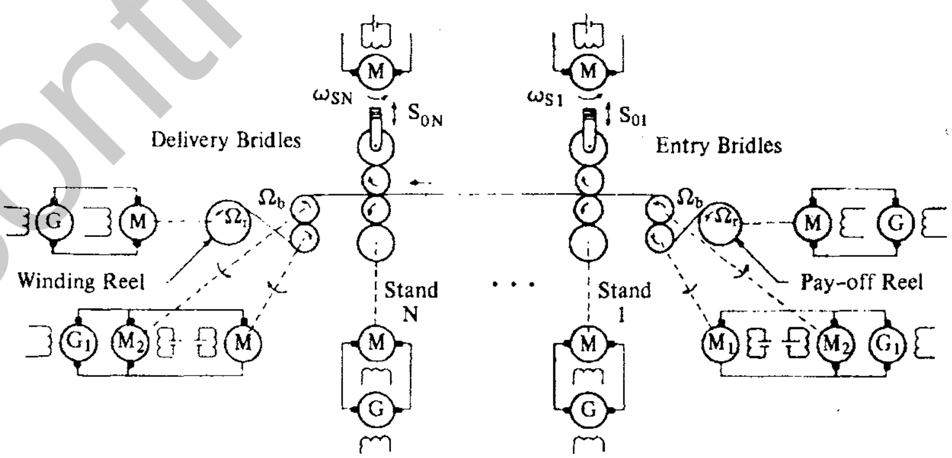


Figure 2.12 An N-stand cold rolling mill.

controlengineers.ir

$$A(r)=\left[\begin{array}{cccccc|cccccc|ccccc}
-0.318 & & & & & & & & & & & & & & & & \\
 & -0.318 & & & & & & & & & & & & & & & \\
 & & -0.33 & & & & & & & & & & & & & & \\
 & & & -0.11 & & -0.532 & & & & 0 & & & & & 0 & & \\
a_{51} & & a_{53} & & a_{55} & a_{56} & & & & & & & & & & & \\
 & & & 23.4 & a_{65} & -0.805 & -0.805 & & & & & & & & & & \\
\hline
 & & & & & -7.65 & -19.0 & & & & & & & & & & \\
 & & & & & 0.173 & & -0.1195 & & 0.0224 & & & & & & & \\
 & & & & & & -3.95 & -19.0 & & & & & & & & & \\
 & & 0 & & & & 0.041 & & -0.0111 & & & & & 0 & & -0.0216 & \\
 & & & & & & & & -4.3 & -18.8 & & & & & & & 6.53 \\
 & & & & & & & & & & & -0.318 & & & & & \\
\hline
 & & & & & & & & & & & & -0.318 & & & & \\
 & & & & & & & & & & & & & -0.33 & & & \\
 & & 0 & & & & & & & 0 & & 2.21 & & & -0.11 & & -0.205 \\
 & & & & & & & & & & & a_{16,12} & & a_{16,14} & & a_{16,16} & a_{16,17} \\
 & & & & & & & & & & -0.973 & & & & -0.19 & a_{17,16} & -0.973
\end{array}\right] \tag{2.3.5a}$$

where $a_{51}(r)=(N_1(r))/D_1(r)$, $a_{53}(r)=0.1a_{51}(r)$, $a_{55}(r)=-0.7a_{51}(r)$, $a_{56}(r)=(N_2(r))/D_1(r)$, $a_{65}(r)=6.6r$, $a_{16,12}(r)=(N_1(r))/D_2(r)$, $a_{16,14}(r)=0.1a_{16,12}(r)$, $a_{16,17}(r)=(N_2(r))/D_2(r)$, and $a_{17,16}(r)=7.96r$ with $N_1(r)=1.98(r+0.152r^3)$, $D_1(r)=1+0.0185r^4$, $N_2(r)=-8.52r$, $D_2(r)=1+0.0483r^4$, and

$$B(r)=\left[\begin{array}{c|c|c}
\begin{matrix}0.318 & & \\ & -0.318 & \\ & & b_{33}(r)\end{matrix} & 0 & 0 \\ \hline
0 & \begin{matrix}-1.22 & -1.22 & & & & \\ & & -0.826 & -0.826 & & \\ & & & & -0.424 & -0.424\end{matrix} & 0 \\ \hline
0 & 0 & \begin{matrix}0.318 & & \\ & 0.318 & \\ & & b_{14,12}(r) \\ & & 0 \\ & & 0 \\ & & 0\end{matrix}
\end{array}\right] \tag{2.3.5b}$$

where $b_{33}(r)=0.105/(r(1-5.25r^2))$, $b_{14,12}(r)=0.105/(r(1-8.78r^2))$.

controlengineers.ir
@controlengineers

The corresponding state equation is

$$\dot{x}(t) = A(r)x(t) + B(r)u(t) \tag{2.3.4}$$

where r is the payoff (winding) reel radius considered to be a slow-varying parameter. It is desired to decouple the system.

SOLUTION: The details of this system will be given in Chapter 6; however, for the sake of present example, the $A(r)$ and $B(r)$ matrices are given below for a 3×3 partitioned form. The first six variables represent the dominant modes of the winding reel (coiler); the next six variables describe the dominant modes of the three-stands; and the last five variables belong to the payoff reel (decoiler) subsystems. In the formulation shown most, off-diagonal submatrices are highly sparse. The $A(r)$ and $B(r)$ matrices for the three-stand cold rolling mill are shown in (2.3.5a)–(2.3.5b).

The research regarding weakly coupled systems has taken two main lines. The first line is to set $\varepsilon = 0$ in (2.3.2) and tries to find a quantitative measure of the resulting approximation when in fact $\varepsilon \neq 0$ in actual condition. Bailey and Ramapriyan (1973) have provided conditions which would give an estimation on the loss in the optimal performance in a linear state regulator formulation of (2.3.2). Furthermore, they have presented conditions for a criterion of a weak coupling, a task similar to that of Milne (1965), whose results will be presented here. The loss of optimal performance for various large-scale systems control laws is discussed in Chapter 6. In Chapter 4, a formal treatment of the effects of subsystems interactions (i.e., $\varepsilon \neq 0$) on the overall performance under the context of hierarchical control will be presented. Pérez-Arriaga *et al.* (1981) have proposed a so-called "selective modal analysis" procedure for the separation of "relevant" (not necessarily slow) and "less relevant" (not necessarily fast) parts of the system. Although they have presented a separation criterion based on a " participation factor" and an algorithm to implement it, relatively little computational results and widespread applications of their procedure are available at this time.

Consider a coupled A matrix as presented in (2.3.2) and assume that A_1, A_{12}, A_{21}, and A_2 are $n_1 \times n_1$, $n_1 \times n_2$, $n_2 \times n_1$, and $n_2 \times n_2$, respectively, with $n = n_1 + n_2$ being the order of the original large-scale coupled system. Furthermore, let

$$\begin{aligned}
&\lambda_i\{A_1\} = \{\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{n1}\}, \quad i = 1, 2, \ldots, n_1 \\
&\lambda_j\{A_2\} = \{\hat{\lambda}_{n_1+1}, \ldots, \hat{\lambda}_n\}, \quad j = n_1 + 1, \ldots, n \\
&\lambda_k\{A\} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}, \quad k = 1, 2, \ldots, n
\end{aligned} \tag{2.3.6}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

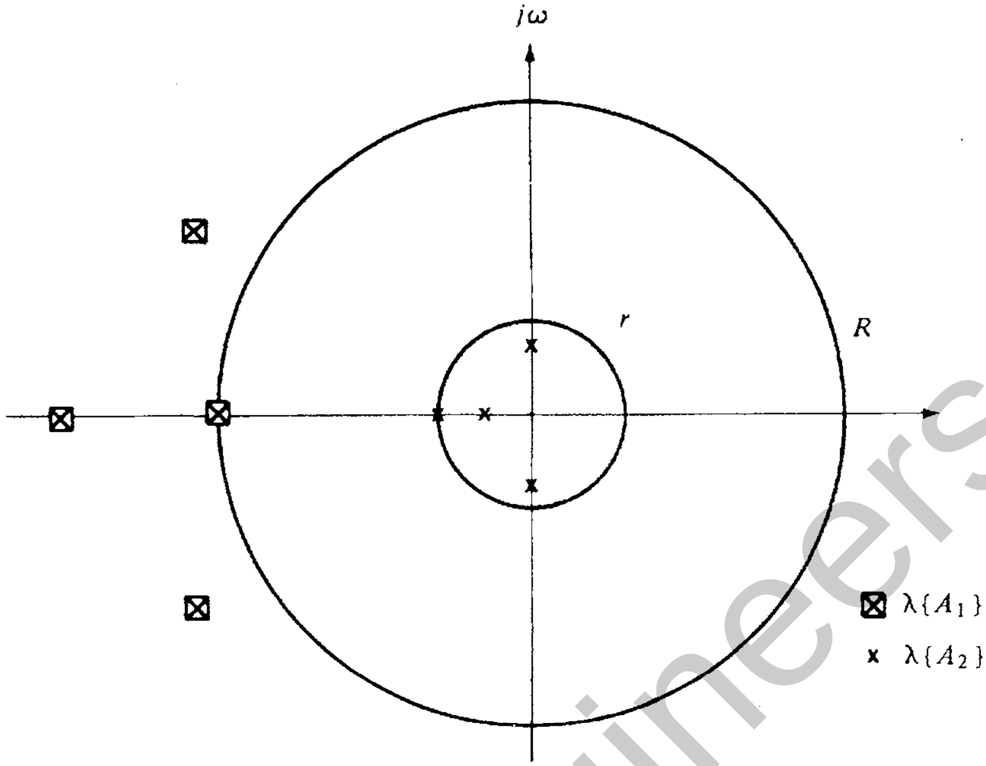


Figure 2.13 Relative locations of the eigenvalue of A_1 and A_2 submatrices in a weakly coupled system.

be the eigenvalues of diagonal submatrices A_1, A_2, and matrix A. Let us postulate that the moduli of the eigenvalues of A_1 and A_2 are widely separated from each other (Milne, 1965). Without any loss of generality one can take $|\lambda_i\{A_1\}| << |\lambda_j\{A_2\}|$. Let the eigenvalues of A_1 be on or inside a circle with radius $r = \max|\lambda_i\{A_1\}|$, $i = 1,2,\ldots,n_1$ and the eigenvalues of A_2 be on or outside a circle with radius $R = \min|\lambda_j\{A_2\}|$, $j = n_1+1, n_1+2,\ldots,n$, as shown in Figure 2.13. If the following conditions are satisfied, then the system is said to be weakly coupled (Aoki, 1971):

$$(r/R) << 1 \tag{2.3.7a}$$

$$(n_1\varepsilon_{12}\varepsilon_{21})/R^2 << 1 \tag{2.3.7b}$$

where $\varepsilon_{12} = \max|(A_{12})_{i,j}|$ and $\varepsilon_{21} = \max|(A_{21})_{k,l}|$ for $i, j = 1, 2, \ldots, n_1$ and $k,l = 1, 2, \ldots, n_2$. The term (r/R) is called the *separation ratio*, ε_{12} and ε_{21} represent the maximum of the moduli of the elements of the A_{12} and A_{21} submatrices, respectively.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Example 2.3.2. Consider a sixth-order system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.2 & 0.05 & 0 \\ -400 & -170 & -23 & 0.05 & -0.02 & 0 \\ 0 & -0.01 & -0.02 & 0 & 1 & 0 \\ 0 & 0.01 & -0.015 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.04 & -0.53 & -1 \end{bmatrix} \tag{2.3.8}$$

We wish to check whether it is weakly coupled.

SOLUTION: The eigenvalues of A are $-9.94, -8.08, -4.97, -0.0816, -0.7924$, and -0.52, indicating that the first three are much farther away from the $j\omega$-axis than the last three which can be considered as dominant. For two 3×3 diagonal submatrices, it can be seen that the submatrices A_1 and A_2 are both in companion forms with the following eigenvalues:

$$\lambda_i\{A_1\} = \{-5, -10, -8\}, \quad \lambda_j\{A_2\} = \{-0.1, -0.8, -0.5\} \tag{2.3.9}$$

implying that $r = 0.8$, $R = 5$, and $(r/R) = 0.16$, which is much smaller than 1; hence condition (2.3.7a) holds. The values of ε_{12} and ε_{21} are 0.2 and 0.02, respectively, and the quantity $(n_1\varepsilon_{12}\varepsilon_{21})/R^2 = 0.00048 << 1$. Therefore, it is concluded that system (2.3.8) is weakly coupled.

The second line of research regarding weakly coupled systems has been to exploit such a system in an algorithmic fashion to find an approximate optimal feedback gain through a MacLaurin's series expansion of the accompanying Riccati matrix in the coupling parameter ε. It has been shown that retaining k terms of the Riccati matrix expansion would give an approximation of order $2k$ to the optimal cost (Kokotović *et al.*, 1969a,b). In Chapter 6 this approximate solution of the Riccati matrix will be used for near-optimum design of large-scale systems. The remainder of this section is devoted to strong coupling of large-scale systems.

2.3.2 Strongly Coupled Models

Strongly coupled systems are those whose variables have widely distinct speeds. The models of such systems are based on the concept of "singular perturbation," which differs from the regular perturbation (weakly coupled systems) in that perturbation is to the left of the system's state equation, i.e., a small parameter multiplying the time derivative of the state vector.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

In practice many systems, most of them large in dimension, possess fast changing variables displaying a singularly perturbed characteristic. A few examples were given before; in others, like power systems, the frequency and voltage transients vary from a few seconds in generator regulators, shaft stored energy, and speed governor motion, to several minutes in prime mover motion, stored thermal energy, and load voltage regulators (Kokotović, 1979). Similar time-scale properties prevail in many other practical systems and processes, such as industrial control systems, e.g., cold rolling mills as in Figure 2.12 (Jamshidi, 1974, 1983); biochemical processes (Heineken *et al.*, 1967); nuclear reactors (Kelley and Edelbaum, 1970); aircraft and rocket systems (Asatani *et al.*, 1971; Ardema, 1974); and chemical diffusion reactions (Cohen, 1974).

$$\dot{x}(t) = A_1 x(t) + A_{12} z(t) + B_1 u(t), \quad x(t_0) = x_0 \tag{2.3.10}$$

$$\varepsilon \dot{z}(t) = A_{21} x(t) + A_2 z(t) + B_2 u(t), \quad z(t_0) = z_0 \tag{2.3.11}$$

If A_2 is nonsingular, as $\varepsilon \to 0$, (2.3.10) and (2.3.11) become

$$\dot{\hat{x}}(t) = \left(A_1 - A_{12} A_2^{-1} A_{21}\right)\hat{x} + \left(B_1 - A_{12} A_2^{-1} B_2\right)\hat{u} \tag{2.3.12}$$

$$\hat{z}(t) = -A_2^{-1} A_{21} \hat{x} - A_2^{-1} B_2 \hat{u} \tag{2.3.13}$$

Equation (2.3.12) is an approximate aggregated model for (2.3.10)-(2.3.11), which in effect means that the n eigenvalues of the original system are approximated by the l eigenvalues of the $(A_1\text{-}A_{12}A_2^{-1}A_{21})$ matrix in (2.3.12). This observation follows the same line of argument when discussing conditions for weakly coupled systems considered by Milne (1965), Aoki (1971, 1978), and Bailey and Ramapriyan (1973). The following example illustrates a singularly perturbed system.

Example 2.3.3. Consider a classical third-order position control servosystem shown in block diagram form in Figure 2.14. Find its corresponding state-space model.

SOLUTION: This system has three states, $x_1 = \theta$, $x_2 = \dot{\theta} = \omega$, and $x_3 = i_a$. The corresponding time-domain equations for this system can be written by inspection,

$$\dot{\theta} = \omega, \quad \dot{\omega} + P_m \omega = k i_a, \quad \tau_a di_a/dt + i_a = Ke - k_b \omega, \\ e = \theta_i - \theta \tag{2.3.14}$$

controlengineers.ir @controlengineers

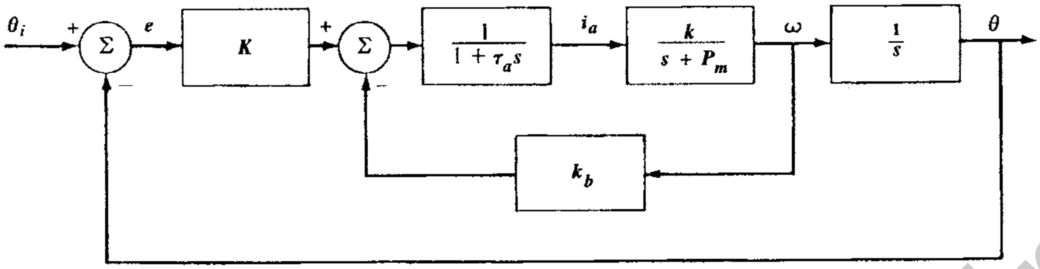


Figure 2.14 A position control servosystem.

where $\theta =$ angular position, $\dot{\theta} = \omega =$ angular velocity, $k =$ motor time constant, $P_m =$ motor pole (reciprocal of time constant), $i_a =$ armature current, $\tau_a = L_a/R_a$ is the armature circuit time constant, $k_b =$ back emf constant, $K =$ amplifier gain, and θ_i input position. Letting $u = \theta_i$ and the defined states, the three state equations from (2.3.14) are

$$
\begin{aligned}
\dot{x}_1 &= \dot{\theta} = \omega = x_2 \\
\dot{x}_2 &= \dot{\omega} = kx_3 - P_m x_2 \\
\tau_a \dot{x}_3 &= Ku - Kx_1 - k_b x_2 - x_3
\end{aligned}
\tag{2.3.15}
$$

It is known that the armature circuit inductance L_a is of the order of a few millihenries, while the armature resistance is of the order of a few ohms. Thus the armature current varies much faster than mechanical variables such as motor speed and position. If we let $\varepsilon = \tau_a$ and $z = x_3$, then (2.3.15) will become

$$
\begin{aligned}
\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & -P_m \end{bmatrix} x + \begin{bmatrix} 0 \\ k \end{bmatrix} z \triangleq A_1 x + A_{12} z \\
\varepsilon \dot{z} &= \begin{bmatrix} -K & -k_b \end{bmatrix} x - z + Ku \triangleq A_{21} x + A_2 z + B_2 u
\end{aligned}
\tag{2.3.16}
$$

which is in a singularly perturbed form. Therefore the usual assumption of ignoring the armature circuit dynamic in a DC motor is in fact a singular perturbation of the original third-order system to a second-order reduced model:

$$\dot{x} = \left(A_1 - A_{12} A_2^{-1} A_{21}\right)\hat{x} - A_{12} A_2^{-1} B_2 u \tag{2.3.17}$$

$$\hat{z} = -A_2^{-1}\left(A_{21}\hat{x} + B_2 u\right) \tag{2.3.18}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.3.2.a Boundary Layer Correction

It is noted that in going from (2.3.10)–(2.3.11) to (2.3.12) the initial condition z_o of $z(t)$ is lost and the values of $\hat{z}(t_o)$ and $z(t_o)=z_o$ are in general different; the difference is termed a left-side "boundary layer," which corresponds to the fast transients of (2.3.10)–(2.3.11) (Kokotović *et al.*, 1976). To investigate this phenomenon, which in effect explains under what conditions $\hat{x}$ and $\hat{z}$ approximate x and z, let u be zero in (2.3.10)–(2.3.11) and let the error between z and $\hat{z}$ be defined by

$$\eta(t)=z(t)-\hat{z}(t)=z(t)+A_2^{-1}A_{21}\hat{x}(t) \tag{2.3.19}$$

and choose a matrix $E_1(\varepsilon)$ so that when

$$\eta(t)=z(t)+A_2^{-1}A_{21}x(t)+\varepsilon E_1(\varepsilon)x(t) \tag{2.3.20}$$

is substituted in (2.3.10) and (2.3.11) with $u = 0$, the error vector η and slow state x are separated as

$$\dot{x}(t)=\left(A_1-A_{12}A_2^{-1}A_{21}+\varepsilon E_2\right)x(t)+A_{12}\eta(t) \tag{2.3.21}$$

$$\varepsilon\dot{\eta}(t)=\left(A_2+\varepsilon E_3\right)\eta(t) \tag{2.3.22}$$

It can be shown that there exists an ε^* such that $E_i = E_i(\varepsilon)$, $i = 1, 2, 3$, is bounded over $[0, \varepsilon^*]$. As $\varepsilon \to 0$, the eigenvalues of (2.3.22) would tend to infinity very much like $\lambda\{A_2/\varepsilon\}$ would. Now a new time variable τ, called a "stretched time-scale," is defined:

$$\tau=(t-t_o)/\varepsilon \tag{2.3.23}$$

where $\tau = 0$ at $t = t_o$ and $dt = \varepsilon d\tau$. For a change of t to τ, the system (2.3.22) will become

$$d\eta(\tau)/d\tau=\left(A_2+\varepsilon E_3\right)\eta(\tau) \tag{2.3.24}$$

which continuously depends on ε, and at $\varepsilon = 0$ it becomes with initial condition

$$d\eta(\tau)/d\tau=A_2\eta(\tau) \tag{2.3.25}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

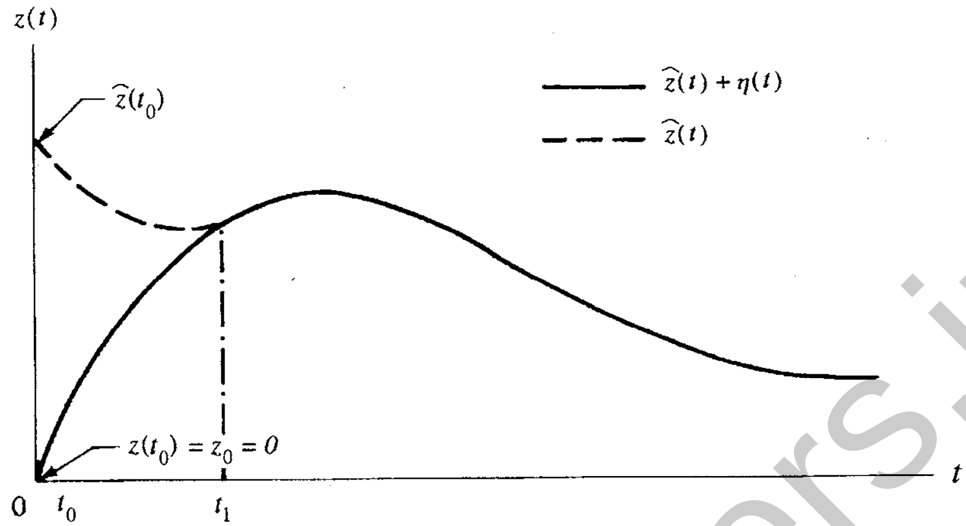


Figure 2.15 The boundary layer correction for fast state $z(t)$.

$$\eta(0) = z(t_o) - \hat{z}(t_o) \tag{2.3.26}$$

Equations (2.3.25) and (2.3.26) constitute the so-called boundary layer correction for $z(t) = \hat{z}(t) + \eta((t - t_o)/\varepsilon)$. From the above formulation, it can be shown that the *slow* and *fast* states $x(t)$ and $z(t)$ are

$$x(t) = \hat{x}(t) + 0(\varepsilon) \tag{2.3.27}$$

$$z(t) = \hat{z}(t) + \eta(\tau) + 0(\varepsilon) \tag{2.3.28}$$

where $0(\varepsilon)$ is a "large-0" order of ε and is defined as a function whose norm is less than $k\varepsilon$, with k being a constant (Kokotović *et al.*, 1976). It is noted that the boundary layer correction is only significant for the first few seconds away from t_o and reduces to zero after $t = t_1$ seconds as an exponential decay in $\tau = (t - t_o)/\varepsilon$. Figure 2.15 shows the boundary layer phenomenon for the fast state $z(t)$.

2.3.2.b Time-Scale Separation

Systems which possess multitime scales often have distinct clusters of eigenvalues (Avramovic, 1979; Kokotović, 1979; Kokotović *et al.*, 1980). It is commonly beneficial to show that linked with this system property there is a distinct possibility of decoupling the system into subsystems,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

i.e.,"slow" and "fast." One usual scheme for showing this coupling has been decomposition. In fact, the initial model reduction due to Davison (1966,1968) discussed earlier makes use of this property to separate the slowest subsystem. The main shortcoming of that method is the need to compute the entire set of eigenvalues and eigenvectors. Here an iterative scheme based on a successive weakening of the coupling between slow and fast subsystems is presented. A computational algorithm as well as a numerical example illustrates the separation procedure.

Consider a linear unforced system in singularly perturbed form:

$$\dot{x} = Ax + Ez, \quad x(t_o) = x_o \tag{2.3.29}$$

$$\varepsilon \dot{z} = Cx + Fz, \quad z(t_o) = z_o \tag{2.3.30}$$

where matrix F is assumed to be nonsingular. It was deduced already that when $\varepsilon = 0$, corresponding to a "quasi-steady-state" (qss), the actual x and z differ from qss values $\hat{x}$ and $\hat{z}$ mainly in their fast portions of response. We will now follow this notion in accordance with a report by Kokotović *et al.* (1980) to develop an iterative procedure for the separation of time scales. Let $\eta_1(t)$ be

$$\eta_1(t) = z - \hat{z} = z + F^{-1}Cx \tag{2.3.31}$$

which transforms (2.3.29)–(2.3.30) into the following set of equations. similar to (2.3.21)–(2.3.22):

$$\dot{x} = A_1 x + E\eta_1 \tag{2.3.32}$$

$$\varepsilon \dot{\eta}_1 = C_1 x + F_1 \eta_1 \tag{2.3.33}$$

where

$$A_1 \triangleq A - EF^{-1}C, \quad C_1 \triangleq \varepsilon F^{-1}CA_1 \triangleq \varepsilon B_1, \quad F_1 \triangleq F + \varepsilon F^{-1}CE \tag{2.3.34}$$

Equations (2.3.32)–(2.3.33) are in a linear time-invariant singularly perturbed form, as in the unforced case of (2.3.10)–(2.3.11). However, the important difference is that in the latter case, as is evident from (2.3.34), slow state x has a weaker presence in (2.3.33). In a similar fashion, for $\varepsilon = 0$ in (2.3.32)–(2.3.33), the qss of η_1 is obtained from

$$0 = C_1 \hat{x} + F_1 \hat{\eta}_1 \tag{2.3.35}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which is $\hat{\eta}_1 = -F_1^{-1}C_1\hat{x}$. Now introducing η_2,

$$\eta_2 = \eta_1 - \hat{\eta}_1 = \eta_1 + F_1^{-1}C_1\hat{x} \tag{2.3.36}$$

to represent the error due to letting $\varepsilon \to 0$, η_1 can be eliminated from (2.3.32)–(2.3.33) in a similar fashion using (2.3.36) to yield

$$\dot{x} = A_2x + E\eta_2 \tag{2.3.37}$$

$$\varepsilon\dot{\eta}_2 = C_2x + F_2\eta_2 \tag{2.3.38}$$

$$A_2 \triangleq A_1 - EF_1^{-1}C_1, \quad C_2 \triangleq \varepsilon F_1^{-1}C_1A_2, \quad F_2 \triangleq F_1 + \varepsilon F_1^{-1}C_1E \tag{2.3.39}$$

Note that (2.3.37)–(2.3.38) are again singularly perturbed with x now having an even weaker presence in (2.3.38) as is evident from (2.3.39) and (2.3.34), where

$$C_2 = \varepsilon^2 F_1^{-1}B_1A_2 \triangleq \varepsilon^2 B_2 \tag{2.3.40}$$

In a similar fashion, after the ith step (2.3.37)–(2.3.38) become

$$\dot{x} = A_ix + E\eta_i \tag{2.3.41}$$

$$\varepsilon\dot{\eta}_i = C_ix + F_i\eta_i \tag{2.3.42}$$

where, similar to (2.3.39),

$$A_i \triangleq A_{i-1} - EF_{i-1}^{-1}C_{i-1}, \quad A_o = A \tag{2.3.43a}$$

$$C_i \triangleq \varepsilon F_{i-1}^{-1}C_{i-1}A_i \triangleq \varepsilon^i B_i, \quad C_o = C \tag{2.3.43b}$$

$$F_i = F_{i-1} + \varepsilon F_{i-1}^{-1}C_{i-1}E, \quad F_o = F \tag{2.3.43c}$$

where C_i is reduced to an order $0(\varepsilon^i)$, as is evident from (2.3.43b). A combination of (2.3.31), (2.3.36), and

$$\eta_i = \eta_{i-1} + F_{i-1}^{-1}C_{i-1}\hat{x} \tag{2.3.44}$$

reveals that

$$\sum_{k=1}^{i}(\eta_k - \eta_{k-1}) = \eta_i - z = \sum_{k=1}^{i}\left(F_{k-1}^{-1}C_{k-1}\right)\hat{x} \tag{2.3.45}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which indicates that the slow state x remains the same, while the new fast state η_i has identical meaning with z (Kokotović *et al.*, 1980). As the iteration i approaches infinity, $A_\infty = A - EF^{-1}C + 0(\varepsilon)$ and $F_\infty = F + 0(\varepsilon)$. It is noted that even after the ith iteration, the fast state η_i still influences x, as shown by (2.3.41). Now if η_i is solved by (2.3.42) and substituted in (2.3.41),

$$\dot{x} - \varepsilon EF_i^{-1}\dot{\eta}_i = \left(A_i - EF_i^{-1}C_i\right)x \triangleq A_{i+1}x \tag{2.3.46}$$

it is seen that

$$\zeta_1 = x - \varepsilon EF_i^{-1}\eta_i \tag{2.3.47}$$

is the slow part of x. Following this observation, the slow subsystem (2.3.41) becomes

$$\dot{\zeta}_1 = A_{i1}\zeta_1 + \varepsilon A_{i1}EF_i^{-1}\eta_i \triangleq A_{i1}\zeta_1 + E_{i1}\eta_i \tag{2.3.48}$$

Note that E_{i1} is $0(\varepsilon)$, which means that the influence of the fast state η_i in the slow system has been reduced. In general,

$$\dot{\zeta}_j = A_{i1}\zeta_j + E_{ij}\eta_i, \quad \zeta_j(0) = \zeta_j^o \tag{2.3.49}$$

$$\varepsilon\dot{\eta}_i = C_i\zeta_j + F_{ij}\eta_i, \quad \eta_i(0) = \eta_i^o \tag{2.3.50}$$

where

$$A_{ij+1} = A_{ij} - E_{ij}F_{ij}^{-1}C_i, \quad A_{io} = A_i \tag{2.3.5 la}$$

$$E_{ij+1} = \varepsilon A_{ij+1}E_{ij}F_{ij}^{-1}, \quad E_{io} = E \tag{2.3.51b}$$

$$F_{ij+1} = F_{ij} + \varepsilon C_i E_{ij}F_{ij}^{-1}, \quad F_{io} = F_i \tag{2.3.51c}$$

which completes the iterative separation of slow and fast modes. Using previous discussions on weakly coupled systems conditions and above development, the following algorithm summarizes the separation of timescales:

Algorithm 2.2. Separation of Time Scales

Step 1: Set $i = j = 0$ and start with $A_i = A$, $C_i = C$, E and $F_i = F$ in (2.3.29)–(2.3.30) .

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Step 2: Evaluate A_{i+1}, C_{i+1}, and F_{i+1} from (2.3.43). Set $i = i + 1$.

Step 3: Use A_{ij}, E_{ij}, F_{ij}, and (2.3.51) to compute A_{ij+1}, E_{ij+1}, and F_{ij+1}. Set $j = j + 1$.

Step 4: Check for conditions for weakly coupled systems outlined by (2.3.7):

$$f_1 = (r/R) << 1, \quad f_2 = n_f \delta_{sf} \delta_{fs} / R^2 << 1 \tag{2.3.52a}$$

where

$$r = \max_k \left| \gamma_k \{A_{ij}\} \right|, \quad R = \min_k \left| \gamma_k \{F_{ij}\} \right| \tag{2.3.52b}$$

$$\delta_{sf} = \max_{l,k} \left| \{E_{ij}\}_{l,k} \right|, \quad \delta_{fs} = \max_{l,k} \left| \{C_i/\varepsilon\}_{l,k} \right| \tag{2.3.52c}$$

Step 5: If conditions (2.3.52) are satisfied, stop. Otherwise go to Step 2.

In the above algorithm δ_{sf} and δ_{fs} are the maximum moduli of elements in the interaction matrices between slow-fast and fast-slow subsystems, respectively, and n_f is the order of the fast subsystem. Kokotović *et al.* (1980) have reported excellent results in applying one iteration of (2.3.43) and one of (2.3.51) for a seventh-order model of a single-machine infinite-bus system, which is considered in detail here to illustrate the method.

Example 2.3.4. Consider the following unforced system, which was already considered in Example 2.2.5:

$$\dot{x} = \begin{bmatrix} -0.58 & 0 & 0 & -0.269 & 0 & 0.2 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1.0 & 0 \\ 0 & 0 & -5 & 2.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 377 & 0 & 0 \\ -0.141 & 0 & 0.141 & -0.2 & -0.28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0838 & 2 \\ -173 & 66.7 & -116 & 40.9 & 0 & -66.7 & -16.7 \end{bmatrix} x \tag{2.3.53}$$

which represents a single-machine infinite bus (Jamshidi, 1983). The above method is required to separate its time scales.

SOLUTION: The system's eigenvalues are found to be $0.362 \pm j0.556$, $-0.858 \pm j8.38$, -3.94, $-8.55 \pm j8.2$, indicating that two slow and five

$$
\tilde{A} = \left[\begin{array}{cc|ccccc}
-0.754 & 0.202 & 3.37\times 10^{-3} & -2.77\times 10^{-2} & 2.67\times 10^{-5} & 2.97\times 10^{-5} & -2.44\times 10^{-6} \\
-2.315 & 3.5\times 10^{-2} & 1.05\times 10^{-2} & -6.42\times 10^{-4} & 0.104 & 9.74\times 10^{-5} & -1.46\times 10^{-5} \\
\hline
6.84\times 10^{-3} & 3.68\times 10^{-4} & -5.01 & 2.005 & -0.611 & 0.083 & -2.7\times 10^{-4} \\
1.45\times 10^{-3} & 4.6\times 10^{-5} & -0.026 & -0.272 & 375.53 & 0.196 & -6.24\times 10^{-4} \\
-5.66\times 10^{-5} & -2.24\times 10^{-6} & 0.141 & -0.199 & -0.28 & 9.98\times 10^{-7} & 2.64\times 10^{-7} \\
-9.1\times 10^{-3} & -3.08\times 10^{-3} & 0.231 & -0.67 & 1.564 & -0.496 & 1.992 \\
-3.25\times 10^{-2} & 1.84\times 10^{-2} & -116.0 & 40.93 & -0.066 & -66.67 & -16.69
\end{array}\right]
\tag{2.3.54}
$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

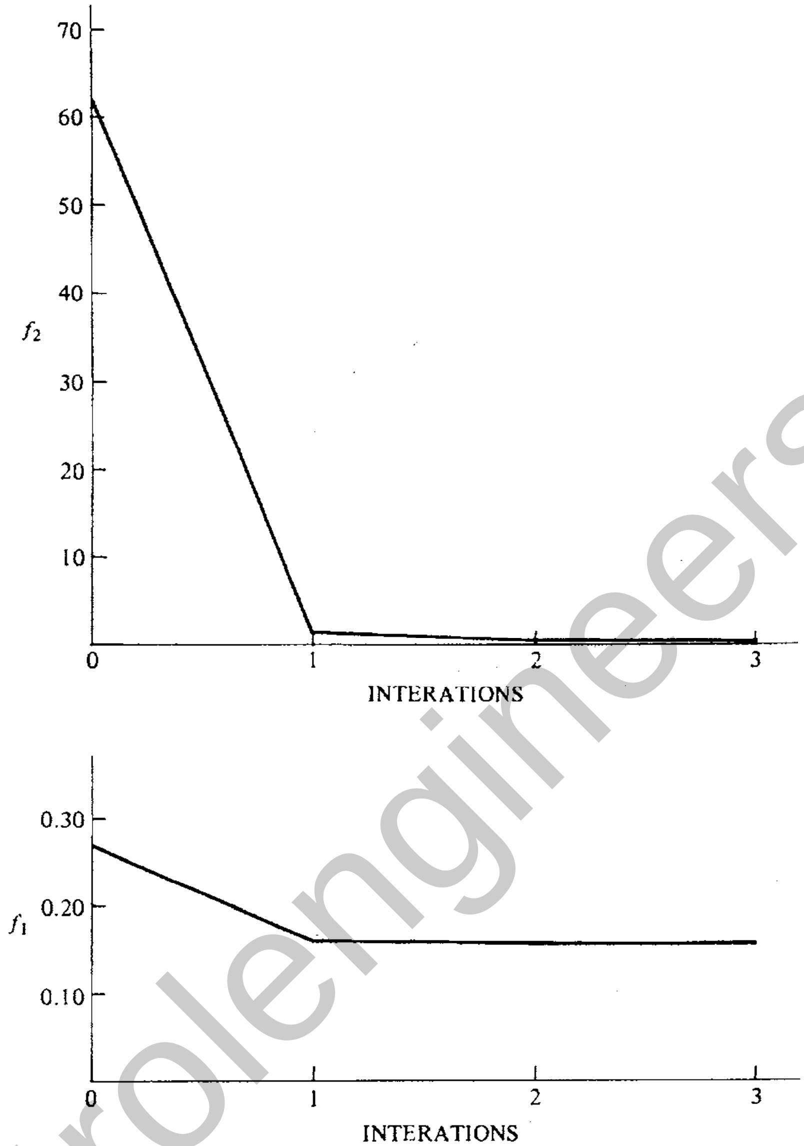


Figure 2.16 Coupling factors f_i, i = 1, 2 versus iteration number of time-scale separation of Algorithm 2.1.

$$\text{Slow: } \dot{\zeta}_3 = \begin{bmatrix} -0.754 & 0.202 \\ -2.315 & 0.035 \end{bmatrix} \zeta_3 \tag{2.3.55}$$

$$\textit{Fast:} \quad \varepsilon\dot{\eta}_3 = \begin{bmatrix} -5.01 & 2.005 & -0.611 & 0.083 & 0 \\ -0.026 & -0.272 & 375.53 & 0.196 & 0 \\ 0.141 & -0.199 & -0.28 & 0 & 0 \\ 0.231 & -0.67 & 1.564 & -0.496 & 1.992 \\ -116 & 40.93 & 0.066 & -66.67 & -16.69 \end{bmatrix} \eta_3 \tag{2.3.56}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

fast states are present. The system in its present form has the following coupling factors, defined by (2.3.52): $r = \max\left|\lambda_i\{A_1\}\right| = 1$, $R = \min\left|\lambda_j\{A_2\}\right| = 3.7225$, $\delta_{sf} = 1$, $\delta_{fs} = 173$, and $n_f = 5$; hence $f_1 = r/R = 0.2686$, $f_2 = 62.4$, which indicates that the slow and fast subsystems are highly coupled. The iterative time-scale decoupling Algorithm 2.1 was simulated on a digital computer and the factor f_1 was reduced to 0.16825, while f_2 decreased to 0.00108 in three iterations. Figure 2.16 shows the weakening between the two subsystems. The system A matrix after three iterations is shown in (2.3.54) which indicates that (2.3.53) can be reduced to (2.3.55)–(2.3.56).

Note that the iterative time-scale separation has been applied for A, E, C, and F. This implies that the actual value of parameter ε does not have to be explicitly specified in order for the method to converge (Kokotović *et al.*, 1980). The method was successfully applied to several other examples, some of which appear as problems at the end of this chapter.

Another iterative time-scale separation scheme, due to Avramović (1979) and Phillips (1979), makes use of a basis for the dominant eigenspace (Stewart, 1976). Due to space limitations, this scheme could not be considered here, but it too is very straightforward and computationally effective.

2.4 Modeling via System Identification

When no scientific data and/or equations are available about a system, often experimental input/output data may be the only knowledge on the system with an unknown model. The process of constructing models and estimating the best values of unknown parameters from experimental data is called *system identification*. In developing techniques for identification, one would be strongly influenced by the control design objectives for which its models will be used. The object of this section is to introduce the notion of system identification (ID) and system ID toolbox of MATLAB. The interested reader may contact Ljung (1987, 1988) for further details.

2.4.1 Problem Definition

Consider the dynamic model of a discrete-time linear system,

$$x(k+1) = Ax(k) + Bu(k) \tag{2.4.1a}$$

$$y(k) = Dx(k) \tag{2.4.1b}$$

whose transfer function is

$$H(z) = D(zI - A)^{-1}B \tag{2.4.2}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The system identification problem is to estimate a model of a system based on observed input-output data. To be more specific, consider a third-order system with one unit delay,

$$H(z^{-1}, p) = \frac{z^{-1}\left(b_1 + b_2 z^{-1} + b_3 z^{-2}\right)}{1 - a_1 z^{-1} - a_2 a^{-2} - a_3 z^{-3}} \tag{2.4.3}$$

where the parameter vector

$$p = \left(a_1 \; a_2 \; a_3 \; b_1 \; b_2 \; b_3\right)^T \tag{2.4.4}$$

is unknown. Let $\{U(k)\}$ represent a set of input sample values and let $\{Y(k)\}$ be the corresponding output variables of a plant whose model is given by (2.4.3) for some "true" value p^* of the parameter vector (2.4.4). Here, we are assuming that we know the order of the system. It is possible to estimate the order n of the system as well (Franklin and Powell, 1980).

One can approach the identification problem from either the frequency transfer function such as (2.4.3) or the set of state-space matrices (A,B,D) described by (2.4.1). Without any loss of generality, one can assume an observable canonical form to correspond to (2.4.3), i.e.,

$$A = \begin{bmatrix} a_1 & 1 & 0 \\ a_2 & 0 & 0 \\ a_3 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

It is noted here that the elements of the parameter vector p are now incorporated in matrices A and B. Whether the model is described in time or frequency domain, the problem statement can be summarized as follows: Given a sequence of input/output measurements for a system, find the model parameters such that some criterion for the system is satisfied. The criterian used most often are the (i) Least-Squares Estimate (LS), (ii) Best Linear Unbiased Estimate (BLUE), or (iii) Maximum Likelihood Estimate of some function of either the output error or output prediction error $e = y_p - y_m$, where y_p and y_m are plant and model outputs, respectively. Our object here is not to get too deeply into the system identification techniques but rather give sufficient background to use computer-aided identification tools such as MATLAB's System Identification Toolbox (Ljung, 1988) comfortably. Interested readers may consult the work of Ljung (1987) for a comprehensive treatment.

controlengineers.ir

controlengineers.ir
@controlengineers

2.4.2 System ID Toolbox

The System ID Toolbox is a collection of so-called ".m" files designed to give the system engineers a set of tools to identify parametric models for single-input, single-output (SISO) and multiinput, single-output (MISCO) systems.

Consider a SISO system described by Figure 2.17. The input-output variables of this system are

$$u(k);\ k = 1, 2, \ldots, n,$$
$$y(k);\ k = 1, 2, \ldots, n,$$

and noise

$$v(k);\ k = 1, 2, \ldots, n.$$

Now, let us assume that a linear relationship is governing these signals, i.e.,

$$y(k) = H(q)u(k) + w(k) \tag{2.4.5}$$

where q is the shift operator, $w(k)$ is an additional, immeasurable disturbance (noise), and $H(q)u(k)$ is represented by,

$$H(q)u(k) = \sum_{i=1}^{\infty} H(i)u(k-i) \tag{2.4.6}$$

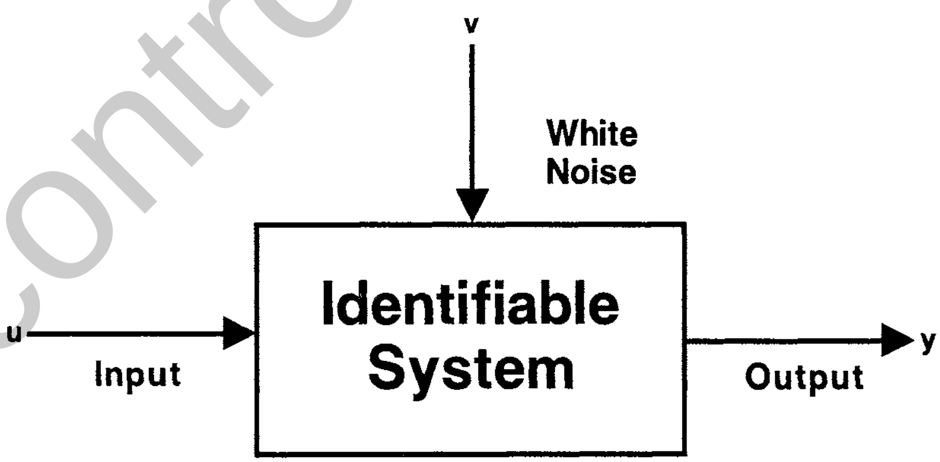


Figure 2.17 A SISO system corrupted by a white noise disturbance.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$H(q)=\sum_{i=1}^{\infty} g(i) q^{-j} \tag{2.4.7}$$

and $q^{-1}\ U(k) = U(k\text{-}1)$. The transfer function $H(q)$ evaluated on the unit circle gives the frequency function

$$H(q)\Big|_{q=e^{j\omega}} = H\left(e^{j\omega}\right) \tag{2.4.8}$$

As mentioned before, $w(k)$ in (2.4.5) is an immeasurable disturbance (noise). Its auto spectrum is defined by

$$\Phi_w(\omega)=\sum_{\tau=-\infty}^{\infty} R_w(\tau) e^{-j\omega\tau} \tag{2.4.9}$$

where $R_w(\tau)$ is the covariance function of $w(t)$:

$$R_w(t)=E\{w(k)w(k-\tau)\} \tag{2.4.10}$$

and $E\{.\}$ denotes mathematical expectation. Alternatively, the disturbance $w(k)$ can be described as filtered white noise:

$$w(k)=G(q)v(k) \tag{2.4.11}$$

where $v(k)$ is white noise with variance λ. In this case we have

$$F(w)=\lambda\left|G\left(e^{i\omega}\right)\right|^{2} \tag{2.4.12}$$

Equations (2.4.5) and (2.4.12) together give a time-domain description of the system:

$$y(k)=H(q)u(k)+G(q)v(k) \tag{2.4.13}$$

while (2.4.8) and (2.4.9) constitute a frequency-domain description.

Parametric Models To be able to estimate the functions G and H in (2.4.13), they typically have to be parameterized, most often as rational functions in the delay operator q^{-1}.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

A popular parmetric model is the ARMAX structure (prediction error)

$$A(q)y(k) = B(q)u(k-nt) + C(q)e(k) \tag{2.4.14}$$

Here $A(q)$ and $B(q)$ and $C(q)$ are

$$A(q) = 1 + a_1 q^{-1} + \ldots.. + a_{n_a} q^{-n_a} \tag{2.4.15}$$

$$B(q) = b_1 + b_2 q^{-1} + \ldots.. + b_{n_b} q^{-n_{b+1}} \tag{2.4.16}$$

$$C(q) = 1 + c_1 q^{-1} + \ldots.. + c_{n_c} q^{-n_c} \tag{2.4.17}$$

The so-called Box-Jenkin (BJ) model structure is given by

$$y(k) = \frac{B(q)}{F(q)} u(k-nt) + \frac{C(q)}{D(q)v(k)} \tag{2.4.18}$$

with

$$D(q) = 1 + d_1 q^{-1} + \ldots.. + d_{n_d} q^{-n_d} \tag{2.4.19}$$

All these models are special cases of the general parametric model structure:

$$A(q)y(k) = \frac{B(q)}{F(q)} u(k-nt) + \frac{C(q)}{D(q)} v(k) \tag{2.4.20}$$

The variance of the white noise $\{v(k)\}$ is supposed to be λ.

Within the structure (2.4.10), we recognize virtually all the usual linear black-box model structures as special cases. The ARX structure is obtained for $n_c = n_d = n_f = 0$. The ARMAX structure corresponds to $n_f = n_d = 0$. The ARARX structure (or the generalized least-squares model) is obtained for $n_c = n_f = 0$, while the ARARMAX structure (or extended matrix model) corresponds to $n_f = 0$. The output error model is obtained with $n_a = n_c = n_d = 0$, while the Box-Jenkins model corresponds to $n_a = 0$. See section 4.2 in Ljung (1987) for a detailed discussion.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Identification Given a description (2.4.13) and having observed the input-output data u, y, we can compute the (prediction) error $v(k)$:

$$v(k) = G^{-1}(q)\left[y(k) - H(v)u(k-nk)\right] \tag{2.4.21}$$

These errors will, for given data y and u, be functions of G and H. The most common parametric identification method is, naturally enough, to determine estimates of G and H by minimizing

$$V_N(G,H) = \sum_{t=1}^{N} v^2(t) \tag{2.4.22}$$

$$\left[\hat{G}_N, \hat{H}_N\right] = \arg\min \sum_{t=1}^{N} v^2(t) \tag{2.4.23}$$

This is called a prediction error method. For Gaussian disturbances it coincides with the Maximum Likelihood method (Ljung, 1987).

Finally, there are methods that estimate the frequency functions directly, without using parametric models. These are the spectral analysis methods (2.4.9). Briefly, they form estimates of the covariance functions (as defined in (3.19)) $\hat{R}_y(\tau)$, $\hat{R}_{yu}(\tau)$ and $\hat{R}_u(\tau)$, and then form estimates of the corresponding spectra:

$$\Phi_y(\omega) = \sum_{\tau=-M}^{M} \hat{R}_y(\tau)W(\tau)e^{-j\omega\tau} \tag{2.4.24}$$

and analogously for $\hat{\Phi}_u$ and $\hat{\Phi}_{yu}$. Here $W(\tau)$ is the so-called lag window and M is the width of the lag window. The estimates (Ljung, 1987) are then formed as

$$\hat{G}_N\left(e^{i\omega}\right) = \frac{\hat{\Phi}_{yu}(\omega)}{\hat{\Phi}_u(\omega)}; \qquad \hat{\Phi}_u(\omega) = \hat{\Phi}_y(\omega) - \frac{\left|\hat{\Phi}_{yu}(\omega)\right|^2}{\hat{\Phi}_u(\omega)} \tag{2.4.25}$$

Model Representation and Formats Within MATLAB's System Identification toolbox, a system model can be represented by four formats described briefly below:

The theta-format. The most common model format in the toolbox is

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

called the theta format. The format is a packaged matrix representing the parametric identification of the general multiinput, single-output linear model structure

$$A(q)y(k)=\frac{B_1(q)}{F_1(q)}u_1(k-nt_1)+\ldots+\frac{B_{nu}(q)}{F_{nu}(q)}u_{nu}(k-nt_{nu})+\frac{C_{nu}(q)}{D_{nu}(q)}v(k) \tag{2.4.26}$$

A, B, C, D, and *F* are polynomials in the delay operator of orders n_a, n_b, n_c, n_d, and n_f, respectively. If the system has *nu* inputs, *nb, nf,* and *nk* are row vectors of dimension *nu* containing information about the orders and delays associated with each of the inputs. In the case of a time series (*no u*), *B* and *F* are not defined.

Let *n* be the sum of all the orders (the number of estimated parameters). Let

$$r=\max(n,7,6+3*n_u)$$

Then theta is a $(3 + n) \times$ r-matrix organized as follows:

1. Row 1 has entries: Estimated variance of v, sampling interval T, n_u, n_a, n_b, n_c, n_d, n_f, n_k.

2. Row 2 has entries: FPE (Akaike's Final Prediction Error), year, month, date, hour, minute, and command by which the model was generated.

3. Row 3 is the vector of estimated parameters, *A, B, C, D,* and *F,* (excluding leading 1's and 0's).

4. Rows 4 to 3 + *n* contain the estimated covariance matrix.

Theta matrices are created using the function *mk* theta among other functions.

The freqfunc-format. The frequency function format stores the functions $F_w(\omega)$ and $H(e^{j\omega})$ in (2.4.12) and (2.4.13) as column vectors. It has between two and five columns. The first column contains the values of the frequencies and the second column contains the amplitude values. Depending on the context, the next columns may be, in order: standard deviation of amplitude; phase (in degrees); and, standard deviation of

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

phase. Some or all of these may be missing. The first row of a freqfunc matrix contains integers n that identify the columns. They are interpreted as follows:

$n = 0$: The column is a spectrum.
$n = 20$: The column contains standard deviations of a spectrum.
$n = 100$: The column contains frequencies for the spectrum.
$n = k$, where k is a value between 1 and 19: The column contains amplitude values for the transfer function associated with input number k.
$n = k + 20$: The column contains phase values for input number k.
$n = k + 50$: The column contains amplitude standard deviations for input number k.
$n = k + 70$: The column contains phase standard deviations for input number k.
$n = k + 100$: The column contains the frequency values for input number k.

The specified frequencies are, by default, equally spaced from 0 (excluded) to π/T over 128 values. Here T is the sampling interval (default = 1).

The polynomial format. The polynomials $A, B, C, D,$ and F in (2.4.20) are given in the standard MATLAB format for polynomials: Polynomial coefficients are stored in row vectors, ordered by descending powers. For example, $A = [1 a_1 a_2 \ldots a_{na}]$.

The zero-pole format. The zeros and poles of a model are stored in a matrix as columns. The standard deviations of the zeros and poles may also be included. The first row of the zero-pole matrix is reserved for "administrative information." By utilizing the function *ZP* in [*Zepo,k*] = *ZP*(*th*) one can compute the zeros, poles, and static gains of a theta-format model. For the general model given by Eq. (2.4.26) described in theta format by *th*, the poles and zeros and static gains are computed. The zeros are the roots of $z^{nb+nk} B(z)$ (with z replacing the forward shift operator q), and the poles are the roots of $z^{na+nf} A(z)F(z)$. The static gain is $k = B(1)/(A(1)F(1))$.

Row vector *ku* contains the number of the inputs for which the zeros and poles are to be computed. In this context, the noise $e(t)$ is counted as input number zero.

Zepo is returned as a matrix whose column numbers (2*j-1) and (2*j) are the zeros and poles, respectively, associated with input number j. K is a matrix containing the corresponding static gains. The first rows of *zepo*

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and k contain integers with information about the column in question. For the zeros and the gains, this integer is simply the input number, while for the poles it is 20 plus the input number. The default value of ku is ku = [1:nu], where nu is the number of inputs.

Notice that you can not rely upon information about zeros and poles at the origin and at infinity.

Example 2.4.1. Here, we provide a computer-aided system identification example of using the System Identification Toolbox. In the example which follows, simulated data is created for a system.

$$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{1}{A(q)}v(k)$$
$$= \frac{0.5q^{-1} + 0.25q^{-2}}{1 - q^{-1} + 0.5q^{-2}}u(k) + \frac{1}{1 - q^{-1} + 0.5q^{-2}}v(k)$$

after assuming that the input $U(k)$ and disturbance $V(k)$ are randomly generated, with a normal distribution.

SOLUTION: Below is the listing and result of this example.

```
% System ID Example
% Form simulated data
b = [0  0.5  0.25];
a = [1  -1   0.5 ];
% Use the special theta structure
th = mk theta (a, b, [1  -1  0.25])

th =
    Columns 1 through 7

    1.0000    1.0000   1.0000   2.0000   2.0000   2.0000   0
    0        19.8900   7.0000  31.0000   9.0000  26.0000   6.0000
   -1.0000    0.5000   0.5000   0.2500  -1.0000   0.2500   0
    0         0        0        0        0        0        0
    0         0        0        0        0        0        0
    0         0        0        0        0        0        0
    0         0        0        0        0        0        0
    0         0        0        0        0        0        0
    0         0        0        0        0        0        0
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

```
Columns 8 through 9
0       1.0000
0       0
0       0
0       0
0       0
0       0
0       0
0       0
0       0
```

```
% Present system information from
% simulated model
>> present (th)
```

This matrix was created by the command MKTHET on 7/31 1989 at 9:26 Loss fcn: 1 Akaike's FPD: 0 Sampling interval 1 The polynomial coefficients and their standard deviations are

```
B =
    0          0.5000          0 . 2500
    0          0               0

A =
    1.0000     -1.00001        0.5000
    0           0              0

C=
    1.0000     -1.0000         0.2500
    0           0              0
```

```
% Generate an input signal u, a disturbance
% signal v, and simulate the response
% of the model to these inputs
rand ('normal')
u = sign (rand (200. 1));
v = rand (200. 1);
y = idsim ([uvl,th):
% Plot input u and output y

subplot (211)
plot (u(1:50)), title('Input Signal U: First 50 Values'), . . .

subplot (212)
plot(y), title('Output Signal y'), pause
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers


```
% Now that the simulated data has been corrupted,
% we can estimate a model using the
% least squares (ARX) method. First form a
% matrix z whose columns are the
% output y and the input u.
z = [y u];
% ARX model identification with two poles,
% two zeros, and a single delay on the input
a_arx = arx (z, [2 2 1])

a_arx =

  Columns 1 through 7
   1.2996    1.0000   1.0000    2.0000   2.0000   0         0
   1.3526   19.8900   7.0000   31.0000   9.0000  33.0000    1.0000
  -0.2425    0.0184   0.3842    0.5246   0        0         0
   0.0042   -0.0014  -0.0001    0.0014   0        0         0
  -0.0014    0.0041   0.0008   -0.0003   0        0         0
  -0.0001    0.0008   0.0068    0.0005   0        0         0
   0.0014   -0.0003   0.0005    0.0071   0        0         0

  Columns 8 through 9
  0       1.0000
  0       0
  0       0
  0       0
  0       0
  0       0
  0       0

% present system information from
% ARX model
present (a_arx)

This matrix was created by the command ARX   on 7/31 1989 at 9:33 Loss fcn: 1.3
Akaike's FPE: 1.353 Sampling interval 1
The polynomial coefficients and their standard deviations are

B =
  0  0.3842    0.5246
  0  0.0824    0.0841

A =
  1.0000      -0.2425        0.0184
  0            0.0652        0.0639
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

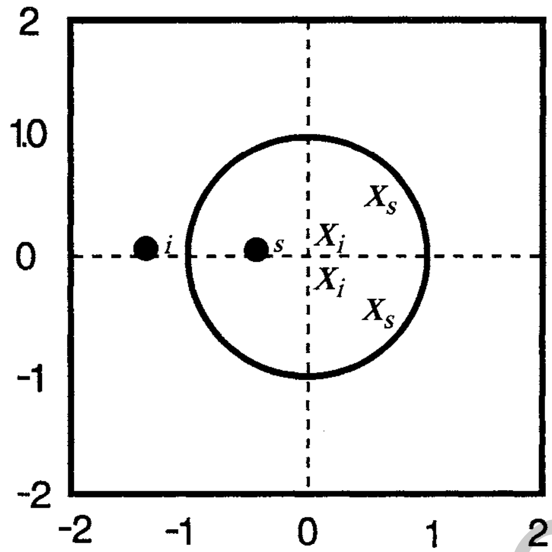


Figure 2.18 Pole-zero locations for simulated and identified models (Note: *s* = simulated and *i* = identified).

```
% zeros and poles of simulated model
zpo=zp(th,1)
zpo =
      1 . 0000     21.000
     -0.5000       0.5000 + 0.500i
      0            0.5000 - 0.5000i
% zeros and poles of ARX model

xp_arx=zp(a_arx,1)

zp_arx =

      1 . 0000     21.000
     -1.3654       0.1212 + 0.0605i
      0            0.1212 - 0.0605i
      % Frequency response of simulated
      % and ARX models

[go,no]=trf(th);
[grx,nrx]=trf(a_arx);

% Bode plots of simulated and ARX (identified) models
bodeplot ([go grx])
% zero-poles plots of simulat and ARX (identified) models
zpplot ([zpo zp_arx])
```

controlengineers.ir

controlengineers.ir
@controlengineers

For the previous example, the least-square (ARX) model is given by

$$y(k)=\frac{B(q)}{A(q)}u(k)=q^{-1}\frac{\left(0.3842+0.5246q^{-1}\right)}{1-0.2425q^{-1}+0.018q^{-2}}u(k)$$

whose poles and zero were already obtained from the command ZP. A plot of the pole-zeros of the simulated and ARX models is shown in Figure 2.18.

2.5 Modeling via Fuzzy Logic

Similar to neural networks, fuzzy logic can also be used to identify a mapping function or an implicit transfer function from the input to output of the system. The approach here is to take numerical data from the input and output of the system and design a fuzzy associate memory (FAM) or simply a model for the system. This section is based on the work of Wang and Mendel (1990).

Consider a two-input, one-output system S whose model is unknown. Assume that the following input/output set of N data points are known:

$$\begin{array}{llll}
1 & \left(u_1^{(1)},u_2^{(1)}\right) & \rightarrow & y^{(1)} \\
2 & \left(u_1^{(2)},u_2^{(2)}\right) & \rightarrow & y^{(2)} \\
\vdots & & & \\
i & \left(u_1^{(i)},u_2^{(i)}\right) & \rightarrow & y^{(i)} \\
\vdots & & & \\
N & \left(u_1^{(N)},u_2^{(N)}\right) & \rightarrow & y^{(N)}
\end{array} \tag{2.5.1}$$

where u_1 and u_2 are inputs and y is the output of the system. A FAM (a set of fuzzy if-then rules) is sought to determine a mapping f: $(u_1,u_2) \rightarrow y$. The following algorithm would create a model using a fuzzy rule-based system.

Algorithm 2.3. Model Identification via Fuzzy Rules

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Step 1: Fuzzy Regions Formulation. Let the domain of intervals (universes of discourse) be $[\underline{u}_1, \bar{u}_1],[\underline{u}_2, \bar{u}_2]$, and $[\underline{y}, \bar{y}]$, respectively. Divide each domain of the interval into $2N + 1$ regions, where N is an arbitrary integer whose value can be different for different variables and the resulting regions can have unequal lengths. Let the $1N + 1$ regions be denoted by SL (small N), . . . $S1$ (small 1), CE (center), $B1$ (big 1), . . . , BN (big N), and assign each region with a fuzzy membership function. Figure 2.19 shows a case where variables u_1 and u_2 are divided into five ($N = 2$) and three ($N = 1$), respectively and the variable y into five regions ($N = 2$). The shape of the membership functions, as shown here, is chosen to be triangular without any loss of generality.

Step 2. Fuzzy Rules Generation. For every pair of data $\left(u_1^{(i)}, u_2^{(i)}, y^{(i)}\right)$ determine its degrees within various regions (linguistic labels). For example $u_1^{(i)}$, in Figure 2.19 has degree 0.8 in $B1$, degree 0.2 in $B2$, and zero degrees in all other regions. Next, assign variables in each set to regions in which they have maximum degree. For example, u_2 has degree 1 in CE and it will be assigned CE as its linguistic label. At this point, obtain one rule from each pair of the (u, y) data set. As an example (see Figure 2.19).

$$\left(u_1^{(1)}, u_2^{(1)}, y^{(1)}\right) \Rightarrow \left[u_1^{(1)}(0.8 \textit{ in } B1, \max), u_1^{(1)}(1 \textit{ in } CE, \max);\right.$$
$$\left. y^{(1)}(0.9 \textit{ in } CE, \max)\right] \Rightarrow \tag{2.5.2}$$

Rule 1: If u_1 is $B1$ and u_2 is CE, THEN y is CE.

$$\left(u_1^{(2)}, u_2^{(2)}, y^{(2)}\right) \Rightarrow \left[u_1^{(2)}(0.6 \textit{ in } B1, \max), u_2^{(2)}(0.7 \textit{ in } B1, \max);\right.$$
$$\left. y^{(2)}(0.7 \textit{ in } B1, \max)\right] \Rightarrow \tag{2.5.3}$$

Rule 2: If u_1 is $B1$ and u_2 is $B1$, THEN y is $B1$.

These rules use "and" for antecedents, which is by no means a shortcoming.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

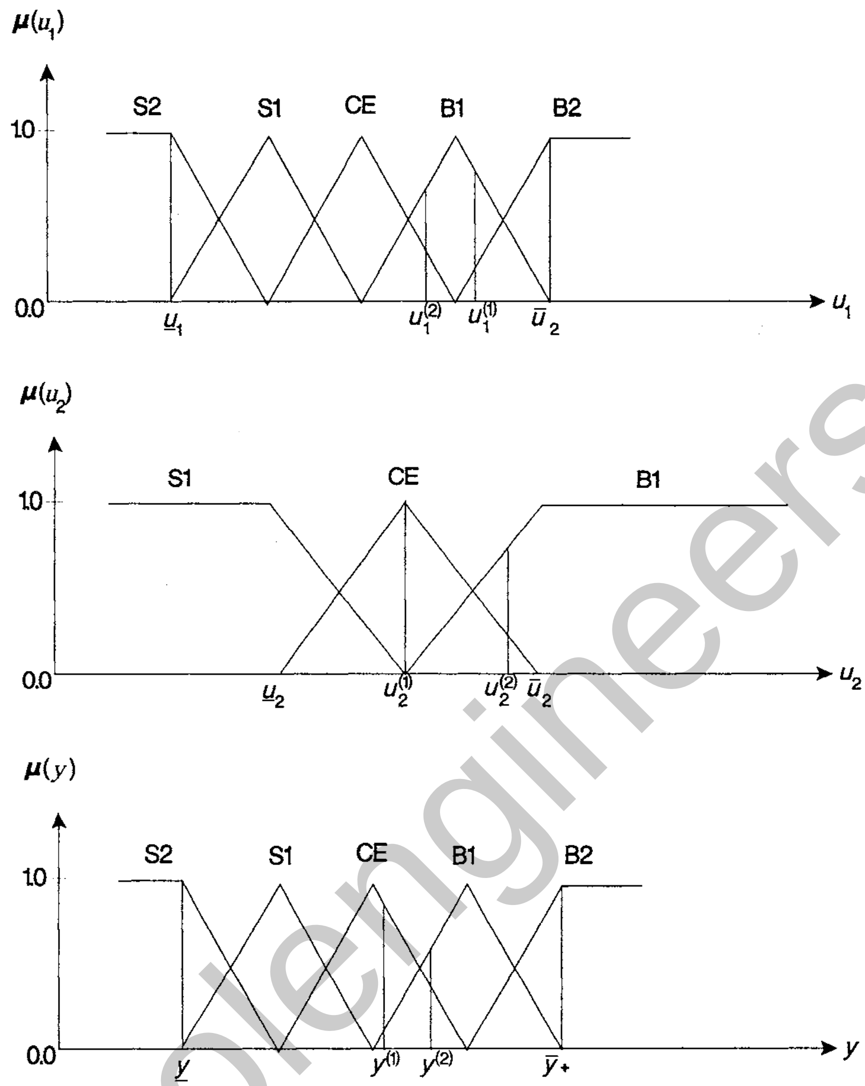


Figure 2.19 An example division of system's input/output variables into linguistic labels.

Step 3: Rules Degree Assignment. Thus far, a rule has been generated for each pair of input/output data. However, since there are, in general, numerous data points, there are many rules which would be in conflict with each other, i.e., rules with the same antecedents may have different consequents. Wang and Mendle (1990) have proposed a degree assignment to each rule and retain the conflicting rule with maximum degree. Through this

controlengineers.ir

controlengineers.ir @controlengineers

process, one can both eliminate conflicting rules and reduce the total number of rules. The degree of a rule is defined as follows:

For the rule "IF u_1 is A and u_2 is B, THEN y is C," define its degree as

$$D(\text{Rule}) = \underset{A}{\mu}(u_1)\ \underset{B}{\mu}(u_2)\ \underset{C}{\mu}(y) \tag{2.5.4}$$

As examples, considering the two rules associated with Figure 2.20, we have

$$D(\text{Rule 1}) = \underset{B1}{\mu}(u_1)\ \underset{CE}{\mu}(u_2)\ \underset{CE}{\mu}(y) = 0.8\times 1\times 0.9 = 0.72 \tag{2.5.5}$$

and for Rule 2, the degree is

$$D(\text{Rule 2}) = \underset{B1}{\mu}(u_1)\ \underset{B1}{\mu}(u_2)\ \underset{B1}{\mu}(y) = 0.6\times 0.7\times 0.7 = 0.294 \tag{2.5.6}$$

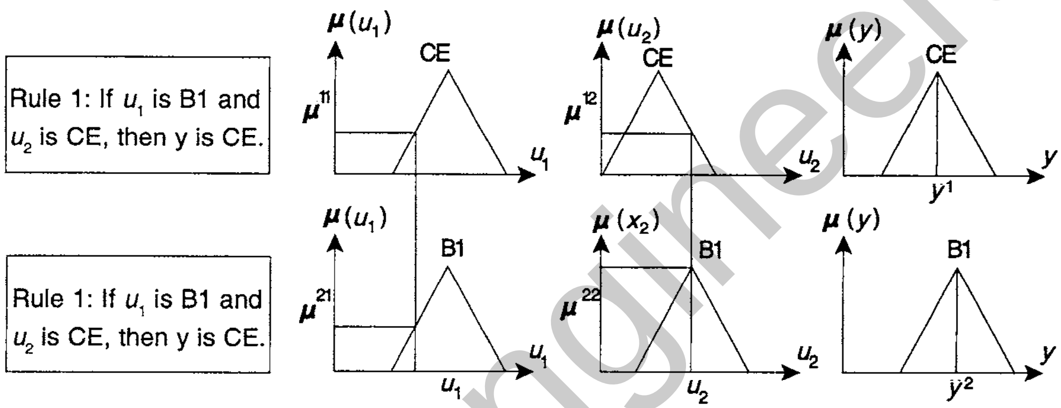
Step 4: FAM Bank Creation. A FAM Bank is shown in Figure 2.20. The boxes in the FAM will be filled by fuzzy rules according to the following strategy: A combined FAM Bank is assigned rules from those generated from numerical data. If there is more than one rule to be assigned to a box in the FAM Bank, use the rule with maximum degree. If a rule is an "and" rule, then only one box in FAM will be filled (see Figure 2.20), and when a rule is an "or" rule (i.e., when an OR separates segments of the antecedent), it will fill all the boxes in the rows or columns corresponding to the labels of the IF part. For example, suppose we have a rule: IF u_1 is CE or u_2 is $S1$ THEN y is $S2$. Under this rule, the three boxes of column CE and the five boxes of row $S1$ will be filled up with $S2$, as shown in Figure 2.20. The degrees of all the $S2$'s in these boxes equal the degree of this "or" rule.

Step 5: Mapping Based on the FAM Bank. Given an input pair (u_1, u_2), one needs to determine the output y based on the FAM Bank. This is done by the process of defuzzification as briefly introduced in Chapter 1 and Appendix B. Figure 2.21 shows the centroid defuzzification process to determine output y based on the two rules discussed by Eqs. (2.5.2) and (2.5.3) and depicted in the FAM of Figure 2.20. As seen, the antecedents of

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

u_2 \ u_1	S2	S1	CE	B1	B2
B1			S2	**B1**	
CE			S2	**CE**	
S1	S2	S2	S2	S2	S2

Figure 2.20 The form of a FAM Bank.



$$y = \frac{\mu^1_{CE}\bar{y}^1 + \mu^2_{B1}\bar{y}^2}{\mu^1_{CE} + \mu^2_{B1}} \qquad \mu^1_{CE} = \mu^{11}\mu^{12}, \quad \mu^2_{B1} = \mu^{21}\mu^{22}$$

Figure 2.21 Determining the output strategy from FAM Bank using product fuzzy inference with centroid defuzzification.

the ith fuzzy rule using *product* operations to determine the degree, $\mu^i_{0^i}$, of the output corresponding to inputs (u_1,u_2), i.e.,

$$\mu^i_{0^i} = \mu_{I_1^i}(u_1)\mu_{I_2^i}(u_2) \tag{2.5.7}$$

where 0^i denotes the output region of Rule i, and denotes the input region of Rule i for the jth component, e.g., Rule 1 gives

$$\mu^1_{CE} = \mu_{B1}(u_1)\ \mu_{CE}(u_2) \tag{2.5.8}$$

controlengineers.ir

controlengineers.ir @controlengineers

Finally, a centroid defuzzification formula can be used to determine the output of the system,

$$y = \frac{\sum_{i=1}^{N} \mu_{0^i}^{i} \bar{y}^i}{\sum_{i=1}^{N} \mu_{0^i}^{i}} \tag{2.5.9}$$

where $\bar{y}^i$ is the center value of region 0^i, i.e., the point which has the smallest absolute value among all the points at which the membership function is equal to one. N in (2.5.9) is the number of fuzzy rules in the FAM Bank.

The above five-step algorithm can be used as an alternative scheme for system identification and modeling very much like neural networks. The only difference is that through fuzzy rule-based systems a mathematical model is not explicitly obtained. Below, two examples are presented to describe fuzzy rule-based approximation and the use of the above algorithm.

Example 2.5.1. Consider a function $y = \sin(x) + \cos(x)$ where $-\pi \le x \le \pi$. It is desired to find a fuzzy rule-based system to approximate y versus x.

SOLUTION: We begin by plotting the function y versus 0 as shown in Figure 2.22. The rules will be obtained by first dividing the universe of discourse of y and x, i.e., $-1.5 \le y \le$ and $-\pi \le x \le \pi$ into an appropriate number of linguistic regions (labels or fuzzy sets). For this example, fuzzy logic software environment FULDEK (Dreier, 1991) was used (Appendix B provides a brief overview of FULDEK). The variable x was divided into 16 labels with equal distant triangular forms from NEG7, NEG6, … , NEG1, ZERO, POS1, … ,POS8. For y, the interval was divided into nine labels from NEG4 to ZERO to POS4. In order to realize the mapping between x and y, nine fuzzy IF-THEN rules were composed within FULDEK:

Rule 1: IF x is x_POS2 THEN y is y_POS4.

Rule 2: IF x is x_POS1 or x is x_POS3 THEN y is y_POS3.

Rule 3: IF x is x_ZERO or x is x_POS4 THEN y is y_POS2.

Rule 4: IF x is x_NEG1 or x is x_POS5 THEN y is y_POS1.

Rule 5: IF x is x_NEG1 or x is x_POS6 THEN y is y_ZERO.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

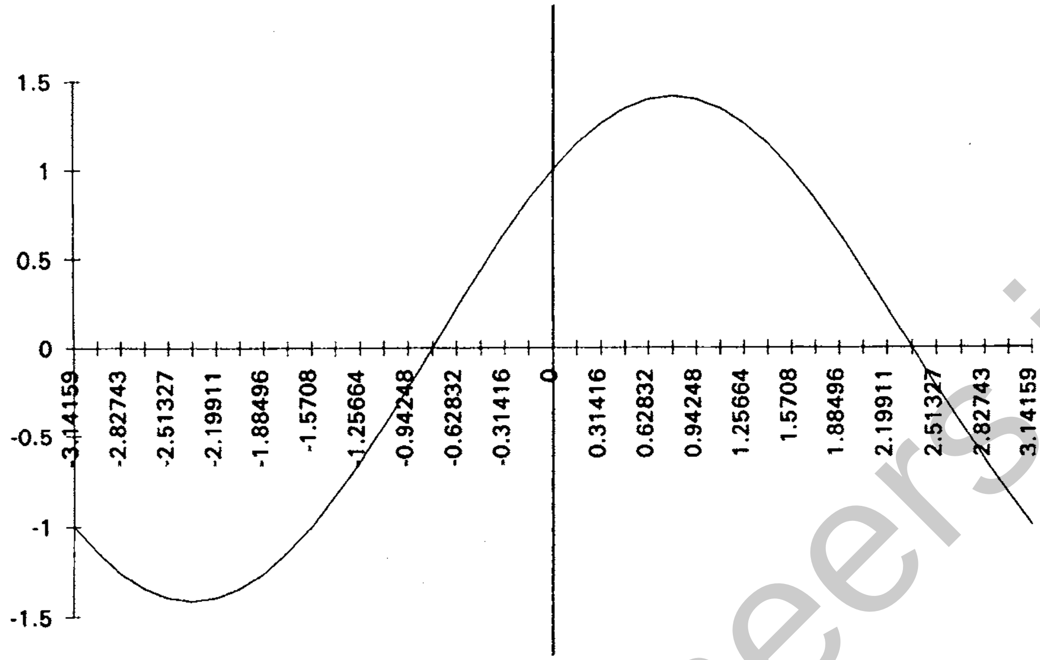


Figure 2.22 A fuzzy rule-based approximation function example.

Rule 6: IF x is x_NEG3 or x is x_POS7 THEN y is y_NEG1.

Rule 7: IF x is x_NEG4 or x is x_POS8 THEN y is y_NEG2.

Rule 8: If x is x_NEG7 or x is x_NEG5 THEN y is y_NEG3.

Rule 9: IF x is x_NEG6 THEN y is y_NEG4.

Executing all these rules, a y versus x plot is obtained as shown in Figure 2.23. Next, rules 2 and 8 were eliminated and the execution was repeated. The new approximation is shown in Figure 2.24. It is seen that the loss of two rules has not degraded the approximation of y as a function of x too much.

Example 2.5.2. Consider the five-step fuzzy rule generation Algorithm 2.4. A program was written in C Language by Xue and Chong (1993) to implement this algorithm (a list of this code can be obtained from the author). It is desired to use this algorithm to generate a number of fuzzy rules to identify a relationship $y = u_1 + u_2$ between two inputs (u_1, u_2) and the output y.

controlengineers.ir

controlengineers.ir
@controlengineers

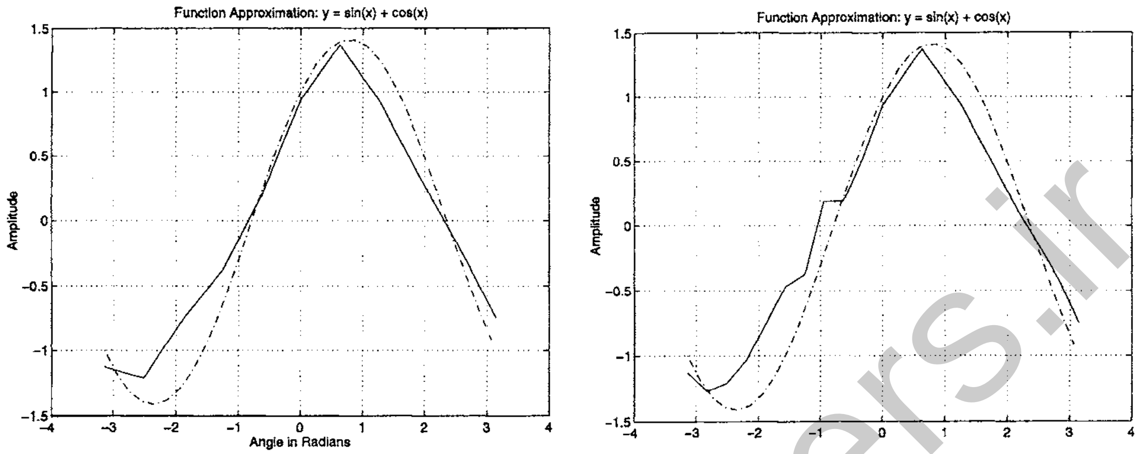


Figure 2.23 Fuzzy logic approximation of sinusoidal function using all nine rules and two sets of membership functions.

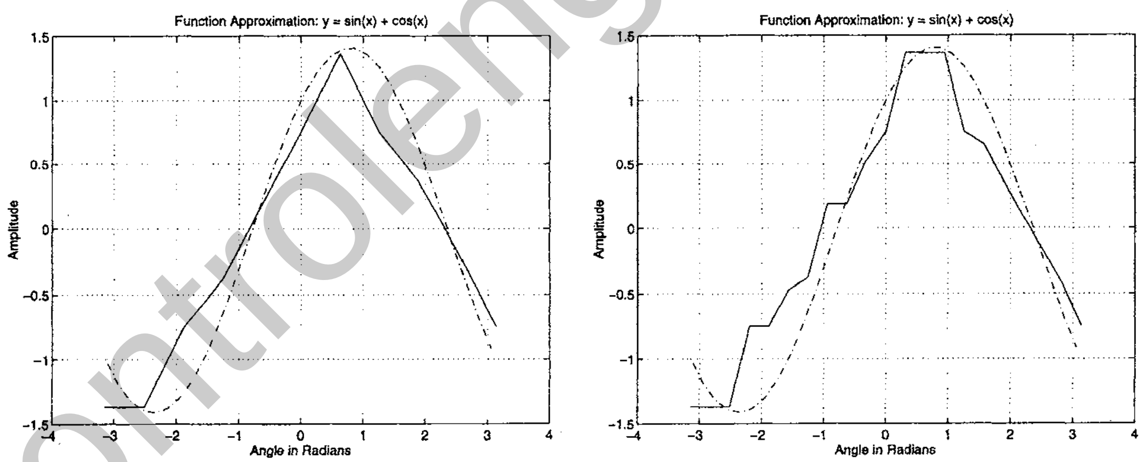


Figure 2.24 Fuzzy logic approximation of sinusoidal function using seven rules and two sets of membership functions.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

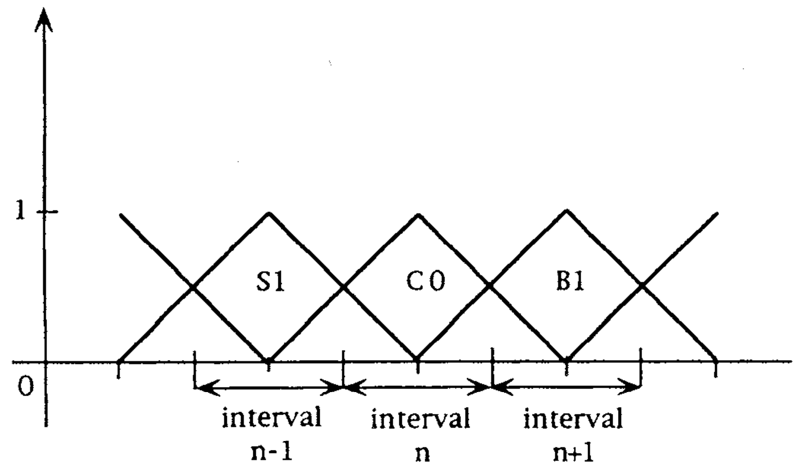


Figure 2.25 Triangular membership functions for Algorithm 2.3.

SOLUTION: The program implementing Algorithm 2.4 was used in two different manners. One was to use an equal number of labels $Nu_1 = Nu_2 = Ny = 7$ without any loss of generality and use various number of data points in the initial data, i.e., 100, 225, or 400. Another case was also tried by keeping the number of data points at 225 and changing the number of regions (labels) from 5 to 7 to 9. Figure 2.25 shows the general shape of membership functions used in this algorithm. The output data from the data sets are compared with those output data after defuzzification. In the crisp set case, the relationship between the two output data is a straight line ($y1 = y2$). However, in fuzzy set case, the relationship will be somewhat "fuzzy" as we can see in Figures 2.26 – 2.27. The diversity of the data point from the *Crisp Line* can be used as a measurement to find out the fitness of the combination FAM Bank and defuzzification method.

Table 2.1 shows the generated rules and the resulting FAM Banks of five cases, i.e., Nu1 = Nu2 = Ny = 5, 7, and 9 as well as two cases in which the number of regions are fixed at 7 and use 100 and 400 data points. As we can conclude by intuition, these figures show that the more intervals we divide the universe discourses (either inputs or outputs) into and the more training data points we use, the better fitness we get from the rule-generating program. However, more intervals will cause a larger FAM Bank,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

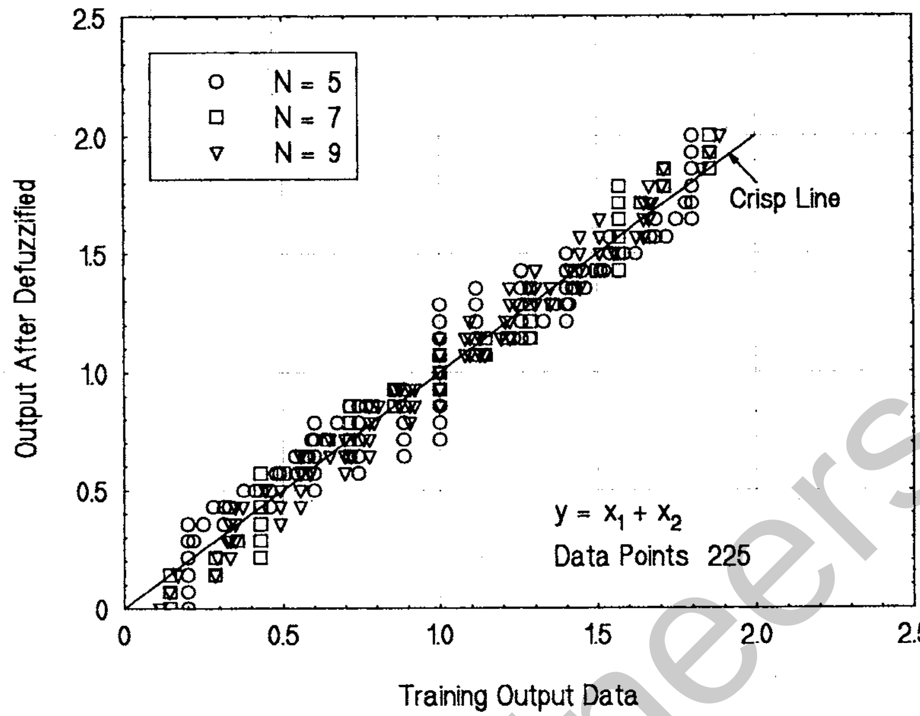


Figure 2.26 Rule-generating program results for Example 2.5.2 with fixed number of data points.

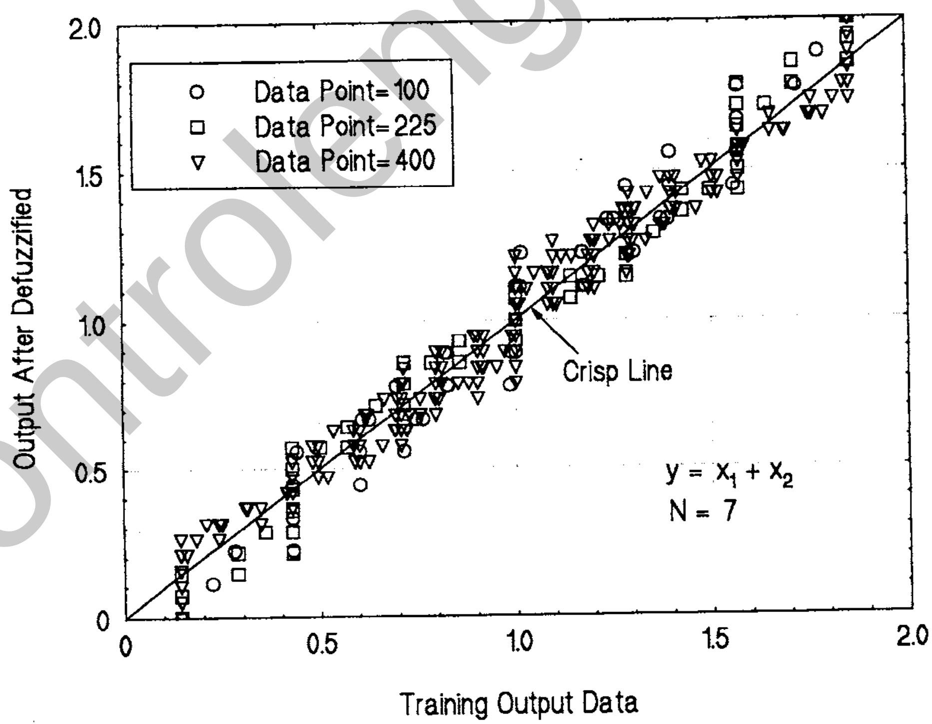


Figure 2.27 Rule generating program results for Example 2.5.2 with fixed number of regions.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

TABLE 2.1
A Summary of FAM Banks for Example 2.5.2

Fuzzy Associate Memory Bank for function of y = x1 + x2

1) Data points = 225

1-1) N = 5

B2	C0	C0	B1	B2	B2
B1	C0	C0	B1	B1	B2
C0	S1	S1	C0	B1	B1
S1	S2	S1	S1	C0	C0
S2	S2	S2	S1	C0	C0
	S2	S1	C0	B1	B2

1-2) N = 7

B3	C0	B1	B1	B2	B2	B2	B3
B2	S1	C0	B1	B1	B2	B2	B2
B1	S1	S1	C0	B1	B1	B2	B2
C0	S2	S1	S1	C0	B1	B1	B2
S1	S2	S2	S1	S1	C0	B1	B1
S2	S2	S2	S2	S1	S1	C0	B1
S3	S3	S2	S2	S2	S1	S1	C0
	S3	S2	S1	C0	B1	B2	B3

1-3) N = 9

B4	C0	C0	B1	B1	B2	B2	B3	B3	B4
B3	C0	C0	B1	B1	B2	B2	B3	B3	B3
B2	S1	S1	C0	C0	B1	B1	B2	B3	B3
B1	S1	S1	C0	C0	B1	B1	B1	B2	B2
C0	S2	S2	S1	S1	C0	B1	B1	B2	B2
S1	S2	S2	S1	S1	S1	C0	C0	B1	B1
S2	S3	S3	S2	S1	S1	C0	C0	B1	B1
S3	S3	S3	S3	S2	S2	S1	S1	C0	C0
S4	S4	S3	S3	S2	S2	S1	S1	C0	C0
	S4	S3	S2	S1	C0	B1	B2	B3	B4

2) Interval Number, N = 7

2-1) Data Points = 100

B3	C0	C0	B1	B1	B2	B2	B3
B2	C0	C0	C0	B1	B2	B2	B2
B1	S1	C0	C0	B1	B1	B2	B2
C0	S1	S1	S1	C0	B1	B1	B1
S1	S2	S2	S1	S1	C0	C0	B1
S2	S2	S2	S2	S1	C0	C0	C0
S3	S3	S2	S2	S1	S1	C0	C0
	S3	S2	S1	C0	B1	B2	B3

2-2) Data Points = 400

B3	C0	C0	B1	B2	B2	B3	B3
B2	C0	C0	B1	B1	B1	B2	B3
B1	S1	S1	C0	C0	B1	B1	B2
C0	S2	S1	C0	C0	C0	B1	B2
S1	S2	S1	S1	C0	C0	B1	B1
S2	S3	S2	S1	S1	S1	C0	C0
S3	S3	S3	S2	S2	S1	C0	C0
	S3	S2	S1	C0	B1	B2	B3

controlengineers.ir
@controlengineers

which will, in turn, dramatically need more CPU time to defuzzify the output and make the fuzzy controller slow. In contrast, more training data points do not require too much CPU time to establish the FAM Bank for better fitness. Therefore, it is a better way to enhance the fitness of the rule-generating program by using as many training data as available.

Problems

2.1. For a system

$$(A,B)=\left(\begin{bmatrix}-1 & 0.5 & 1\\ 2 & -10 & 0\\ 1 & 0 & -12\end{bmatrix},\ \begin{bmatrix}1\\1\\1\end{bmatrix}\right)$$

find a second-order aggregated model using the two approaches discussed in Example 2.2.1. Are these dynamically exact?

2.2. A third-order system is described by

$$\dot{x}=\begin{bmatrix}-1.25 & 0.02 & 0\\ -0.5 & -0.1 & -1\\ -1 & -0.5 & -0.08\end{bmatrix}x+\begin{bmatrix}1\\0\\1\end{bmatrix}u$$

Find an aggregated model using the model scheme of Davison (1966) described in Section 2.2.1.

2.3. Repeat Problem 2.2 using the model scheme of Chidambara (1969).

2.4. A second-order system is given by

$$\dot{x}=\begin{bmatrix}-1 & 0\\ 1 & -10\end{bmatrix}x+\begin{bmatrix}1\\1\end{bmatrix}u.$$

Find an aggregation matrix which would provide a dynamically exact aggregation. Repeat for both the dominant and non-dominant eigenvalues of the system.

2.5. Show that aggregation matrix C described by Equation (2.2.58) provides perfect aggregation.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.6. Suppose that a 4×4 A matrix has $\lambda\{A\}=\{a, a, b \pm jc\}$; then it is known that its modal matrix is

$$M = \left[\xi_1 \;\vdots\; v_1 \;\vdots\; \upsilon_1 \;\vdots\; w_1\right],$$

where ξ_1 is a right eigenvector, v_1 is a generalized eigenvector, and $u_{1,3} = \upsilon_1 \pm jw_1$ is a pair of complex conjugate eigenvectors. Using this, extend the modal aggregation of Davison (1966) for repeated and complex eigenvalues-eigenvectors.

2.7. Repeat Problem 2.6 for the method of Chidambara (1969).

2.8. Use modal aggregation methods to find a second-order reduced model for

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -400 & -460 & -262 & -32 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} u$$

2.9. Repeat Problem 2.8 for

$$(A, B) = \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -20 & -22 & -12 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)$$

2.10. A system is described by

$$(A, B) = \left(\begin{bmatrix} -2 & -1 & -0.5 & -0.1 & -0.05 \\ 0.5 & -1 & 0.2 & -4 & 0.8 \\ 1 & -0.1 & -0.25 & 0 & -1 \\ 0 & 1 & -0.1 & -0.1 & 0 \\ 1 & -1 & -1 & 0.5 & -5 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right)$$

Using LSSTB for eigenvalue-eigenvector evaluations, find an aggregated model using the modal method of Davison (1966).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.11. CADLAB Repeat Problem 2.10 using program "MODAL" of package LSSPAK.

2.12. CADLAB An unstable system is described by

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.5 & -1 & -10 & 3 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \underline{u}$$

$$\underline{y} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \underline{x}$$

Find a reduced-order model using the balanced approach using MATLAB. Hint: Transform the A matrix by an appropriate amount (A-2I) before model reduction.

2.13. Repeat Problem 2.12 for the following stable system:

$$\dot{\underline{x}} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -1 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \underline{u}$$

$$\underline{y} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 2 & 1 \end{bmatrix} \underline{x}$$

2.14. CADLAB Is the following system weakly coupled?

$$\dot{x} = \begin{bmatrix} -1 & -1 & 0 & 0 & 1 \\ 0.2 & -1 & 0.2 & -1 & 0 \\ 0 & -0.5 & -0.2 & -5 & -10 \\ 1 & 1 & -0.1 & -10 & -20 \\ 0 & -2 & 1 & -20 & -25 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 2 \\ 0 & 2 \end{bmatrix} u$$

2.15. Is the system

$$(A, B, C) = \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.5 & -5.6 & -6.1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \right)$$

weakly coupled?

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.16. For a system

$$(A,B,C)=\left(\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -1 & -a & -b\end{bmatrix},\ \begin{bmatrix}1\\ 0\\ 1\end{bmatrix},\ \begin{bmatrix}1 & 0 & 1\end{bmatrix}\right)$$

find the region(s) in the $(b-a)$ plane such that the system is weakly coupled.

2.17. Let the following system be dependent on two parameters:

$$\dot{\underline{x}}=\begin{bmatrix}-1 & 0 & a\\ 0 & 0 & -1\\ b & 0 & -1\end{bmatrix}\underline{x}$$

For what range on a and b is this system weakly coupled?

2.18. A singly perturbed system is described by

$$\begin{aligned}\dot{x}&=-\tfrac{1}{2}x+z \qquad & x(0)&=1\\ \varepsilon\dot{z}&=-x-2z \qquad & z(0)&=1\end{aligned}$$

Find a boundary layer for it.

2.19. CADLAB Consider an eighth-order unforced system

$$\dot{x}=\begin{bmatrix}-0.5 & 0 & 0 & -0.2 & 0.1 & 0.2 & 0 & 0.1\\ 0 & -1 & 0 & 0.1 & 0 & 1 & 0 & 0\\ 0.1 & 0 & -2 & 0 & 0 & 0 & 1 & 0\\ 0.05 & 0 & 0 & -20 & -1 & 0 & 0 & 0\\ -0.15 & 1 & 0 & 1 & 30 & 1 & 0 & 0\\ 0 & 0 & 0.15 & -0.2 & -0.3 & -1 & 0 & 0\\ -15 & 0 & 0 & 1 & 0 & 0 & 0.08 & 0\\ -80 & -10 & -10 & 2 & -50 & 0 & 6 & -15\end{bmatrix}x$$

Use Algorithm 2.1 and your favorite computer program or LSSTB or LSSPAK to separate the time scales.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.20. Use an appropriate computer program to check whether the system

$$\dot{x} = \begin{bmatrix} -0.21054 & -0.10526 & -0.007378 & 0 & 0.0706 & 0 \\ 1 & -0.03537 & -0.000118 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -605.16 & -4.92 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -3906.25 & -12.5 \end{bmatrix} x$$

is weakly coupled.

2.21. Use MATLAB or other packages. Repeat Problem 2.18 for the system

$$\dot{x} = -x + z, \quad x(0) = -1$$
$$\varepsilon\dot{z} = x - 5x, \quad z(0) = 1$$

2.22. Use FULDEK to approximate the function $y = |x|,\ -1 \le x \le 1$.

Repeat for $y = \cos(2x)$ for $-\frac{\pi}{4} \le x \le \frac{\pi}{4}$.

2.23. A function $y = u_1^2 + u_2^2$ represents the mapping between two inputs and one output. Use the rule generation steps of Section 2.6 (Algorithm 2.4) to generate at least three FAM Banks and compare the defuzzified (approximate) y.

2.24. Show that for a stable matrix A, Grammian matrices G_c and G_0 satisfy Eqs. (2.2.61) and (2.2.62)

2.25. A system is described by

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.5 & -1 & -10 & -3 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} x$$

Find a reduced-order model using the balanced approach.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

2.26. Use definition of $\hat{G}_o$, $\hat{G}_c$ and transformation matrix S in Equation (2.2.73) to show that $\hat{G}_c = \hat{G}_o = \Sigma$.

2.27. The balanced method of Sec. 2.2.3 can be extended to unstable linear time-invariant systems if one notes that their balancing Lyapunov equations (Santiago and Jamshidi, 1986) would be

$$G_o(A-kI)+(A-kI)^T G_o + C^T C = 0$$
$$G_c(A-kI)^T + (A-kI)G_c + BB^T = 0$$

where k is greater than the real part of the positive-most eigenvalue of the A matrix. Use this concept to devise a balancing algorithm for unstable systems.

2.28. Use the approach of Problem 2.27 to reduce the following unstable system using the balancing method:

$$(A,B,C) = \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -60 & -8 & 9 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 50 & 15 & 3 \end{bmatrix} \right)$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 3

Structural Properties of Large-Scale Systems

controlengineers.ir

3.1 Introduction

The high dimensionalities, nonlinearities, and complexities of interconnection in large-scale systems provide computational and analytical difficulties not only in modeling, control, or optimization but also in the fundamental issues of stability, controllability, and observability. As the dimension of the system increases, the problem of assessing these structural properties becomes much more difficult. In this chapter these three important properties of system theory from the large-scale systems viewpoint are briefly considered.

When the stability of large-scale system is of concern, one basic approach, consisting of three steps, has prevailed: decompose a given large-scale system into a number of small-scale subsystems, analyze each subsystem using the classical stability theories and methods, and combine the results leading to certain restrictive conditions with the interconnections and reduce them to the stability of the whole. This approach has been termed the "composite system method" by Araki (1978a) and many others.

One of the earliest efforts regarding the stability of composite systems is due to Bailey (1966), which assumed a Lyapunov function for each subsystem; then using the theory of the vector Lyapunov function (Šiljak, 1972a; Matrosov, 1972), the stability of the composite system was checked (Moylan and Hill, 1978). Others (Araki and Kondo, 1972; Michel and

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Porter, 1972) have constructed a scalar Lyapunov function as a weighted sum of the Lyapunov functions of the individual subsystems. This line of work has given rise to the so-called "Lyapunov methods." These methods have been extended by many authors, including Michel (1975a,b), Rasmussen and Michel (1976a,b), and Šiljak (1976).

An alternative approach, known as the "input-output method," describes each subsystem by a mathematical relation or an operator on functional space, and then functional analysis methods are employed. Porter and Michel (1974) and Cook (1974) considered the situation where subsystems have finite gain and are conic, which is a generalization of Zames's (1966) single-loop results (Moylan and Hill, 1978). The input-output stability has been extended by several authors, including Sundareshan and Vidyasagar (1975), Araki (1978a), and Lasley and Michel (1976a,b).

A basic issue involved in interconnected systems stability is the question of how large the interactions magnitudes and strength can be before the stability of the composite system is affected? Furthermore, as mentioned by Sandell *et al.* (1978), in some systems a strong coupling exists between various subsystems which makes a major contribution to stability. Such issues lead us to connective stability, which is essentially the extension of stability, in the sense of Lyapunov, to take into account the structural perturbations (Šiljak, 1972a,b, 1978). We will see in Chapters 4 and 6 that when a system is expected to undergo structural perturbations, the interactions have major effects on control and estimation (observability) and, just as importantly, stability.

A system is said to be "completely state controllable" if it is possible to find an unconstrained control vector $u(t)$ that would transfer any initial state $x(t_o)$, for any t_o, to any final state $x(t)$, say origin, in a finite time interval $t_o \le t \le t_f$. Observability, on the other hand, is a concept related to the determination of the state from the measurement of output. A system is said to be "completely observable" at any time t_o if it is possible to determine $x(t_o)$ by measuring $y(t)$ over the interval $t_o \le t \le t_1$. Consider a linear TIV system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.1.1}$$

$$y(t) = Cx(t) + Du(t) \tag{3.1.2}$$

where x, u, and y are n-, m-, and r-dimensional state, control, and output vectors, respectively, and A, B, C, and D are constant matrices of appropriate dimensions. The standard criteria for checking controllability and

controlengineers.ir

controlengineers.ir
@controlengineers

observability of this system are the following two rank conditions (Kalman, 1960):

$$\text{rank } P = \text{rank}\{B \quad AB \quad \cdots \quad A^{n-1}B\} = n \tag{3.1.3}$$

$$\text{rank } Q = \text{rank}\{C^T \quad A^TC^T \quad \cdots \quad (A^T)^{n-1}C^T\} = n \tag{3.1.4}$$

where P and Q are the $n \times nm$ controllability and $n \times nr$ observability matrices, respectively. These conditions are useful and computationally simple only if n is small, i.e., less than 10. However, if the system is large in scale or has particular inherent properties which make these two conditions difficult to check, alternative criteria are required.

The bulk of research in the controllability and observability of large-scale systems falls into four main problems: controllability and observability of composite systems, controllability (and observability) of decentralized systems, structural controllability, and controllability of singularly perturbed systems.

The controllability and observability of composite (series, parallel, and feedback) systems was first considered by Gilbert (1963), where the controllability and observability of the system were studied in terms of those of the subsystems. Since that initial work, many researchers have considered this problem; among them are Chen and Desoer (1967), Rosenbrock (1970), Brasch *et al.* (1971), Klamka (1972, 1974), Yonemura and Ito (1972), Bhandarkar and Fahmy (1972), Grasselli (1972), Wolovich and Huang (1974), Hautus (1975), Davison (1977), Porter (1976), Sezer and Hüseyin (1977, 1979).

The "structural controllability" has been introduced by Lin (1974) which determines the controllability of the pair (A, b) through the properties of system structure through the graph of (A, b). Structural controllability has been further considered by many other authors, including Shields and Pearson (1976), Corfmat and Morse (1975), Glover and Silverman (1975), and Davison (1977). The graph theoretic concepts have also been used by Davison (1976) for composite systems.

It therefore bears repeating that the stability, controllability, and observability of large-scale systems have been of concern to many researchers in the field, with stability being perhaps the most dominant one. The literature on these topics is indeed so vast that a dedicated text can be written on any of these subjects. Consequently, it is well beyond the scope of this book to go into great detail. However, we attempt to present here the most relevant methods of testing large-scale systems stability, controllability, and observability. In specific, two fundamental techniques, one of Lyapunov and the other the input-output method, are introduced in Sections 3.2 and 3.3, respectively. An important extension of Lyapunov stabil-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ity, i.e., connective stability, is also considered in Section 3.2. The topics considered in Section 3.4 are controllability and observability of composite systems. Structural controllability and observability of large-scale systems are discussed in Section 3.5. Section 3.6 deals with computer-aided structural analysis of large-scale systems. This and other pertinent materials, such as dissipative systems stability along with a comparative discussion on the topics, are presented in Section 3.7.

3.2 Lyapunov Stability Methods

One of the most celebrated methods of investigating system stability is the direct method of Lyapunov. In simple terms, the method is described as follows: For a given system, a scalar function $\upsilon(x, t)$, known as the Lyapunov function (sometimes representing a system's total energy), is found in terms of the state variables; then based on the properties of $\upsilon(x, t)$ and its time derivative $\upsilon(x, t)$, various conclusions may be made regarding the system stability. It is assumed here that the reader is already familiar with the basic notions of Lyapunov stability, and our main purpose here is to point out how the Lyapunov method may be applied to large-scale interconnected systems. The next section provides the necessary definitions and a statement of the problem, while Section 3.2.2 gives a few criteria for checking the stability of large-scale systems.

3.2.1 Definitions and Problem Statement

The following definitions are necessary elements in the understanding of stability criteria set forth in Section 3.2.2.

Definition 3.1. A scalar real-valued function $\upsilon(x, t)$ of state x and time t is said to be "positive-definite" if there is a nondecreasing real-valued function $\hat{\upsilon}(x,t)$ such that $\hat{\upsilon}(0,0)=0$ and $0, \hat{\upsilon}(x,t) \le \hat{\upsilon}(x,t)$ for all $x \neq 0$. The function $\upsilon(x, t)$ is further called radially unbounded if $\hat{\upsilon}(\infty,t)=\infty$. The single bars $|\ |$ denote the Euclidean norm defined by $|x| = \left(x_1^2 + \ldots + x_n^2\right)^{1/2}$.

Definition 3.2. A positive-definite function $\upsilon(x, t)$ is called "decrescent" if a nondecreasing function $\tilde{\upsilon}(|x|)$ exists such that $\tilde{\upsilon}(0)=0$ and $\upsilon(x,t) \le \tilde{\upsilon}(|x|)$. A function which is positive-definite, decrescent, and radially unbounded is denoted by pdu.

Definition 3.3. An $n \times n$ matrix A is said to be an "M-" or "Metzler-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

matrix" if the following equivalent conditions hold:

1. All principal minors of A are positive.
2. All leading principal minors of A are positive.
3. For a vector x (or y) whose elements are all positive, the elements of Ax (or A^Ty) are all positive.
4. A^{-1} exists and all its elements are nonnegative.
5. Re $\{\lambda_i(A)\} > 0$ for $i = 1,2, \ldots, n$.
6. A diagonal matrix $B = \text{diag}(b_1, \ldots, b_n)$, $b_i > 0$, exists such that $BA + A^TB$ is a positive-definite matrix. This condition is sometimes termed Lyapunov type.

Definition 3.4. A real square matrix Ω is called an "*M*-matrix" if there is a diagonal $\tilde{D} = \text{diag}\left(\tilde{d}_i, \ldots, \tilde{d}_N\right)$ matrix with $\tilde{d}_i > 0$ such that $\tilde{D} - \Omega^T \tilde{D} \Omega$ is a positive-definite matrix (Araki, 1978a).

With the above preliminary definitions presented, the system whose stability is of concern is now defined. Consider a large-scale unforced system described by

$$\text{Lss: } \dot{x} = f(x,t) \tag{3.2.1}$$

which is assumed to consist of N subsystems,

$$\text{Css: } \dot{x}_i = f_i(x_i,t) + g_i(x,t), \quad i = 1,\ldots,N \tag{3.2.2}$$

where $x_i, f_i(\cdot)$, and $g_i(\cdot)$ satisfy $x = \left(x_1^T \ldots x_N^T\right)$, $f = \left\{f_1^T(\cdot) + g_1^T(\cdot) \ldots f_N^T(\cdot) + g_N^T(\cdot)\right\}$. The notations Lss and Css refer, respectively, to the original large-scale system (3.2.1) and composite subsystem representation (3.2.2). A further assumption is that the origin $x_e = 0$ is an equilibrium state, i.e.,

$$f(0,t) = 0, \;\; f_i(0,t) = 0, \quad g_i(0,t) = 0 \tag{3.2.3}$$

for $i = 1, 2, \ldots, N$. The set of N composite subsystems (3.2.2) can be rewritten as

$$\dot{x}_i = f_i(x_i,t) + u_i \tag{3.2.4}$$

by the substitution of $u_i = g_i(x,t)$, which in effect represents an interaction or interconnection input into the ith subsystem. When the ith subsystem is

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

completely decoupled,

$$\text{Iss:}\ \ \dot{x}_i = f_i(x_i, t) \tag{3.2.5}$$

then it is referred to as an isolated subsystem for the sake of our discussions. Furthermore, the present discussion is restricted to stability with respect to the equilibrium point $x_e = 0$ or $x_{ie} = 0$, $i = 1, \ldots, N$. The following definitions define uniform stability, uniform asymptotic stability, and uniform asymptotic stability in the sense of Lyapunov.

Definition 3.5. For a positive-definite, decrescent function $\upsilon(x, t)$ of system state x and time t, the equilibrium point $x_e = 0$ of (3.2.1) is said to be "uniformly stable" if $-\dot{\upsilon}(x,t)\big|_{(\text{Lss})} \geq 0$ for all x and t. Note that the notation $k(\cdot)\big|_{(\text{Lss})}$ means that the arguments of $k(\cdot)$ are evaluated along the trajectories of the large-scale system in (3.2.1).

Definition 3.6. In view of Definition 3.5, the equilibrium point $x_e = 0$ is "uniformly asymptotically stable" if $-\dot{\upsilon}(x,t)\big|_{(\text{Lss})} \geq 0$ and uniformly a.s.i.L. (asymptotically connectively stable in the Large) if, in addition, $u(x, t)$ is radially unbounded (Definition 3.1). The function $\upsilon(x, t)$ of Definitions 3.5 and 3.6 is called a Lyapunov function.

After the above developments and definitions, the problem can be stated as follows: For a large-scale system Lss is defined by (3.2.1), which can be decomposed into N subsystems Css (3.2.2) with isolated subsystems Iss (3.2.5) and interconnections u; defined in (3.2.4), under what conditions is the original composite system (Lss) a.s.i.L.? This problem is usually approached in two different ways. One is to assume a Lyapunov function $\upsilon_i(x_i, t)$ for the isolated subsystems Iss (3.2.5), which are presumably obtained more easily than the original system and use a weighted sum

$$\upsilon(x,t) = \sum_{i=1}^{N} c_i \upsilon_i(x_i, t) \tag{3.2.6}$$

as a potential Lyapunov function for the Lss (3.2.1). In (3.2.6) the coefficients c_i, $i = 1, \ldots, N$ are positive constants. This approach has been considered by many authors in the field, Araki and Kondo (1972), Grujić and Šiljak (1973a,b), and Thompson (1970) to name a few. The other approach is to define a vector Lyapunov function $\upsilon = (\upsilon_1(x_1, t), \ldots, \upsilon_N(x_N, t))$ and by

controlengineers.ir
@controlengineers

virtue of the Metzler matrice properties and positive-definite decrescent and radially unbounded functions (Definitions 3.1–3.3) appropriate conditions are attached to $\dot{\upsilon}$ for stability of the Lss (3.2.1). The latter approach, not considered here, had been considered originally by Bailey (1966) and later by Cuk and Šiljak (1973), Matrosov (1972), Piontkovskii and Rutkovskaya (1967), and Šiljak (1972a,b, 1974a,b, 1975).

Referring back to the weighted sum of (3.2.6), as long as each $\upsilon_i(x_i, t)$ is positive-definite, decrescent, and radially unbounded (pdu), the Lss composite system Lyapunov function $u(x, t)$ is also pdu. Therefore, in view of Definitions 3.5 and 3.6, the only condition to be checked is the negative definiteness of $\dot{\upsilon}\big|_{(\mathrm{Lss})}$. Differentiating (3.2.6) yields

$$\dot{\upsilon}(x,t)\big|_{(\mathrm{Lss})} = \sum_{i=1}^{N} c_i \dot{\upsilon}_i(x_i,t)\big|_{(\mathrm{Lss})} \tag{3.2.7}$$

where

$$\begin{aligned}\dot{\upsilon}(\cdot)\big|_{(\mathrm{Lss})} &= \partial\upsilon_i/\partial t + (\partial\upsilon_i/\partial x_i)^T \left(f_i(x_i,t) + g_i(x,t)\right) \\ &= \dot{\upsilon}(\cdot)\big|_{(\mathrm{Lss})} + (\partial\upsilon_i/\partial x_i)^T g_i(x,t)\end{aligned} \tag{3.2.8}$$

and the gradient

$$\partial\upsilon_i/\partial x_i = \left(\partial\upsilon_i/\partial x_{i1}, \ldots, \partial\upsilon_i/\partial x_{in_i}\right) \tag{3.2.9}$$

with n_i being the order of the ith subsystem and x_{ij}, $j = 1,2, \ldots, n_i$, representing the jth element of the ith subsystem state vector. The above development suggests the following algorithm for analyzing a composite large-scale system which has been suggested by several authors (Sandell *et al.*, 1978; Araki, 1978b).

Algorithm 3.1. Lyapunov Stability of Large-Scale Composite Systems

Step 1: Decompose the Lss (3.2.1) into N subsystems Css (3.2.2), assume a Lyapunov function $\upsilon_i(x_i, t)$ for each isolated subsystem Iss (3.2.5), and obtain a bound for $\dot{\upsilon}_i(\cdot)\big|_{(\mathrm{Iss})}$.

Step 2: Find a bound for each interconnection or interaction term $g_i(x, t)$, $i = 1, \ldots, N$.

Step 3: Obtain a condition for the existence of positive constants c_i, $i = 1, \ldots, N$, such that the negative-definiteness of (3.2.7)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is guaranteed from the above bounds.

The next section provides two theorems which provide stability criteria for Css (3.2.2).

3.2.2 Stability Criteria

The uniform stability in the sense of Lyapunov for Css (3.2.2) has been extensively treated by many authors, including Araki (1978b), who has presented one of the more conveniently applicable approaches. Here two of Araki's results will be given.

Theorem 3.1. *The large-scale system Lss (3.2.1) represented by composite subsystems Css (3.2.2) is uniformly a.s.i.L. if the following conditions hold:*

1. *For each Iss there is a pdu function $\upsilon_i(x_i, t)$ such that*

$$\dot{\upsilon}_i(x_i,t)\big|_{(\mathrm{Iss})} \le -a_i\{w_i(x_i)\}^2 \tag{3.2.10}$$

$$|\partial \upsilon_i(\cdot)/\partial x_i| \le w_i(x_i) \tag{3.2.11}$$

where a_i, $i = 1, \ldots, N$ are positive constants and $w_i(x_i)$ is a positive definite function.

2. *The interconnection terms are bounded by*

$$|g_i(x,t)| \le \sum_{\substack{j=1 \\ j\ne i}}^{N} b_{ij} w_j(x_j) \tag{3.2.12}$$

where b_{ij} are nonnegative constants.

3. *The $N \times N$ matrix $E = (e_{ij})$ given by*

$$e_{ii} = a_i - b_{ii}, \quad e_{ij} = -b_{ij}, \quad i \ne j \tag{3.2.13}$$

is an M-matrix; i.e., the leading principle minors of E are all positive:

$$D_i \triangleq \det\begin{bmatrix} e_{11} & \cdots & e_{1i} \\ \vdots & \ddots & \vdots \\ e_{i1} & \cdots & e_{ii} \end{bmatrix} > 0, \quad i = 1, \ldots, N \tag{3.2.14}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

PROOF: Through condition 3 and recalling Definition 3.3, one can choose $d_i > 0$ such that $DE + E^TD$ is a positive-definite matrix where $D = \text{diag}(d_1, \dots, d_N)$. Define $\upsilon(x, t)$ by (3.2.6). Then $u(x, t)$ is a pdu function.

Using (3.2.7)–(3.2.13), one can obtain

$$
\begin{aligned}
\dot{\upsilon}(x,t)\Big|_{(Lss)} &\le \sum_{i=1}^{N} d_i \left[-a_i w_i(x_i)^2 + w_i(x_i) \left(\sum_{\substack{j=1 \\ j\neq i}}^{N} b_{ij} w_j(x_j) \right) \right] \\
&= -\tfrac{1}{2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \left(d_i e_{ij} + d_j e_{ji} \right) w_i(x_i) w_j(x_j)
\end{aligned}
\tag{3.2.15}
$$

Now since $DE + E^TD$ is positive-definite, the right-hand side of (3.2.15) is a negative-definite function of x. Thus, Lss is uniformly a.s.i.L by Definition 3.6. Q.E.D. ■

Before the use of the above theorem is illustrated by examples, it is worthwhile to interpret the three conditions of Theorem 3.1. Condition 1 implies that each isolated subsystem is a.s.i.L. with $\upsilon_i(x_i,t)$ as its Lyapunov function. A common candidate for $\upsilon_i(x_i,t)$ is a quadratic function. The constant a_i is considered as the degree of stability in view of the fact that it gives a lower bound for the decrease in υ with respect to $w_i^2(x_i)$. Moreover, the constants b_{ij} in (3.2.12) indicate the interconnections' strength in the sense that they provide an upper bound with respect to $w_i^2(x_i)$. Condition 3 indicates that the diagonal elements of E are, in general, larger than the off-diagonal ones. As a whole, Theorem 3.1 indicates that if the subsystems are stable to a degree larger than the strength of the interconnections, then the composite system is stable.

Example 3.2.1. Consider a fifth-order system decomposed into third- and second-order subsystems:

$$
\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{ccc:cc} -1 & 0.1 & 0.2 & 0.1 & 0.2 \\ 0.2 & -2 & 0.5 & 0.1 & 0.1 \\ 0.1 & -1 & -3 & 0.5 & 0.4 \\ \hdashline 1 & 0 & 1 & -4 & 0.2 \\ 0.2 & 0.5 & 0 & 1 & -5 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix}
\tag{3.2.16}
$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It is desired to check its stability by means of Theorem 3.1.

SOLUTION: Since the two subsystems are linear and the A_i, $i = 1, 2$ are stable matrices, the following Lyapunov functions $\upsilon_i(x_i, t)$ and positive-definite functions $w_i(x_i)$ are chosen:

$$\upsilon_i(x_i,t) = x_i^T P_i x_i, \quad w_i(x_i) = \alpha_i\left(x_i^T Q_i x_i\right)^{1/2} \tag{3.2.17}$$

for $i = 1, 2$ and P_i, Q_i satisfy

$$P_i A_i + A_i^T P_i + Q_i = 0 \tag{3.2.18}$$

Choosing $Q_1 = I_3$ and $Q_2 = 2I_2$ in an arbitrary fashion, two solutions of (3.2.18) provide

$$P_1 = \begin{bmatrix} 0.508 & 0.023 & 0.032 \\ 0.023 & 0.254 & -0.006 \\ 0.032 & -0.006 & 0.168 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0.26 & 0.03 \\ 0.03 & 0.20 \end{bmatrix} \tag{3.2.19}$$

A value of $\alpha_i = 1/a_i^2$ where $a_i = \max_{x_i \neq 0}\left(|2P_i x_i| / \left(x_i^T Q_i x_i\right)^{1/2}\right)$ would guarantee condition (3.2.10) of Theorem 3.1. Furthermore, if we choose $b_{11} = b_{22} = 0$,

$$b_{12} = \max_{x_2 \neq 0}\left\{|G_{12}x_2| / \alpha_2\left(x_2^T Q_2 x_2\right)^{1/2}\right\}$$

and

$$b_{21} = \max_{x_1 \neq 0}\left\{|G_{21}x_1| / \alpha_1\left(x_1^T Q_1 x_1\right)^{1/2}\right\}$$

the condition (3.2.12) would be satisfied. In above formulation, G_{12} and G_{21}, are the off-diagonal interaction submatrices defined in (3.2.16). These four maximation procedures were performed through a coordinate rotation method due to Rosenbrock (1960), and the results after a few iterations were, $\alpha_1 = -1.0255$, $a_1 = 0.9509$, $a_2 = 0.38$, $a_2 = 6.93$, b_{21}, $= 1.387$, and $b_{12} = 1.28$. Using these parameters, the matrix E in (3.2.13) becomes

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$E = \begin{bmatrix} 0.9509 & -1.280 \\ -1.3870 & 6.930 \end{bmatrix} \tag{3.2.20}$$

which is clearly a Metzler matrix. Thus we conclude that the composite systems (3.2.16) are uniformly a.s.i.L. according to Theorem 3.1. This checks with the application of Lyapunov stability to the overall system whose eigenvalues are -0.9, $-2.32 \pm j0.54$, -4.52, -4.94.

Example 3.2.2. Consider a second-order nonlinear system

$$\begin{aligned} \dot{x}_1 &= -2/3x_1^3 + g_1(x,t) \\ \dot{x}_2 &= -1/3x_2^3 + g_2(x,t) \end{aligned} \tag{3.2.21}$$

where the interaction terms $g_i(x)$, $i = 1, 2$, will be defined shortly. Check for a.s.i.L.

SOLUTION: This example has been inspired by a similar example by Araki (1978a). It turns out that this system's isolated subsystems fall within the class of $\dot{x}_i = \psi_i(x_i)$ such that $0 < \psi_i(x_i)x_i$, $x_i \neq 0$, and $\overline{\psi}_i(x_i) = \int_0^{x_i} \psi_i(\tau)d\tau \to \infty$ as $|x_i| \to \infty$. A set of possible choices for $\upsilon_i(x_i, t)$ and $\omega_i(x_i)$ are $\upsilon_i(x_i, t) = \psi_i(x_i)$ and $w_i(x_i) = |\psi_i(x_i)|$. Using these values, the condition (3.2.10) would be satisfied for all $a_i \leq 12$, $i = 1, 2$. If the interaction values are assumed to be $g_i(x,t) = C_i\psi_i(x_i) + \hat{g}_i(x,t)$, where $\hat{g}_i(\cdot)$ is not dependent on x_i, then by appropriate choices of positive constants c_i and parameter b_{ij}, condition (3.2.12) can also be satisfied.

Theorem 3.1 assumes that all isolated subsystems Iss (3.2.5) are a.s.i.L., which is a rather strong assumption. The following theorem provides the stability criteria using a weaker set of assumptions.

Theorem 3.2. *The large-scale system Lss (3.2.1) represented by composite subsystem Css (3.2.2) is uniformly a.s.i.L. if the following conditions hold:*

1. *For each Iss there is a pdu function $\upsilon_i(x_i, t)$ such that*

$$\dot{\upsilon}_i(x_i,t)\big|_{(\mathrm{Iss})} \leq -a_i\{w_i(x_i)\}^2 - z_i(x_i) \tag{3.2.22}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where a_i, $i = 1, \ldots, N$, are positive constants and $w_i(x_i)$ and $z_i(x_i)$ are positive-semidefinite and positive-definite functions, respectively.

2. *The interconnection terms satisfy*

$$\left(\partial v_i(x_i,t)/\partial x_i\right)^T g_i(x,t) \le w_i(x_i)\sum_{\substack{j=1\\ j\neq i}}^{N} b_{ij} w_j(x_j) \tag{3.2.23}$$

3. *The $N \times N$ matrix E defined by (3.2.13) is an M-matrix.*

The proof of this theorem is similar to Theorem 3.1 and is left to the reader as an exercise (see Problem 3.8). It is noted that $w_i(x_i)$ in (3.2.22) can be positive-semidefinite due to $z_i(x_i)$ and b_{ij}, and, hence, a_i can be negative, since $\left(\partial v_i/\partial x_i\right)^T g_i$ is estimated directly. This development would allow the isolated subsystems Iss (3.2.5) to be unstable. The following example illustrates the application of Theorem 3.2.

Example 3.2.3. Consider a second-order nonlinear interconnected system

$$\begin{aligned}\dot{x}_1 &= 0.25x_1 - x_1 \cos x_2 + 0.2x_2 \sin x_1x_2\\ \dot{x}_2 &= x_2 - x_2 \sin x_1 + 0.125x_1 \cos x_1x_2\end{aligned} \tag{3.2.24}$$

Check if it is a.s.i.L.

SOLUTION: In this example $n_1 = n_2 = 1$ and $N = 2$. The two isolated subsystems $\dot{x}_1 = 0.25x_1$ and $\dot{x}_2 = x_2$ are unstable, which implies that Theorem 3.1 is not applicable. Let $v_i(x_i, t) = x_i^2/2$ and $w_i(x_i) = |x_i|$, $i = 1, 2$ and follow the conditions (3.2.22) and (3.2.23) for $z_i(x_i)$, any arbitrary positive-definite function. The resulting parameters are $a_1 = 0.25$, $a_2 = 1$, $b_{11} = b_{22} = -1$, $b_{12} = 0.2$, and $b_{21} = 0.125$. With these values, the E matrix in (3.2.13) is

$$E = \begin{bmatrix} 1.25 & -0.2 \\ -0.125 & 2 \end{bmatrix} \tag{3.2.25}$$

which is clearly an M-matrix.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

There are several possibilities of extending the above results by changing the assumptions on $\dot{v}_i$, g_i and redefining elements e_{ij} of E. For further details, the reader can consult the work of Araki (1978a).

3.2.3 Connective Stability

An important issue in large-scale systems is the perturbation in its structure, either intentionally (by design) or unintentionally (by fault). The basic issue is how such system perturbations affect the overall system stability. This notion, termed connective stability by Šiljak (1972a), has been developed extensively in a book subsequently by Šiljak (1978). The object of this section is to introduce this notion and present a criterion for connective stability.

3.2.3.a System Structure and Perturbation

Consider a linear large-scale unforced system

$$\dot{x} = Ax \tag{3.2.26}$$

where A is an $n \times n$ matrix whose elements a_{ij} represent the extent of influence that state or "agent" x_i puts on state x_j for $i, j = 1, \ldots, n$. In order to examine the structure of such a system, let $n = 3$ and rewrite (3.2.26) as

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + a_{22}x_3 \\ \dot{x}_3 &= a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{aligned} \tag{3.2.27}$$

whose relations can be shown by a "directed graph" or simply a "diagraph" as in Figure 3.1. The points or nodes on the graph represent subsystem (or system) states, while the directed lines correspond to the elements of matrix A, the extent of the influence of one state on itself or another state. Now suppose that $a_{31} = 0$ in (3.2.27), which implies that matrix A is

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{bmatrix} \tag{3.2.28}$$

This perturbation indicates that the interaction from x_1 to x_3 is nonexistent. This can be shown by introducing a 3×3 so-called "interconnection

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

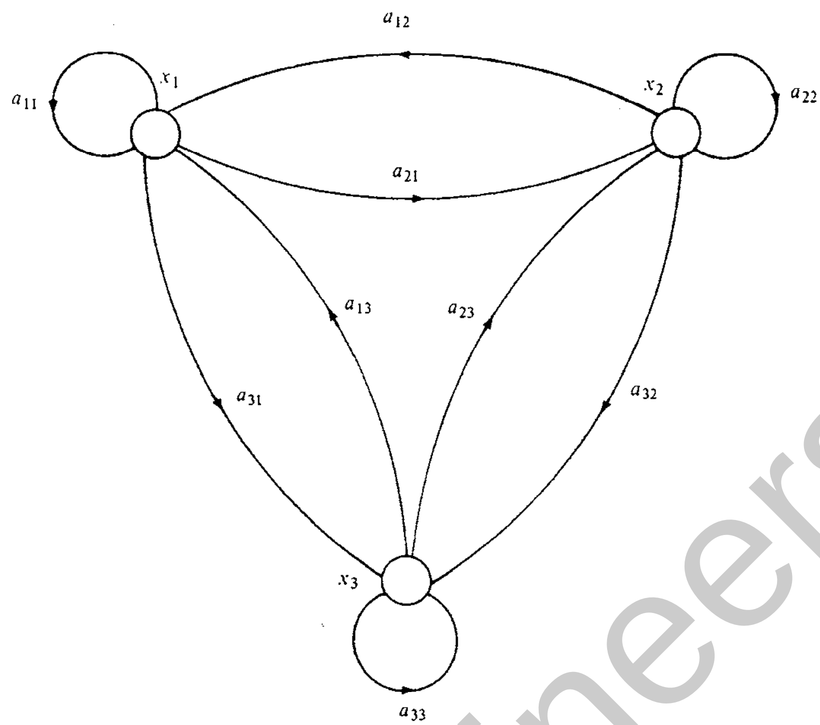


Figure 3.1 A diagraph representing a three-dimensional interconnected system.

matrix" G defined by

$$\tilde{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{3.2.29}$$

which is a binary matrix whose entry $\tilde{g}_{ij}$ is one if x_j "influences" x_i and zero otherwise. The diagraph corresponding to this structural perturbation is shown in Figure 3.2a. Similarly, other perturbations are possible by changing the unity elements of $\tilde{G}$ in (3.2.29) to zero. For example, if $a_{21} = a_{22} = a_{23} = a_{33} = 0$, $\tilde{G}$ is changed to

$$G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.2.30}$$

whose diagraph is shown in Figure 3.2b. If the three states are disconnected from each other, i.e.,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

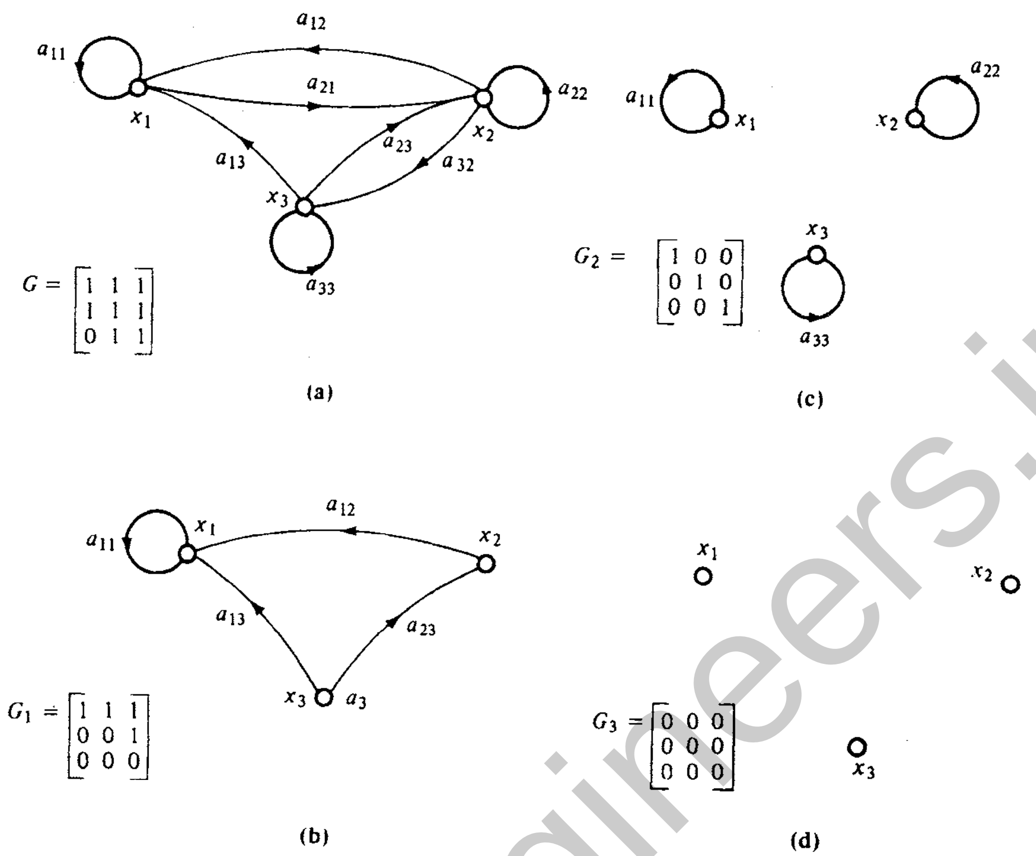


Figure 3.2 Structural perturbations of system of Figure 3.1.

$$G_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2.31}$$

it would correspond to the diagraph in Figure 3.2c. The case in which there is a "total disconnection" occurs when the interconnection matrix is null,

$$G_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.2.32}$$

as shown in Figure 3.2d. The interconnection matrix $\tilde{G}$ given by (3.2.29) is termed the "fundamental interconnection matrix" by Šiljak (1972a, 1978), since all other interaction matrices, such as G_i, $i = 1, 2, 3$, in (3.2.30)–(3.2.32) can be obtained from it by replacing unit elements with zeros.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

With this introduction to structural perturbation, the notion of connective stability is now considered.

Let the a_{ij}, $i, j = 1, 2, 3$, elements of (3.2.27) be represented by

$$a_{ij} = \begin{cases} -\alpha_i + \tilde{g}_{ij}\Delta a_{ii}, & i = j \\ \tilde{g}_{ij}\Delta a_{ij}, & i \neq j \end{cases} \tag{3.2.33}$$

where $\hat{g}_{ij}$ are the elements of the fundamental interconnection matrix G in (3.2.29), α_1 represent system's basic structure, and $\Delta\, a_{ij}$ correspond to the perturbations. The following definition can now be given.

Definition 3.7. The equilibrium point $x_e = 0$ is said to be "connectively stable" for a system if it is stable for all interconnection matrices G.

In order to achieve the connective stability, it suffices to prove the stability for the "fundamental interconnection matrix" $\tilde{G}$ (Šiljak, 1978). This is achieved by the so-called "comparison principle" (Birkhoff and Rota, 1962). This principle is described briefly as follows: Let a continuous function $y(t)$ satisfy the differential inequality

$$\dot{y}(t) \leq h(y,t) \tag{3.2.34}$$

where $h(y, t)$ satisfies a Lipschitz condition

$$|h(y,t) - h(z,t)| \leq L|y - z| \tag{3.2.35}$$

for any two pairs (y, t) and (z, t), and L is called the Lipschitz constant. Furthermore, let $w(t)$ be the solution of the differential equation

$$\dot{w}(t) = h(w,t) \tag{3.2.36}$$

with initial value $w(t_0) = y(t_0)$; then

$$y(t) \leq w(t), \quad t \geq 0 \tag{3.2.37}$$

3.2.3.b A Connective Stability Criterion

In this section a connective stability criterion due to Šiljak (1978) along with a numerical example is given. Let the model of a large-scale unforced system be

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{x} = F(x,t) \tag{3.2.38}$$

where x is the n-dimensional state vector, F is a function from $R^n \times \mathcal{T}$ to R^n, and $x_e = 0$ is a unique equilibrium point, i.e., $F(0, t) = 0$ for all $t \in \mathcal{T}$, $\mathcal{T}$ is the time interval $(t_0,+\infty)$. Let the system (3.2.38) be represented by a set of N interconnected subsystems

$$\dot{x}_i = F_i(x_i,\ \tilde{g}_{i1}x_1,\ldots,\tilde{g}_{iN}x_N,t) \tag{3.2.39}$$

where x_i is the n_i-dimensional state and $\tilde{g}_{ij}$ are elements of the $(N \times N)$-dimensional interaction matrix $G \in \tilde{G}$. There are various stability definitions within the framework of connectiveness. Below, various connective stability points are defined mathematically.

Definition 3.8. The equilibrium point $x_e = 0$ of system (3.2.38) is "connectively stable" if for every $\sigma > 0$, there is a number $\varepsilon > 0$ such that if

$$\|x_o\| < \varepsilon \tag{3.2.40}$$

then

$$\|x(t;x_o,t_o)\| < \sigma \tag{3.2.41}$$

for all interconnections $G \in \tilde{G}$.

In the above definition, ε is considered a function of α and t_o, i.e., e(t_o, α). However, if for each $\alpha > 0$ there is an $\varepsilon(\alpha)$ independent of t_o, condition (3.2.40) implies (3.2.41) for all $G \in \tilde{G}$; then $x_e = 0$ is said to be "uniformly connectively stable." It is noted that connective stability thus defined is a local concept because it is appropriate only near $x_e = 0$. In many applications it is not only necessary to have $x(t;,x_o,t_o)$ bounded but also to converge to the equilibrium point after some disturbance occurs. The following definition considers this situation.

Definition 3.9. The equilibrium point $x_e = 0$ of system (3.2.38) is "asymptotically connectively stable" if it is connectively stable and in addition there is a number $d > 0$ such that if $\|x_o\| < \delta$, then

$$\lim_{t \to +\infty} x(t;x_o,t_o) = 0 \tag{3.2.42}$$

for all interactions $G \in \tilde{G}$.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

This stability is also a local concept because one does not know a priori how small δ can be chosen. However, in many cases the focus is on δ being arbitrarily large and fixed at that value. Then the system is said to be "asymptotically connectively stable in the Large." This is the stability for which a criterion will be given.

Assume that system (3.2.38) is decomposed into N interconnected subsystems

$$\text{Css: } \dot{x}_i = f_i(x_i,t) + h_i(x,t), \quad i = 1,\dots,N \tag{3.2.43}$$

where $f_i(x_i,t)$ represents the ith decoupled or isolated subsystem

$$\text{Iss: } \dot{x}_i = f_i(x_i,t) \tag{3.2.44}$$

and $h_i(x_i,t)$ is the interaction of the ith subsystem with the remaining ones:

$$h_i(x,t) = h_i(g_{i1}x_1,\dots,g_{iN}x_n,t) \tag{3.2.45}$$

where g_{ij} are elements of the $N \times N$ interaction matrix G. The functions $f_i(x_i,t)$ and $g_i(x,t)$ are further assumed to satisfy $f_i(0,t) = 0$ and $g_i(0,t) = 0$ for all $t \in \mathcal{T}$. Before the stability criterion is given, the following definitions are considered.

Definition 3.10. A function $\psi(\varepsilon)$ belongs to a "comparison class" K_e if $\psi \in C(\mathcal{T})$, $\psi(0) = 0$, and if $\varepsilon_1 < \varepsilon_2$, then $\psi(\varepsilon_1) < \psi(\varepsilon_2)$. Moreover, if $\psi(\varepsilon) \to +\infty$ as $\varepsilon \to +\infty$, then $\psi(\varepsilon)$ is said to belong to the class K_∞.

Definition 3.11. A continuous function h_i: $\mathcal{T} \times R^n \to R^{n_i}$ is a member of the class K_n if there are bounded functions η_{ij}: $\mathcal{T} \times R^n \to R$ such that

$$\|h_i(x,t)\| \le \sum_{j=1}^{N} \tilde{g}_{ij}\eta_{ij}(x,t)\psi_{3j}\left(\|x_j\|\right) \tag{3.2.46}$$

for all t and x, where $\tilde{g}_{ij}$ are elements of the $(N \times N)$-dimensional fundamental interaction matrix $\tilde{G}$ and $\psi_{3j} \in K_\varepsilon$ is defined above.

Definition 3.12. An $N \times N$ constant "aggregate" matrix $S = (s_{ij})$ is defined by

$$s_{ij} = -\delta_{ij} + \gamma_i \tilde{g}_{ij}\beta_{ij} \tag{3.2.47}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where δ_{ij}, the Kronecker delta, is one for $i = j$ and zero otherwise; γ_i is a positive number; and nonnegative number β_{ij} is

$$\beta_{ij} = \max\,\{0, \operatorname{Sup}(\eta_{ij}(x,t))\} \tag{3.2.48}$$

The term "aggregate" stems from the theory of vector Lyapunov function υ; i.e., it can be shown that $\dot{\upsilon} \le S\upsilon$, where υ is an N-dimensional Lyapunov vector function.

One can now state the following theorem to provide the necessary conditions for asymptotic connective stability in the Large.

Theorem 3.3. *Let $\upsilon(x, t)$ be a continuously differentiable function which is locally Lipschitzian in x, $\upsilon(0, t) = 0$, and for each subsystem*

$$\begin{aligned} \psi_{1i}(\|x_i\|) \le \upsilon_i(x_i,t) \le \psi_{2i}(\|x_i\|) \\ \dot{\upsilon}_i(x_i,t) \le \psi_{3i}(\|x_i\|) \end{aligned} \tag{3.2.49}$$

for all t, x_i, $\psi_{1i}\ \psi_{2i} \in K_\infty$, and $\psi_{3i} \in K_\varepsilon$ (Definition 3.10). Furthermore, assume that interconnection functions $h_i(x, t) \in K_n$ (Definition 3.11) for all $i = 1, \ldots, N$. Then the quasi-dominant property of the $N \times N$ aggregate matrix S (Definition 3.12) implies that point $x_e = 0$ is asymptotically connectively stable in the Large for the composite system (3.2.38).*

For the complete proof of this theorem, see Šiljak (1978). Here a few comments regarding the properties of the Lyapunov function and the illustration of this theorem for linear systems will be given. The time derivative of the ith subsystem Lyapunov function can be shown to satisfy the inequality

$$\dot{\upsilon}_i(x_i,t)\big|_{(\mathrm{Lss})} \le \dot{\upsilon}_i(x_i,t)\big|_{(\mathrm{Iss})} + L_i\|h_i(x,t)\|, \quad i = 1,\ldots,N \tag{3.2.50}$$

where L_i is a Lipschitz constant, and Css and Iss represent composite subsystem (3.2.43) and isolated subsystem (3.2.44), respectively. In view of the condition (3.2.46) on $h_i(x, t)$, (3.2.50) can be rewritten as a differential inequality using the property (3.2.34):

$$\dot{\upsilon}(x,t) \le Sp\big(\upsilon(x,y)\big) \tag{3.2.51}$$

---

*An $n \times n$ matrix B is said to be quasi-dominant if there is a set of n numbers $c_i > 0$ such that $c_i b_{ii} > \Sigma_{j \ne i} c_i b_{ij}$, i = 1,...,n.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where S is the $N \times N$ aggregate matrix defined by (3.2.47). In (3.2.51) vector function $p: R_+^N \to R_+^N$ is defined by $p(\upsilon) \equiv \left[\psi_{31}(\psi_{11}(\upsilon_1)),\ldots,\psi_{3N}(\psi_{1N}(\upsilon_N))\right]^T$. Next, the interpretation of Theorem 3.3 for linear systems along with an illustrative example are presented.

Let the composite subsystem (3.2.43) be linear

$$\dot{x}_i = A_i x_i + \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} x_j. \quad i = 1,\ldots,N \tag{3.2.52}$$

where A_i and G_{ij} are $n_i \times n_i$ and $n_j \times n_j$ matrices, respectively. The isolated subsystem (3.2.44) will be

$$\dot{x}_i = A_i x_i \tag{3.2.53}$$

which is assumed to be stable; i.e., for any positive-definite matrix Q_i there is a positive-definite matrix P_i which is the solution of the Lyapunov equation

$$A_i^T P_i + P_i A_i + Q_i = 0 \tag{3.2.54}$$

The function $\upsilon_i(x_i) = \left(x_i^T P_i x_i\right)^{1/2}$ is a candidate for a Lyapunov function satisfying the following inequality similar to (3.2.49):

$$\lambda_m^{1/2}(P_i)\|x_i\| \leq \upsilon_i(x_i) \leq \lambda_M^{1/2}(P_i)\|x_i\| \tag{3.2.55}$$

where λ_m and λ_M are the minimum and maximum eigenvalues of the argument matrix, respectively. The Lyapunov function $\upsilon_i(x_i)$ is a Lipschitz function

$$|\upsilon_i(x_i) - \upsilon_i(y_i)| \leq L_i \|x_i - y_i\| \tag{3.2.56}$$

where the Lipschitz constant $L_i = \lambda_M(P_i)/\lambda_m^{1/2}(P_i)$ and the interconnection terms satisfy

$$\|G_{ij} x_i\| \leq \zeta_{ij} \|x_j\| \tag{3.2.57}$$

and $\zeta_{ij} = \lambda_M^{1/2}\left(G_{ij}^T G_{ij}\right)$. Furthermore, the inequality (3.2.50) becomes

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{v}_i(x_i)\Big|_{(3.2.52)} \le \dot{v}_i(x_i)\Big|_{(3.2.53)} + L_i \sum_{\substack{j=1\\ j\neq i}}^{N} \zeta_{ij}\left\|x_j\right\| \le -\psi_{3i}\left(\left\|x_i\right\|\right)$$

$$+2L_i \sum_{\substack{j=1\\ j\neq i}}^{N} \lambda_M^{1/2}(P_j)\lambda_m^{-1}(Q_i)\cdot\zeta_{ij}\psi_{3j}\left(\left\|x_j\right\|\right) \tag{3.2.58}$$

where $\psi_{3i}\left(\left\|x_i\right\|\right) = \frac{1}{2}\lambda_M^{-1/2}(P_i)\lambda_m(Q_i)\left\|x_i\right\|$. Finally, the $N \times N$ aggregate matrix S in (3.2.51) has the elements

$$s_{ij} = \begin{cases} -1, & i = j \\ 2L_i\lambda_M^{1/2}(P_j)\lambda_m^{-1}(Q_j)\zeta_{ij}, & i \neq j \end{cases} \tag{3.2.59}$$

The following example illustrates the application of Theorem 3.3 for the linear case.

Example 3.2.4. Let us reconsider the fifth-order system of Example 3.2.1. The A_i, G_{ij}, Q_i, and P_i, $i, j = 1, 2$, are given in (3.2.16) and (3.2.19). Check whether this svstem is asymptotically connectively stable in the Large.

SOLUTION: The system consists of two subsystems with a fundamental interconnection matrix $\tilde{G}$:

$$\tilde{G} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.2.60}$$

The eigenvalues of P_i, $i = 1,2$, are $\lambda(P_1) = (0.17427, 0.5130)$, $\lambda(P_2) = (0.1876, 0.2724)$. The values of Lipschitz constants L_i using (3.2.56) turn out to be $L_1 = 0.7162$ and $L_2 = 0.522$. The bounds ζ_{ij} are $\zeta_{12} = 0.484$ and $\zeta_{21} = 1.422$. The aggregate matrix S is obtained from (3.2.59), $s_{11} = s_{22} = -1$, $s_{12} = 0.1808$, and $s_{21} = 1.0634$. The quasi-dominancy condition is to have the leading principle minors of S, i.e.,

$$(-1)^k \begin{vmatrix} s_{11} & s_{12} & \cdots & s_{1k} \\ s_{21} & s_{22} & & s_{2k} \\ \cdots & & & \cdots \\ s_{k1} & s_{k2} & & s_{kk} \end{vmatrix} > 0 \tag{3.2.61}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

for all $k = 1, 2, \ldots, N$. In this case $(-1)^1 |s_{11}| = 1 > 0$ and

$$(-1)^2 \begin{vmatrix} -1 & 0.1808 \\ 1.0634 & -1 \end{vmatrix} = 0.8077 > 0 \qquad (3.2.62)$$

Thus, as expected, this system is asymptotically connectively stable in the Large.

3.3 Input-Output Stability Methods

The second method of investigating stability of large-scale systems is through the input-output (IO) technique. In this scheme, every subsystem is described by an operator or mathematical relation on a function space, and through the methods of functional analysis the system stability is studied (Moylan and Hill, 1978). The preliminary developments, definitions, motivations, and problem statement are considered in the next section. The input-output stability criterion itself is presented in Section 3.3.2.

3.3.1 Problem Development and Statement

As mentioned above, the input-output stability is based on function spaces. Let $U^{(\mu)}$ denote a normed space of R^{μ}-valued functions $u(t)$ of time; then the "extended normed space" $U_e^{(\mu)}$ is defined as $U_e^{(\mu)} = \{u | u_\tau \in U^{(\mu)} \quad \forall \tau \in R\}$ where u_τ is a "truncated function" defined by

$$u_\tau(t) = \begin{cases} u(t), & t < \tau \\ 0, & \text{otherwise} \end{cases} \qquad (3.3.1)$$

Let us define an operator $G: U^{(\mu)} \rightarrow U^{(\mu)}$ which is said to be "input-output stable" or simply IO stable if there are two nonnegative constants α and β such that

$$\|(Gu)_\tau\| \le \alpha \|u_\tau\| + \beta \quad \text{for all } \tau \in R \qquad (3.3.2)$$

where $\| \;\|$ is the norm in space $U^{(\mu)}$.

The gain of the operator G is defined by

$$\text{gain G} = \max_{u \in U_e^{(\mu)}} \left\{ \|(Gu)_\tau\| / \|u_\tau\| \right\} \qquad (3.3.3)$$

Let D, F, G, and H be operators from $U^{(\mu)}$ into itself such that $G0 = 0$ and $H0 = 0$ and consider the IO closed-loop system (Zames, 1966)

controlengineers.ir
@controlengineers

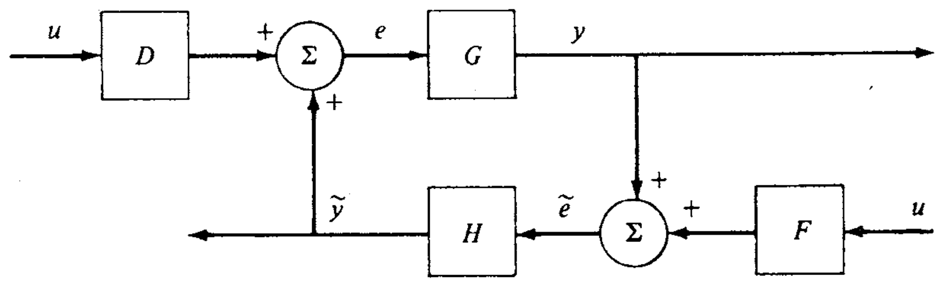


Figure 3.3 A composite input-output system.

$$
\begin{aligned}
e &= Du + \tilde{y}, \quad y = Ge \\
\tilde{e} &= Fu + y, \quad \tilde{y} = H\tilde{e}
\end{aligned} \tag{3.3.4}
$$

where u is the input to the system as shown in Figure 3.3. Let $E, \tilde{E}, Y,$ and $\tilde{Y}$ represent operators which map $u \in U^{(\mu)}$ into the solutions e, $\tilde{e}$, y, and $\tilde{y}$, respectively (Willems, 1970; Desoer and Vidyasagar, 1975). Then the IO system (3.3.4) is said to be IO-stable if the operators E, $\tilde{E}$, Y, and $\tilde{Y}$ are IO stable. Since in most large-scale systems one would be dealing with decompositions into N subsystems, it is useful to think of $U^{(\mu)}$ as the product of $U^{(\mu_i)}$, $i = 1, 2, \ldots, N$, such that $\mu = \mu_1 + \cdots + \mu_N$ and

$$
\|u\| = \left(\sum_{i=1}^{N} \|u_i\|^2 \right)^{1/2}, \quad u = \left(u_1^T, \ldots, u_N^T\right)^T, \quad u_i \in U^{(\mu_i)} \tag{3.3.5}
$$

Let the IO system (3.3.4) be represented by N decoupled subsystems

$$
\begin{aligned}
e_i &= D_i u + \tilde{y}_i, \quad y_i = G_i e_i \\
\tilde{e}_i &= F_i u + y_i, \quad \tilde{y}_i = H_i \tilde{e}_i + K_i \tilde{e}, \quad i = 1, \ldots, N
\end{aligned} \tag{3.3.6}
$$

where $e = \left(e_1^T, \ldots, e_N^T\right)^T$, $y = \left(y_1^T, \ldots, y_N^T\right)^T$, etc.; e_i $\tilde{e}_i$, y_i, and $\tilde{y}_i$ are members of $U_e^{(\mu_i)}$; G_i and H_i are operators from $U_e^{(\mu)}$ into itself; and D_i, F_i, and K_i are operators from $U_e^{(\mu_l)}$ into $U_e^{(\mu_i)}$. Araki (1978a) has made the as-

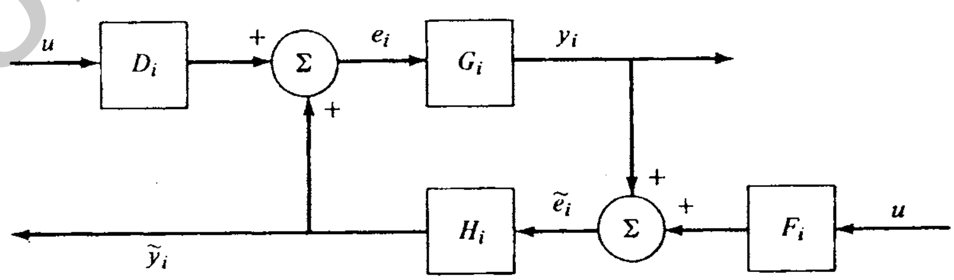


Figure 3.4 A subsystem for a composite input-output system.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

sumption that y_i in (3.3.6) depends on e_i as a result of simplification of the criterion to be discussed and its acceptability in most cases. If the operator K_i is removed from the second equation in (3.3.6), then the ith subsystem becomes isolated, as shown in Figure 3.4, and the operators K_i represent interconnections. In Section 3.7 it will be shown that this interpretation has a correspondence to the Lyapunov methods of Section 3.2.

Let $L_2^{(\mu)}$ denote the L_2 space of R^μ–valued function in which the inner product and norm are defined by

$$\langle u,v\rangle = \int_{-\infty}^{\infty} u^T(t)v(t)dt$$

$$\|u\|^2 = \langle u,u\rangle = \int_{-\infty}^{\infty} \{u_1^2(t)+u_2^2(t)+\cdots+u_\mu^2(t)\}dt \tag{3.3.7}$$

The motivation behind the IO stability of large-scale systems is the work of Zames (1966), who pioneered the IO stability of time-varying nonlinear systems. The following theorem provides the ground rules for it.

Theorem 3.4. *Small Gain Theorem. The IO system (3.3.4) is IO stable if*

$$(\text{gain } G)\cdot(\text{gain } H) < 1 \tag{3.3.8}$$

where the gain G is defined by (3.3.3). The following example illustrates the application of this theorem.

Example 3.3.1. Consider the IO system of Figure 3.5 with the real-valued function $|\psi(\sigma)| \le 0.5\sigma$. Check for its IO stability.

SOLUTION: Let G in Figure 3.5 be represented by G = diag(G_1, G_2), where G_i, i = 1, 2, are second-order rational functions of s. Since $Gu = (G_1u_1, G_2u_2)^T$ and $\|u\|^2 = \|u_1\|^2 + \|u_2\|^2$, the gain G is obtained from

$$\begin{aligned}\text{gain } G &= \max(\text{gain } G_1, \text{gain } G_2)\\ &= \max(0.1, 0.5) = 0.5\end{aligned} \tag{3.3.9}$$

The gain $H = \gamma$ is given by

$$\gamma = \max_{y\neq 0}\left\{\left(\left[y_1+0.2\psi(y_2)\right]^2+\left[0.2y_2+0.5\psi(y_1)\right]^2\right)\Big/\left(y_1^2+y_2^2\right)\right\}^{1/2} \tag{3.3.10}$$

However, since only an upper bound is available for ψ, one can replace $\psi(y_1)$ and $\psi(y_2)$ by $\pm 0.5y_1$ and $\pm 0.5y_2$, respectively, and try to find the maxi-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

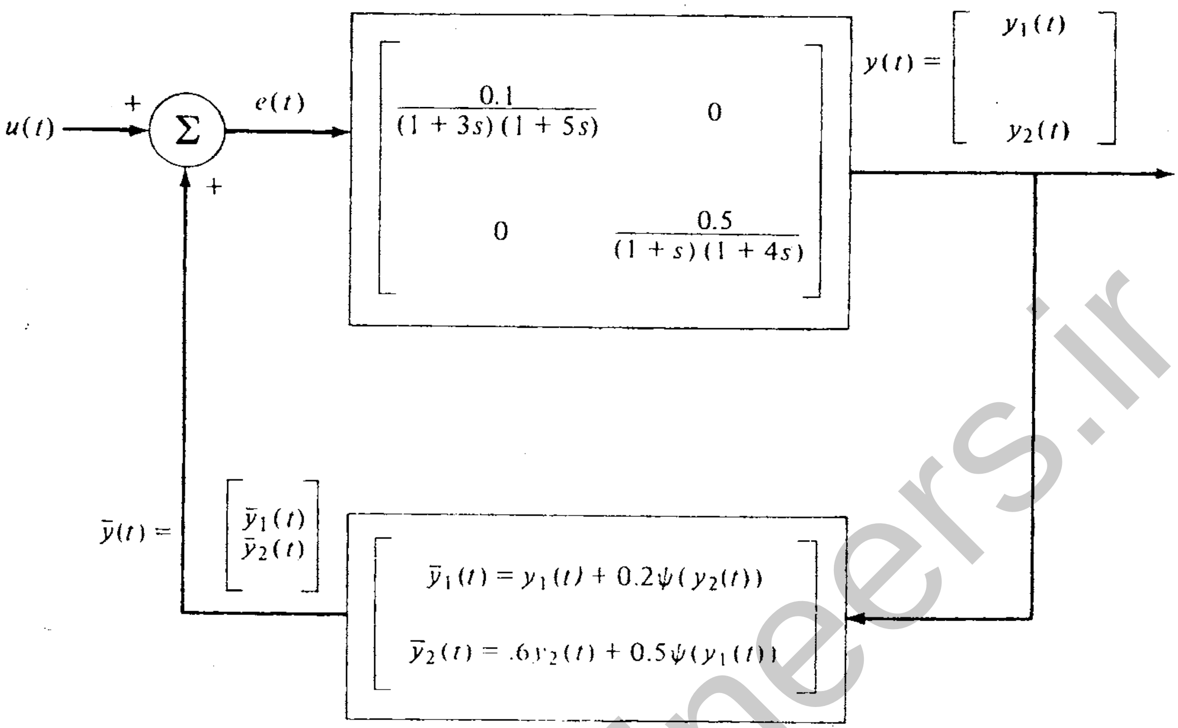


Figure 3.5 An input-output system for Example 3.3.1.

mum in (3.3.10). The gain G would then have an upper bound gain

$$\text{gain } H \le \sqrt{\lambda_M\left(H^T H\right)} \cong 1.2303 \tag{3.3.11}$$

where

$$H = \begin{bmatrix} 1 & 0.2 \\ 0.5 & 0.2 \end{bmatrix}$$

and λ_M corresponds to the maximum eigenvalue of its argument. Clearly condition (3.3.8) is satisfied, indicating that system of Figure 3.5 is IO stable. In general if condition (3.3.8) is not satisfied, one may consider a class of norms for $u \in U^{(u)}$,

$$\|u\|_D = \left\{d_1^2\|u_1\|^2 + \cdots + d_N^2\|u_N\|^2\right\}^{1/2} \tag{3.3.12}$$

and try to find a set of values $D = \text{diag}(d_1, d_2, \ldots, d_N)$ such that (3.3.8) holds.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Note that (3.3.5) is a special case of (3.3.12). Under this norm condition, gain condition (3.3.11) becomes

$$\text{gain}_D H \le \sqrt{\lambda_M \left(DHD^{-1}\right)^T \left(DHD^{-1}\right)} \tag{3.3.13}$$

Based on the above development the IO stability problem can now be stated as follows: For a large-scale IO system (3.3.4) which can be decomposed into N subsystems (3.3.6), under what conditions on the operators G_i, H_i, and K_i is the system IO stable? The IO stability can be conveniently checked by pursuing the following algorithm.

Algorithm 3.2. Input-Output Stability of Large-Scale Systems

Step 1: Investigate the properties of subsystems operators G_i, H_i, $i = 1, \ldots, N$.
Step 2: Find a functional bound for interconnection operators K_i, $i = 1, \ldots, N$.
Step 3: Obtain a set of parameters $d_1, \ldots, d_N$ such that condition *(3.3.8)* is satisfied.

The next section gives a theorem for IO stability criterion.

3.3.2 IO Stability Criterion

The following theorem provides necessary conditions for IO stability of a large-scale interconnected system.

Theorem 3.5. *The large-scale system (3.3.4) decomposed in (3.3.6) is IO stable if the following conditions are satisfied:*

1. *Gain* $G_i = \alpha_i < \infty$, $i = 1, \ldots, N$ (3.3.14)
2. *For a set of* $2N$ *nonnegative constants* β_{ij}, *the following norm condition holds:*

$$\left\|\left(H_i\tilde{e}_i + K_i\tilde{e}\right)_\tau\right\| \le \sum_{j=1}^{N} \beta_{ij} \left\|(e_i)_\tau\right\|, \quad \tilde{e} \in U_e^{(\mu)}, \quad \tau \in R \tag{3.3.15}$$

3. *The* $N \times N$ *matrix* $B = (b_{ij})$ *is an M-matrix*

$$b_{ii} = 1 - \alpha_i \beta_{ii}, \quad b_{ij} = -\alpha_i \beta_{ij} \quad i \ne j \tag{3.3.16}$$

PROOF: By condition 3 there are $d_i > 0$ such that $D^2 - \Omega^T D^2 \Omega$ is positive-definite where $D = \text{diag}(d_1, \ldots, d_N)$, $\Omega = (\omega_{ij})$, and $\omega_{ij} = \alpha_i\beta_{ij}$. This is due to the properties of M-matrices as presented by Definitions 3.3 and 3.4.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

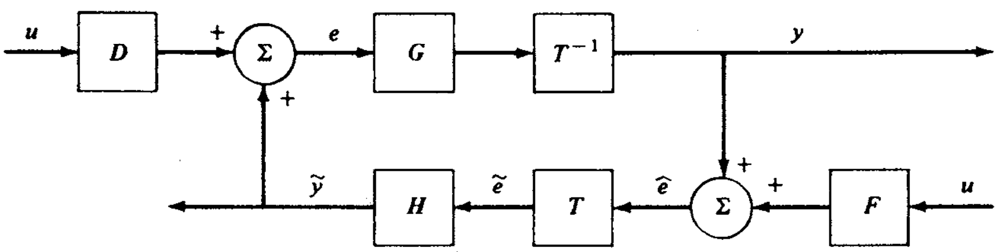


Figure 3.6 A transformed input-output system.

Now define the operator T by $y = Tu$, $y_i = \alpha_i u_i$ and transform the system, as shown in Figure 3.6. By condition 1, one obtains gain $_D(T^{-1}G) \le 1$. From (3.3.15), one can obtain

$$\begin{aligned}\sum_{i=1}^{N} d_i^2 \left\|(\tilde{y}_i)_\tau\right\|^2 &\le \sum_{i=1}^{N} d_i^2 \left\{\sum_{i=1}^{N} \alpha_i \beta_{ij} \left\|(\hat{e}_j)_\tau\right\|\right\}^2 \\ &= \sum_{j=1}^{N}\sum_{k=1}^{N}\left(\sum_{i=1}^{N} d_i^2 \omega_{ij}\omega_{ik}\right)\left\|(\hat{e}_j)_\tau\right\| \cdot \left\|(\hat{e}_k)_\tau\right\| \\ &< \sum_{i=1}^{N} d_i^2 \left\|(e_i)_\tau\right\|^2\end{aligned} \tag{3.3.17}$$

where $\tilde{y} = KT\hat{e}$, $\hat{e}_\tau \ne 0$ for all $\tau \in R$, and the last inequality is assured by the positive-definiteness of $D^2 - \Omega^T D^2 \Omega$. By (3.3.17) one can obtain gain $_D KT < 1$, and hence (3.3.8) is verified. Thus, system (3.3.4) is IO stable. Q.E.D. ■

The following example illustrates the IO-stability criterion.

Example 3.3.2. In this example a fourth-order system block diagram similar to that of Figure 3.5 with the following operators is reconsidered:

$$\begin{aligned}&G_1 = 0.50/\{(s+1)(s+2)\}, \quad G_2 = 6/\{(s+3)(s+4)\} \\ &H_1 = H_2 = 1, \quad K_1 = 1.5, \quad K_2 = 0.2\end{aligned} \tag{3.3.18}$$

Check the system's IO stability.

SOLUTION: The first condition of Theorem 3.5 yields gain $G_1 = \alpha_1 = 0.25$ and gain $G_2 = \alpha_2 = 0.5$. The second condition (3.3.15) on interconnections

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

provides, $\beta_{11} = \beta_{22} = 1$, $\beta_{12} = 1.5$, and $\beta_{21} = 0.2$. Using the α_i and, β_{ij} values, the matrix B of the third condition becomes

$$B = \begin{bmatrix} 0.75 & -0.375 \\ -0.10 & 0.50 \end{bmatrix} \tag{3.3.19}$$

which is clearly an M-matrix. Therefore, the system is IO stable.

Araki (1978a) has considered the $L_2^{(\mu)}$ space in place of $U^{(\mu)}$ and extended some of Zames's (1966) terminology to give an IO stability criterion based on "inside the sectors" conditions on scalar products of certain operations on $(u)_\tau$ using G_i and H_i. The interested reader is encouraged to refer to Araki's (1978a) details on other IO stability criteria.

3.4 Controllability and Observability of Composite Systems via Connectivity Approach

The earliest known work on this topic is due to Gilbert (1963). Consider a parallel composite system shown in Figure 3.7. The system is composed of two subsystems with the same input u, but their respective outputs are added to give the overall system output. Assuming that the eigenvalues of S_1 and S_2 are distinct (Perkins and Cruz, 1969), it can be deduced that a necessary and sufficient condition for system S to be controllable (or observable) is that both S_1 and S_2 be controllable (or observable).

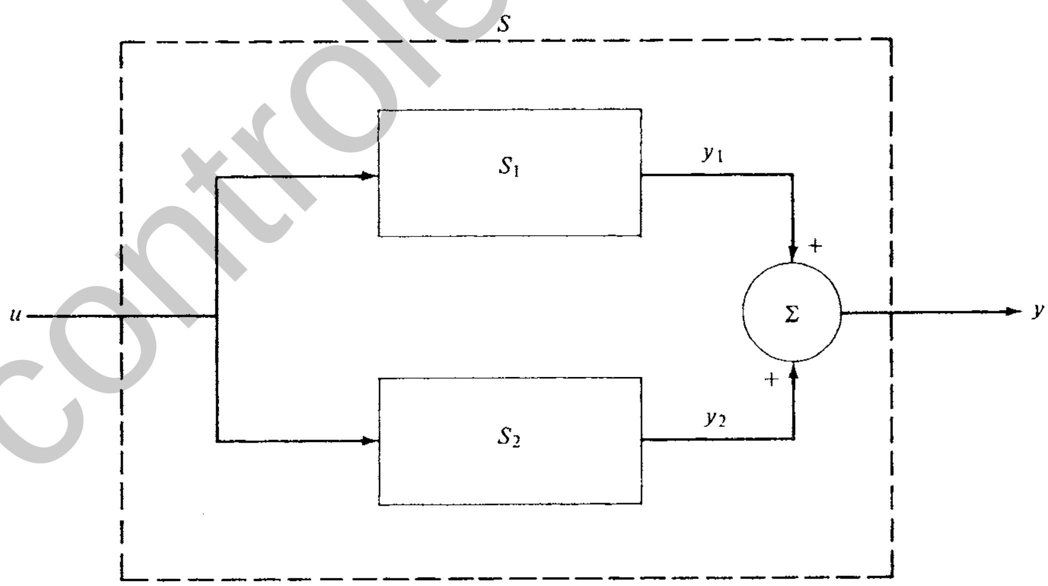


Figure 3.7 A parallel composite system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

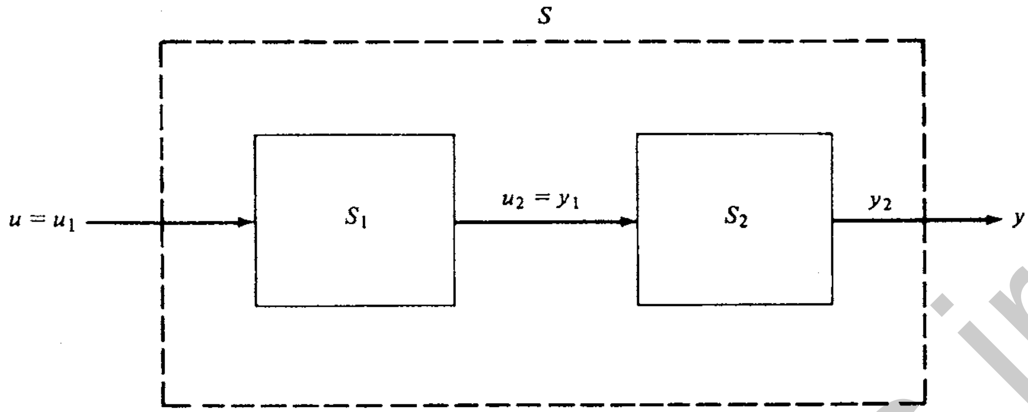


Figure 3.8 A series composite system.

Furthermore, the $n = n_1 + n_2$ eigenvalues of S are the union of n_1 eigenvalues of S_1 and n_2 eigenvalues of S_2. Next consider a series composite system shown in Figure 3.8. Here again, assuming that both subsystems have distinct eigenvalues, a necessary condition for the controllability (or observability) of S is that both S_1 and S_2 be controllable (or observable). Note that the condition is not sufficient, and it is possible for S_1 and S_2 to be both controllable and observable but for S to be neither. An inspection of Figure 3.8 indicates that under such conditions, the uncontrollable and unobservable modes of S must belong to S_2 and S_1, respectively. The following example illustrates this situation.

Example 3.4.1. Consider a second-order series composite system

$$\dot{x}_1 = -1/3\,x_1 + u \tag{3.4.1}$$

$$\begin{aligned} \dot{x}_2 &= -1/6\,x_1 - 1/6\,x_2 + u \\ y &= x_1 - x_2 - 6u \end{aligned} \tag{3.4.2}$$

Check the controllability and observability of the system.

SOLUTION: The two subsystems are clearly both controllable and observable. The composite system's controllability matrix is

$$P = [B \mid AB] = \begin{bmatrix} 1 & -1/3 \\ 1 & -1/3 \end{bmatrix} \tag{3.4.3}$$

while the observability matrix is

$$Q = [C^T \mid A^T C^T] = \begin{bmatrix} 1 & -1/3 \\ -1 & -1/3 \end{bmatrix} \tag{3.4.4}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Both matrices have rank of 1; hence the system (3.4.1) is neither controllable nor observable.

In this section the notion of connectability is defined and its role in the controllability and observability of composite systems is investigated (Davison, 1977). The approach is based on the application of graph theory (Deo, 1974). The following definitions and terminologies are needed for the controllability and observability conditions to follow.

3.4.1 Preliminary Definitions

Consider a denumerable set $x = (x_1, \ldots, x_n)$ and a mapping Λ of x into x; then a pair $M = (x, \Lambda)$ is said to constitute an "nth-order graph." The elements x_i, $i = 1, \ldots, n$, are referred to as "vertices" of the graph. The directed line segment from vertex x_i to x_j, denoted by $\upsilon_i = (x_i, x_j)$, is called an "arc" of the graph. The vertices x_i and x_j are called "initial" and "terminal," respectively. Either the pairs (x, υ) or (x, Λ) can represent a graph. Any line segment connecting a vertex x_i to a vertex x_j is said to be a "path." A closed path, i.e., one in which the initial and terminal vertices coincide, is said to be a "circuit." Now let us consider the following definitions.

Definition 3.13. The "arborescence" of root $x_1 \in x$ of a finite graph $M = (x, \upsilon)$ is itself a graph with the following properties:

1. x_1 is the terminal vertex of no arc.
2. Each $x_i \neq x_1$ is the terminal vertex of only one arc.
3. There is no circuit contained in the graph M.

A "branch arborescence" of root $x_1 \in x$ is a subset of an arborescence resulting from deleting all arcs and vertices of the arborescence except for those vertices and arcs along which are associated with a single arc directed from x_1. It must be emphasized that neither arborescence nor a branch of it are unique.

Definition 3.14. A "composite system" $\dot{x} = Ax + Bu$, $y = Cx$, denoted by $(C, A, B; N; n_1, \ldots, n_N)$, consists of N subsystems interconnected in arbitrary fashion with $x \in R^n$, $u \in R^m$, $y \in R^r$, and A defined by

$$A = \begin{bmatrix} A_1 & G_{12} & \cdots & G_{1N} \\ G_{21} & A_2 & \cdots & G_{2N} \\ \vdots & \vdots & & \vdots \\ G_{N1} & G_{N2} & \cdots & A_N \end{bmatrix} \tag{3.4.5}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

A "sparse composite system" $(C, A, B; N; n_1, \ldots, n_N)$ is a composite system with the following A matrix:

$$A = \begin{bmatrix} A_1^* & G_{12} & \cdots & G_{1N} \\ G_{21} & A_2^* & \cdots & G_{2N} \\ \vdots & \vdots & & \vdots \\ G_{N1} & G_{N2} & \cdots & A_N^* \end{bmatrix} \tag{3.4.6}$$

where $A_i^* = A_i + B_i K_i C_i$, $i = 1, 2, \ldots, N$, with (C_i, A_i, B_i) being an observable and controllable triplet. Furthermore, all the interconnection matrices G_{ij} are zero except for G_{ij}, $i = i_1, i_2, \ldots, i_p$ and $j = j_1, \ldots, j_q$ given by $G_{ij} = k_{ij}\alpha_i\beta_j^T$, $i = i_1, \ldots, i_p$, and $j = j_1, \ldots, j_q$ where k_{ij} is a nonzero scalar called the "ij-interconnection gain," α_i and β_j are nonzero $(n_i \times 1)$- and $(n_j \times 1)$-dimensional vectors, respectively. Note that a "general composite system" $(C, A, B; N; n_1, \ldots, n_N)$ has an interconnection matrix $G_{ij} = L_{ij}k_{ij}E_{ij}$, $i = i_1, \ldots, i_p$, $j = j_1, \ldots, j_q$, where L_{ij} and E_{ij} are nonzero matrices of appropriate dimensions and k_{ij} is defined earlier. The B and C matrices are assumed to be nontrivial and given by $B^T \triangleq (B_1^T, \ldots, B_N^T)$ and $C \triangleq (C_1^T, \ldots, C_N^T)$. It is usually possible to represent a general composite system $(C, A, B; N; n_1, \ldots, n_N)$ by a graph consisting of $N + 2$ vertices.

The following examples illustrate some of the above notions.

Example 3.4.2. For a graph shown in Figure 3.9, find an arborescence and a branch arborescence of it.

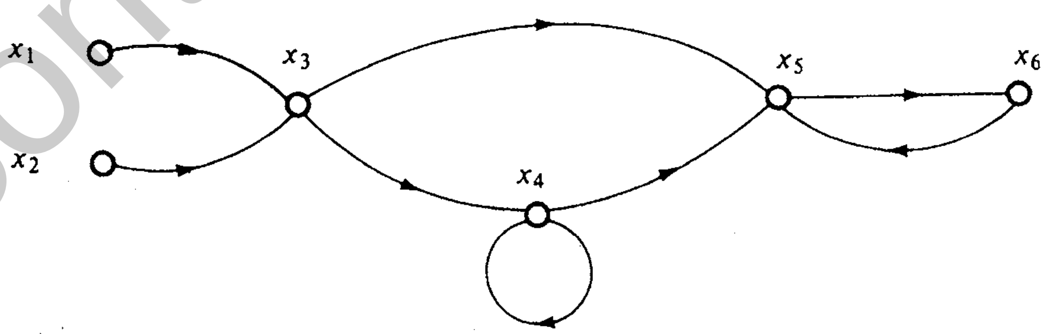
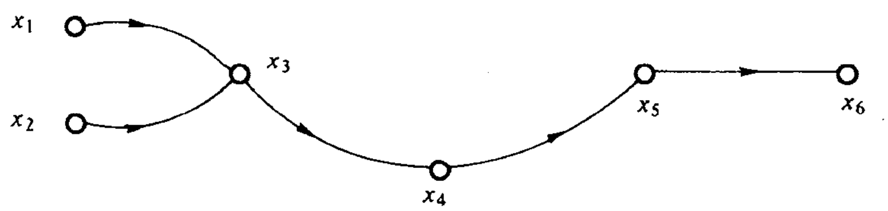


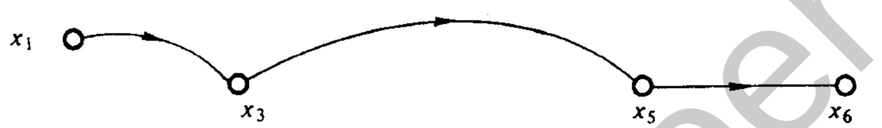
Figure 3.9 A graph for Example 3.4.2.

controlengineers.ir
@controlengineers



(a)

Figure 3.10a An arborescence of root x_4 for graph of Example 3.4.2 (Figure 3.9).



(b)

Figure 3.10b A branch arborescence of root x_3 for graph of Example 3.4.2 (Figure 3.9).

SOLUTION: This graph consists of six vertices and eight arcs. If an arborescence of root x_4 is desired, it is required to eliminate the self-loop of x_4 and one of the two arcs connecting x_3 to x_5 (one directly and the other through vertex x_4). One possible arborescence of x_4 is given in Figure 3.10a. A branch arborescence of root x_3 of the graph of Figure 3.9 is shown in Figure 3.10b.

Example 3.4.3. Consider a composite system described by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} A_1^* & G_{12} & G_{13} \\ G_{21} & A_2^* & 0 \\ G_{31} & 0 & A_3^* \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ B_3 \end{bmatrix} u \tag{3.4.7}$$

$$y = (0 \quad C_2 \quad 0) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.4.8}$$

It is desired to represent this system by a graph $M(C, A, B)$ and find an arborescence of root u for $M(A, B)$ and an arborescence of root y for $M^*(C, A)$. The last notation refers to an "inverse" graph which has all its arrows reversed in direction.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

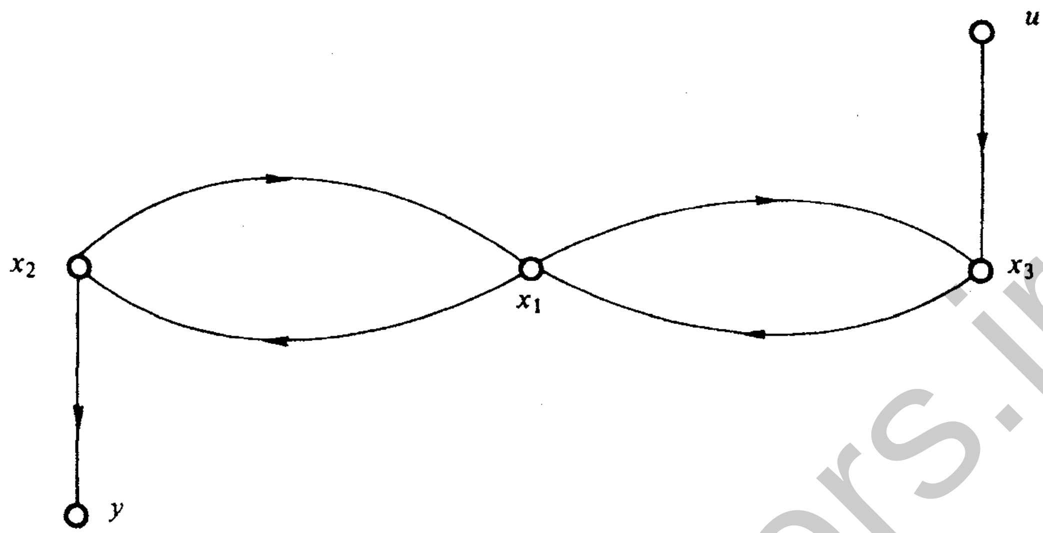


Figure 3.11 Graph $M(C, A, B)$ for system of Example 3.4.3.

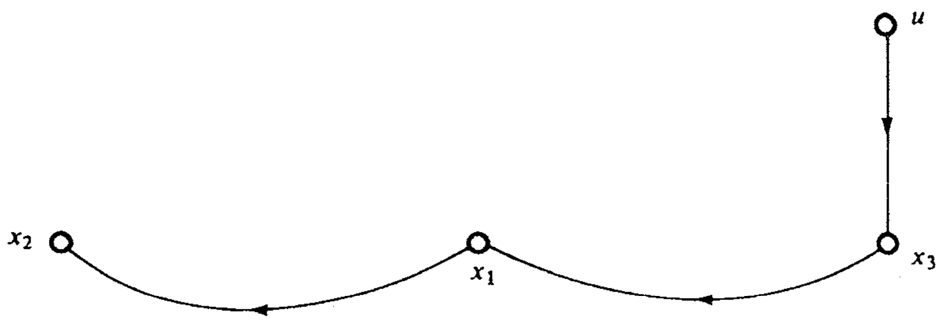
SOLUTION: The graph $M(C, A, B)$ for the system (3.4.4)–(3.4.8) has five (3 + 2) vertices and is shown in Figure 3.11. An arborescence of root y for graph $M(A, B)$ is obtained by first disconnecting all arcs terminating at y and following Definition 3.14. The result is shown in Figure 3.12a. To obtain an arborescence of root y for graph $M^*(C, A)$, it is necessary to reverse the arrows on all arcs, delete all new arcs terminating at u, and follow Definition 3.14. The resulting graph is shown in Figure 3.12b. Note that in this example, both arborescence graphs turn out to be unique as an exceptional case.

Definition 3.15. Let $M(A, B)$ denote the graph of composite system $(A, B; N; n_1, \ldots, n_N)$ and $M(C, A)$ represent the graph of composite system $(C, A; N; n_1, \ldots, n_N)$, where the vertices y and u are eliminated from the two graphs, respectively. Then a composite system $(A, B; N; n_1, \ldots, n_N)$ is called "input connectable" if there exists an arborescence, not necessarily unique, of root u for the graph $M(A, B)$. Furthermore, a composite system $(C, A; N; n_1, \ldots, n_N)$ is called "output connectable" if there exists an arborescence, not necessarily unique, of root y for the inverse graph $M^*(C, A)$. If a composite system is both input and output connectable, it is called "connectable."

The system of Example 3.4.3 is both input and output connectable and hence connectable. The notion of connectability, thus introduced, can be used to check the controllability and observability of a composite system.

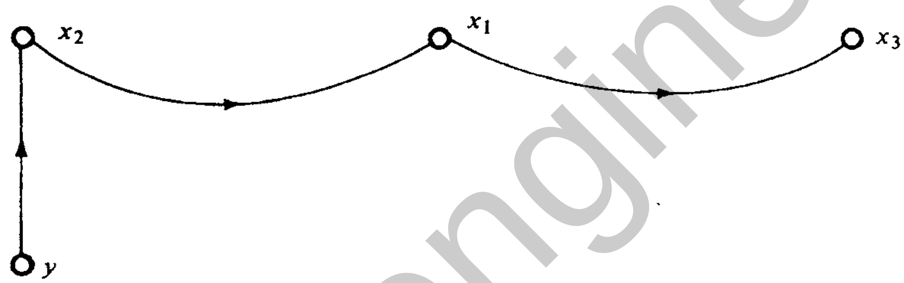
controlengineers.ir

controlengineers.ir
@controlengineers



(a)

Figure 3.12a An arborescence of root y for graph $M(A, B)$ of Example 3.4.3.



(b)

Figure 3.12b An arborescence of root y for graph $M^*(C, A)$ of Example 3.4.3.

3.4.2. Controllability and Observability Conditions

Based on the above preliminaries, the following theorems provide new conditions for the controllability and observability of composite systems.

Theorem 3.6. *Consider a sparse composite system* $(C, A, B; N; n_1, \ldots, n_N)$:

1. *If the system is connectable, then it is controllable and observable for almost all output gain matrices* K_i *and interconnection gains* k_{ij}.
2. *If the system is not connectable, then it is never controllable and observable.*

controlengineers.ir

controlengineers.ir
@controlengineers

A detailed proof of this theorem can be found in Davison (1977). The following theorem gives conditions for general composite systems.

Theorem 3.7. *Consider a general composite system* (C, A, B; N; n_1, ¼ ,n_N):

1. *If the system is connectable, then it is controllable and observable for almost all output gain matrices* K_i *and interconnection gains* K_{ij}.

2. *If the system is not connectable, then the general composite system is never controllable and observable.*

In the above theorems, it has been assumed that the composite system's subsystems (C_i, A_i, B_i) are both controllable and observable. Furthermore, a local output feedback is possible for each subsystem. Now let us consider the following example.

Example 3.4.4. Consider a system described by a graph shown in Figure 3.13. It is desired to check its controllability and observability.

SOLUTION: The arborescences of graphs $M(A, B)$ and $M^*(C, A)$ with respect to roots u and y are shown in Figure 3.14. This system is both input and output connectable, thus connectable. Therefore, by Theorem 3.12 this system is both controllable and observable.

The interpretations of Theorem 3.6 are given in Section 3.7.

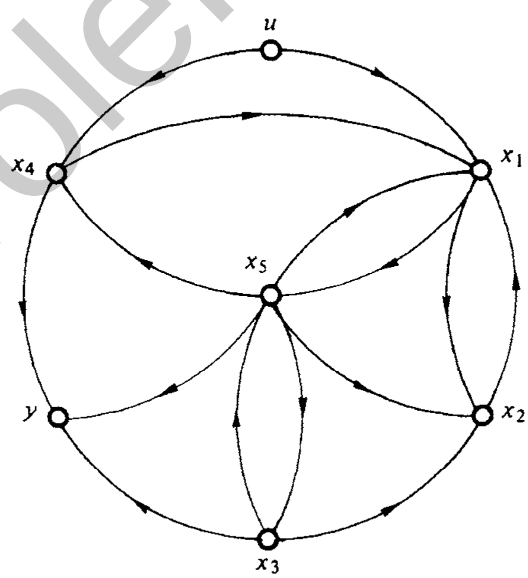
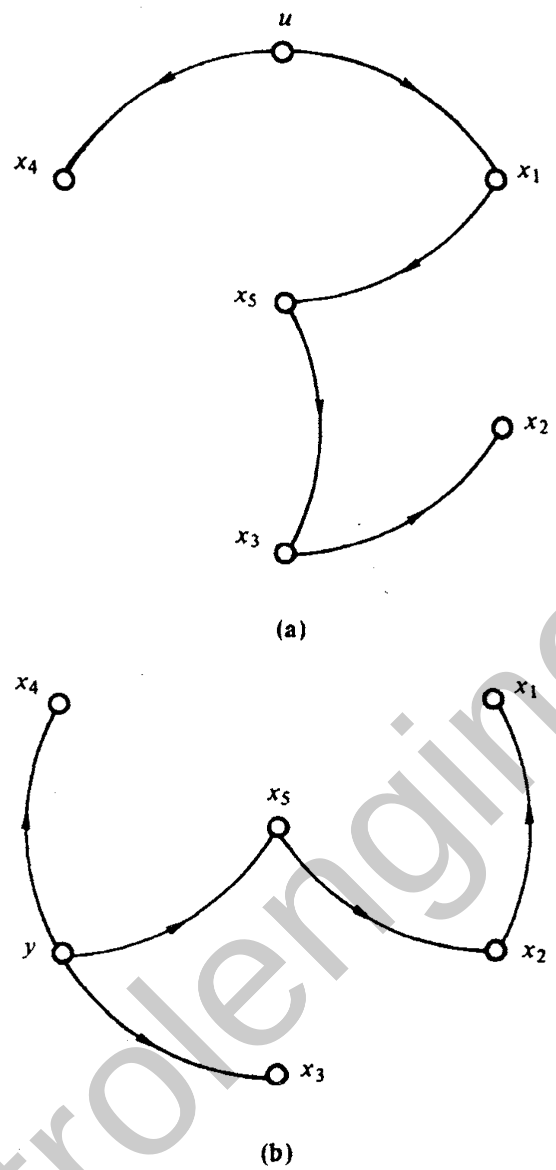


Figure 3.13 A graph for system of Example 3.4.4.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



u

y

Figure 3.14 Arborescences of graph $M(A, B)$ and $M^*(C, A)$ of Example 3.4.4 with respect to (a) root u and (b) root y.

3.5 Structural Controllability and Observability

An important special case of controllability of composite systems is " structural controllability" (Lin, 1974; Glover and Silverman, 1975; Shields and Pearson, 1976; Corfmat and Morse, 1976a), where the subsystems of the composite system all have order unity. Some of the important results in structural controllability as reported by Shields and Pearson (1976) are presented first, and then the role of system connectability in structural controllability will be presented.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

3.5.1 Structure and Rank of a Matrix

The concepts of structure and structural controllability were first introduced by Lin (1974) for single-input linear TIV systems using the graph-theoretic approach. The work of Lin (1974) has been extended by Shields and Pearson (1976).

Consider a linear TIV system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.5.1}$$

where $x(t) \in R^n$, $u \in R^m$, and A and B are matrices whose elements are either fixed (zero) or indeterminate (arbitrary). The two matrix pairs (A, B) and $\left(\tilde{A}\ \tilde{B}\right)$ are said to have the same "structure" if for every fixed (zero) entry of $(A\ B)$, the corresponding entry of $\left(\tilde{A}\ \tilde{B}\right)$ is also fixed (zero) and vice versa. A matrix A is said to be a "structural matrix" if it has fixed zeros in certain locations and arbitrary values in all other locations (Shields and Pearson, 1976). A linear system composed of a pair of structured matrices (A, B) is said to be a "structured system."

Two systems (A, B) and $\left(\tilde{A}, \tilde{B}\right)$ are said to be "structurally equivalent" if there is a one-to-one correspondence between the locations of fixed zero and nonzero entries of their respective matrices. The following definition can now be given.

Definition 3.16. A system (A, B) is said to be "structurally controllable" if it has a structurally equivalent system which is controllable in the usual sense, i.e., Equation (3.1.3).

Shields and Pearson (1976) have presented a detailed discussion on the structural properties of matrices which were investigated first by Frobenius (1912) and generalized by König (1931). Two of these properties most useful for the present discussion are "generic rank" and "form" of a matrix A of $n \times m$ $(n \le m)$ dimension.

Let $\Gamma(A, B)$ be a property of the matrices A and B which is assumed to hold for all data points except those which lie on an algebraic hypersurface in R^K, where K is the total number of arbitrary entries in A and B. More specifically, let $P[\lambda]$ be a polynomial in K variables $\lambda_1, \ldots, \lambda_K$. Let a set of polynomials $\phi \in P[\lambda]$; then a "variety" $C \subset R^K$ is the set of common zeros of polynomials $\phi_1, \phi_2, \ldots, \phi_L$, where L is some finite number. A variety V is said to be "proper" if $V \ne R^K$ and "nontrivial" if $V \ne \varnothing$, an empty set. A property Γ is "generic" relative to a proper variety V if such a variety exists so that $\ker(\Gamma) \subset V$, where $\ker(\cdot)$ is the kernel of A. The kernel of A is also known as the "null-space" of A denoted by II (A), the space of all

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

solutions of $AX = 0$. As an example of the generic property, consider the generic rank of an $n \times m$ matrix A consisting of all arbitrary entries. Let the property $\Gamma(a)$ be defined

$$\Gamma(a) = \begin{cases} 1 & \text{if rank } (A) = n \\ 0 & \text{if rank } (A) < n \end{cases} \tag{3.5.2}$$

where $a \in R^{n \cdot m}$, Moreover, let ϕ be a polynomial in $n \cdot m$ arbitrary variables defined as the sum of squares of all possible minors of A of order n. Shields and Pearson (1976) state that a structured matrix A has a generic rank denoted by $r(g)$ if there exists a proper variety $V \in R^K$ such that all points $a \in R^K$ for which rank $A \neq r$ lie on V. In terms of the property $\Gamma(a)$ in (3.5.2) and the polynomial ϕ, this indicates that if $a \in \ker(\Gamma)$, then $\phi(a) = 0$, and, furthermore, the proper variety V is defined by $V = \{a | \phi(a) = 0\}$. It is thus clear that the maximum rank of an $n \times m$ matrix A is the minimum of n and m. The following definition gives the form of a matrix.

Definition 3.17. An $n \times p$ matrix A is said to be of form (r) for $1 \leq r \leq n$ if for some k, $p - r \leq k \leq p$, A contains a zero submatrix of dimension $(n + p - k - r + 1) \times k$. The following example illustrates this definition.

Example 3.5.1. Consider the matrices

$$A_1 = \left[\begin{array}{cc|ccccc|c} x & x & x & x & 0 & x & x & x \\ 0 & x & x & 0 & 0 & x & x & x \\ \hline x & x & 0 & 0 & 0 & 0 & 0 & x \\ x & x & 0 & 0 & 0 & 0 & 0 & x \\ \hline x & x & x & x & x & x & x & x \end{array}\right], \quad A_2 = \left[\begin{array}{cc|ccc|c} x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & 0 & x & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & x \end{array}\right] \tag{3.5.3}$$

where x denotes nonzero elements. Find the forms of A_1, and A_2 matrices.

SOLUTION: For A_1 matrix, $n = 5$, $p = 8$, and $k = 5$. Then the form of A_1 turns out to be $r = 7$, since a zero submatrix of order $(n + p - k - r + 1) \times k = 2 \times 5$ is contained in A_1, as shown by dashed lines in (3.5.3). As for matrix A_2, there are two zero submatrices of orders 2×4 and 4×2, as shown in (3.5.3). For both cases the form turns out to be $r = 7$. In fact, for any diagonal matrix such as A_2 of rank n, the form is $n + 1$.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The form property of a matrix can be used to find an upper bound on the rank of any matrix.

Lemma 3.1. *For every entry* $a \in R^{n \cdot m}$ *in an* $n \times m$ *matrix* A, *rank* $A < r$, $1 \leq r \leq n$, *if and only if* A *has a form* (r).

A simple proof of this lemma has been presented by Shields and Pearson (1976). Now consider two matrices A and B of dimensions $n \times m$ and $n \times m$, respectively. The following definitions will be needed for the conditions on structural controllability set forth in Section 3.5.2.

Definition 3.18. For the $n \times (n + m)$ augmented matrix $(A\ B)$, let there be a permutation matrix P satisfying

$$P^T(A\ B) = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_1 & 0 & 0 \\ A_2 & A_3 & B_2 \end{bmatrix} \tag{3.5.4}$$

where $\dim(A_1) = r \times r$, $1 \leq r \leq n$; then the matrix $(A\ B)$ is said to have a "form I." Furthermore, if $(A\ B)$ contains an $((n + m + 1 - k) \times k)$-dimensional zero submatrix for $m < k \leq n + m$, i.e., if $(A\ B)$ has a form (n) in the sense of Definition 3.18, then the matrix $(A\ B)$ is said to have a "form II."

3.5.2 Conditions for Structural Controllability

The importance of the form of a matrix is in determining whether a system is structurally controllable. One of the initial results along the lines of system controllability under system matrices perturbations is due to Lee and Markus (1967) and states that if a system $\dot{x} = \tilde{A}x + \tilde{B}u$ is controllable, then there is an $\varepsilon > 0$ such that every linear $\dot{x} = Ax + Bu$ system with $\|A - \tilde{A}\| < \varepsilon$ and $\|B - \tilde{B}\| < \varepsilon$ is also controllable. Shields and Pearson (1976) have generalized this result in terms of a proper variety $V \subset R^{N+M}$, where N and M are the number of arbitrary elements of A and B, respectively. The above developments do characterize structural controllability but do not give conditions for determining it. The first condition along this line is one of Lin's (1974) results given below.

Lemma 3.2. *The system* (A, b) *is structurally uncontrollable if and only if the* $(n \times (n + 1))$*-dimensional matrix* $[A\ b]$ *has either form I or form II.*

The following theorem is the extension of the above result for a multi-input system.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Theorem 3.8. *The system* (A, B) *is structurally uncontrollable if and only if the* $(n \times (n + m))$*-dimensional matrix* $[A\ B]$ *has either form I or form II.*

A proof of this theorem, based on four lemmas concerning various properties of structured matrices is given by Shields and Pearson (1976).

The application of the above theorem to numerical examples requires the recognition of forms I and II. The calculation of form II is simple enough; however, the condition of form I may prove to be a tedious process and sometimes give misleading results.

A generic rank computational algorithm based on combinatorial mathematics (Liu, 1968) due to Prescott and Pearson (1981) is given next. The basic idea is that the generic rank of an $n \times m$ structured matrix A is equivalent to determining the maximum number of "nontaking rooks" that can be placed on an $n \times m$ chessboard, where the zeros of the matrix are interpreted as forbidden positions. A discussion on this theory can be found in Liu (1968).

Algorithm 3.3. Generic Rank of a Structured Matrix $r(g)$

Step 1: Eliminate any rows or columns of matrix A which contain only zeros. Let the dimension of the resulting matrix be $N \times M$.

Step 2: If $\text{Min}(N, M) = 1$, the generic rank of A, $gr(A) = 1(g) = 1$. If $\text{Min}(N, M) = 0$, then $gr(A) = 0(g)$.

Step 3: Identify a nonzero entry in A with the maximum number of zeros in its row and column. This entry is not necessarily unique.

Step 4: Compute the generic rank of A_1, which is the original matrix A excluding the row and column corresponding to the nonzero entry identified in Step 3. If $gr(A_1) = \text{Min}(N, M) - 1$, then $gr(A) = 1 + gr(A_1)$

Step 5: Compute the generic rank of A_2, which is the original matrix A after setting the identified entry of Step 3 to zero. Then $gr(A) = r(g) = \text{Max}[1 + gr(A_1), gr(A_2)]$.

The following example illustrates Algorithm 3.3.

Example 3.5.2. Consider a (6×9)-dimensional matrix

$$A = \begin{bmatrix} x & x & 0 & 0 & 0 & 0 & x & 0 & x \\ x & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & x & x & 0 & x & x & x \\ x & x & x & 0 & x & x & 0 & x & 0 \\ x & x & x & x & x & x & x & x & x \end{bmatrix} \tag{3.5.5}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where x represents nonzero entries. It is desired to find the generic rank of A using Algorithm 3.3.

SOLUTION: After eliminating row 3 and following Step 3 of the algorithm, entry 2, 3 results in the following A_1 and A_2 matrices:

$$A_1 = \begin{bmatrix} x & x & 0 & 0 & 0 & x & 0 & x \\ x & x & x & x & 0 & x & x & x \\ x & x & 0 & x & x & 0 & x & 0 \\ x & x & x & x & x & x & x & x \end{bmatrix},$$

$$A_2 = \begin{bmatrix} x & x & 0 & 0 & 0 & 0 & x & 0 & x \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & x & x & 0 & x & x & x \\ x & x & x & 0 & x & x & 0 & x & 0 \\ x & x & x & x & x & x & x & x & x \end{bmatrix} \tag{3.5.6}$$

The encircled entries denote those identified at Step 3 of the algorithm. Note also that $n = 6$, $N = 5$, and $m = M = 9$. Repeating Steps 3 and 4 for matrix A_1, one obtains the following reduced matrices out of A_1:

$$A_1^{(1)} = \begin{bmatrix} x & x & x & x & 0 & x & x \\ x & x & 0 & x & x & x & 0 \\ x & x & x & x & x & x & x \end{bmatrix}, \quad A_1^{(2)} = \begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \end{bmatrix} \tag{3.5.7}$$

Similarly, the following three reduced matrices result from repeated application of Steps 3 and 4 to A_2:

$$A_2^{(1)} = \begin{bmatrix} x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & x & x & 0 & x & x \\ x & x & x & 0 & x & x & 0 & x \\ x & x & x & x & x & x & x & x \end{bmatrix},$$

$$A_2^{(2)} = \begin{bmatrix} x & 0 & x & x & 0 & x & x \\ x & x & 0 & x & x & 0 & x \\ x & x & x & x & x & x & x \end{bmatrix}, \quad A_2^{(3)} = \begin{bmatrix} x & x & x & 0 & x & x \\ x & x & x & x & x & x \end{bmatrix} \tag{3.5.8}$$

Thus,

$$gr(A_1) = 2 + gr\left(A_1^{(2)}\right) = 4(g) = 4$$
$$gr(A_2) = 3 + gr\left(A_1^{(3)}\right) = 5(g) = 5$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Since $gr(A_1) = \mathrm{Min}(N, M) - 1$, by Step 5 of the algorithm, the generic rank of A is given by

$$gr(A) = \mathrm{Max}\left[1 + gr(A_1),\ gr(A_2)\right] = \mathrm{Max}(1 + 4,5) = 5(g) = 5 \tag{3.5.9}$$

The above algorithm can be used to determine the structural controllability of a system by utilizing the following definition and a lemma due to Rosenbrock (1970).

Definition 3.19. The $n_2 \times$ n $(n + m - 1)$-dimensional "extended controllability matrix" of a system is defined by

$$G_c = \begin{bmatrix} B & I & 0 & 0 & & & \vdots & & 0 \\ 0 & -A & B & I & 0 & 0 & & \vdots & 0 \\ 0 & 0 & 0 & -A & B & I & 0 & \vdots & 0 \\ \vdots & & & & & & \ddots & & \vdots \\ 0 & & & \cdots & & & 0 & -A & B \end{bmatrix} \tag{3.5.10}$$

Lemma 3.3. *For a data point $a \in R^{\tilde{N}+\tilde{M}}$, the associated system (A,B) is controllable if and only if*

$$\mathrm{rank}(G_c) = n^2 \tag{3.5.11}$$

The parameters $\tilde{N}$ and $\tilde{M}$ denote the number of arbitrary elements of A and B, respectively.

Using the above result, one can then state the following theorem for structural controllability of large-scale systems.

Theorem 3.9. *The following equivalent conditions hold simultaneously for the system (A, B):*

1. *System (A, B) is structurally uncontrollable.*
2. *The matrix [A B] is of form I or form II.*
3. *The matrix G_c is of form (n^2).*

A proof of this theorem can be found in Shields and Pearson (1976).

The use of the above lemma and theorem are illustrated by two examples.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Example 3.5.3. Consider a system

$$(A,\ B)=\left(\begin{bmatrix} x & 0 \\ x & x \end{bmatrix},\ \begin{bmatrix} x \\ 0 \end{bmatrix}\right) \tag{3.5.12}$$

It is desired to check whether it is structurally controllable.

SOLUTION: The structural controllability of this system can be checked in two different ways using Lemma 3.3 and Theorem 3.9. The extended controllability matrix G_c in (3.5.10) is given by

$$G_c=\begin{bmatrix} B & I & 0 \\ 0 & -A & B \end{bmatrix}=\begin{bmatrix} x & x & 0 & 0 \\ 0 & 0 & x & 0 \\ 0 & x & 0 & x \\ 0 & x & x & 0 \end{bmatrix} \tag{3.5.13}$$

where the unity and negative entries are still shown by an x. Using Algorithm 3.3, the rank of G_c is

$$\begin{aligned}\text{rank}(G_c) &= \text{Max}[1 + gr(G_c)_1,\ gr(G_c)_2] \\ &= \text{Max}[1 + 3,4] = 4 = n^2(g) = n^2 \end{aligned}\tag{3.5.14}$$

which implies that the system is structurally controllable by Lemma 3.3. Alternatively, the form of G_c can be easily found to be 5 and not $n^2 = 4$, which would indicate that the system is structurally controllable by Theorem 3.9. The above results are of course immediate from general condition (3.1.3) as well.

Example 3.5.4. Consider a third-order system,

$$(A,\ B)=\left(\begin{bmatrix} x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix}\right) \tag{3.5.15}$$

and repeat for structural controllability.

SOLUTION: An inspection of (3.5.15) indicates that the system is not controllable in the general sense. However, this result can be checked

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

through Lemma 3.3 and Theorem 3.9. The extended controllability matrix G_c becomes

$$G_c = \left[\begin{array}{ccccc|cccc} x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & x & x & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{3.5.16}$$

There is a 5×5 zero submatrix in G_c. The dimension of this zero submatrix is $\{n^2 + n(n + m - 1) - k + 1 - r\} \times k$. Since $n = 3$, $m = 1$, and $k = 5$, $r = 9 + 3(3 + 1 - 1) + 1 - 5 - 5 = 9 = n^2$. Thus, G_c has a form (n^2), which indicates that the system is structurally uncontrollable by Theorem 3.9. Alternatively, application of Algorithm 3.3 provides a rank

$$\begin{aligned} \text{rank}(G_c) &= \text{Max}[1 + gr(G_c)_1, gr(G_c)_2] \\ &= \text{Max}[1 + 7, 8] = 8 = n^2 - 1(g) = n^2 - 1 \end{aligned} \tag{3.5.17}$$

which determines, by Lemma 3.3, that the system is structurally uncontrollable.

3.5.3 Structural Controllability and Observability via System Connectability

The notion of system connectability was introduced in Section 3.4.1, where controllability and observability of general composite systems were checked by utilizing this property. In this section, the notion of structural observability is first introduced and then conditions for both structural controllability and observability are given in terms of connectability (Davison, 1977).

Definition 3.20. A system (C, A) is said to be "structurally observable" if it has a structurally equivalent system which is observable in the usual sense, i.e., Equation (3.1.4).

Note that this definition is the dual of Definition 3.16. The following

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

theorem and corollary provide conditions for structural controllability and observability.

Theorem 3.10. *A system (A, B) is structurally controllable if and only if the following two conditions hold simultaneously:*

1. $\mathrm{rank}(A, B) = n(g)$ (3.5.18)

2. $(A, B; n; 1,1, \ldots ,1)$ *is input-connectable.*

An immediate corollary of this theorem for structural observability is given next.

Corollary 3.1. A system (C, A) is structurally observable if and only if the following two conditions hold simultaneously:

1. $\mathrm{rank}\begin{bmatrix} A \\ C \end{bmatrix} = n(g)$.

2. $(C, A; n; 1, 1, \ldots ,1)$ is *output-connectable.* 3.5.19

Note that one can combine the conditions (3.5.18) and (3.5.19) for both properties; i.e., a system (C, A, B) is both structurally controllable and observable if and only if the two generic rank conditions in (3.5.18) and (3.5.19) both hold and the system $(C, A; n; 1, 1, \ldots ,1)$ is connectable. This is one of the results obtained by Davison (1977). In the above conditions, the generic rank can be obtained by either Algorithm 3.3 or any other appropriate algorithm. One such algorithm has been suggested by Davison (1977). However, it has been argued (Morari *et al.*, 1978) that the generic rank algorithms, such as those by Davison (1977), are not appropriate because standard rank-finding routines tend to destroy the structure (i.e., lose track of the true zeros) of structured matrices. The following example illustrates the application of Theorem 3.10 and Corollary 3.1.

Example 3.5.5. Consider the seventh-order system

$$(C, A, B) = \left(\begin{bmatrix} x & x & 0 & 0 & 0 & 0 & x \\ 0 & 0 & x & 0 & x & x & 0 \end{bmatrix} \begin{bmatrix} x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & x & 0 & x & 0 & 0 & x \\ x & 0 & 0 & x & 0 & 0 & 0 \\ 0 & x & 0 & x & x & 0 & x \\ 0 & 0 & x & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x & x & x \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ x & 0 \\ x & x \\ 0 & 0 \\ x & 0 \\ 0 & x \\ x & 0 \end{bmatrix} \right)$$

(3.5.20)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It is desired to check for its structural controllability or observability.

SOLUTION: The system's graph can be seen to have an arborescence with respect to inputs (u_1, u_2) and outputs (y_1, y_2); hence, it is connectable. The remaining conditions are ranks of (A,B) and $\begin{bmatrix} A \\ C \end{bmatrix}$ which can be easily checked by Algorithm 3.3. The results of the two applications are

$$\begin{aligned} \operatorname{rank}(A,\, B) &= \operatorname{Max}\left[1 + gr(AB)_1,\ gr(AB)_2\right] = n(g) = n \\ \operatorname{rank}\begin{bmatrix} A \\ C \end{bmatrix} &= \operatorname{Max}\left[1 + gr\begin{pmatrix} A \\ C \end{pmatrix}_1,\ gr\begin{pmatrix} A \\ C \end{pmatrix}_2\right] = n(g) = n \end{aligned} \tag{3.5.21}$$

which imply that the system (3.5.20) is both structurally controllable and observable.

3.6 Computer-Aided Structural Analysis

In this section the standard criteria for the controllability, observability, and stability of linear systems are briefly discussed prior to a number of CAD examples.

3.6.1 Standard State-Space Forms

The state equation of a system can be transformed into several standard forms, also known as canonical forms. These forms are generally applicable to single-input single-output systems. In this section, we consider four commonly used standard forms.

Controllable Canonical Form

The state equation

$$\begin{aligned} \dot{x} &= \begin{pmatrix} 0 & 1 & & 0 & 0 \\ 0 & 0 & & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & & 0 & 1 \\ -p_o & -p_1 & & -p_{n-2} & -p_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(t) \\ y &= (\gamma_o \gamma_1 \cdots \gamma_{n-1}) x \end{aligned} \tag{3.6.1}$$

is said to be in the controllable canonical form, where $\gamma_0, \ldots, \gamma_{n-1}$ are constant scalars. The scalar transfer-function of this form is

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$g(s) = \frac{Y(s)}{U(s)} = c(sI - A)^{-1} b$$

$$= (\gamma_0 \gamma_1 \cdots \gamma_{n-1}) \begin{pmatrix} s & -1 & & 0 & 0 \\ 0 & s & & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & & s & -1 \\ p_0 & p_1 & & p_{n-2} & s + p_{n-1} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

It is easy to verify that $(sI - A)^{-1}b$ is equal to the cofactors of the last row of $\frac{1}{p(s)}(sI - A)$, where

$$p(s) = \det(sI - A) = s^n + p_{n-1}s^{n-1} + \cdots + p_1 s + p_o$$

The cofactors of the last row of $\frac{1}{p(s)}(sI - A)$ are found to be

$$(sI - A)^{-1} b = \frac{1}{p(s)} \begin{pmatrix} 1 \\ s \\ s \\ \vdots \\ s^{n-1} \end{pmatrix}$$

Hence

$$g(s) = c(sI - A)^{-1} b = \frac{\gamma_{n-1}s^{n-1} + \cdots + \gamma_1 s + \gamma_0}{s^n + p_{n-1}s^{n-1} + \cdots + p_1 s + p_0} \tag{3.6.2}$$

Note that in the controllable canonical form, the coefficient of the characteristic polynomial appears as the elements of the last row of the system matrix A.

Observable Canonical Form

The state equation

$$\dot{x} = \begin{pmatrix} 0 & 0 & & 0 & -p_0 \\ 1 & 0 & & 0 & -p_1 \\ 0 & 1 & \cdots & 0 & -p_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & 0 & -p_{n-2} \\ 0 & 0 & & 1 & -p_{n-1} \end{pmatrix} x + \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} \end{pmatrix} u \tag{3.6.3}$$

$$y = (00 \cdots 01)x$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is said to be in the observable canonical form, where $\beta_0, \ldots, \beta_{n-1}$ are constant scalars. Following a procedure similar to that of controllable canonical form section, it can be shown that the transfer function of Eq. (3.6.3) is

$$g(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B = \frac{\beta_{n-1}s^{n-1} + \cdots + \beta_1 s + \beta_0}{s^n + p_{n-1}s^{n-1} + \cdots + p_1 s + p_0} \tag{3.6.4}$$

Diagonal Form

Consider the state equation

$$\begin{aligned} x &= \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \cdots & \lambda_{n-1} & 0 \\ 0 & 0 & & 0 & \lambda_n \end{pmatrix} x + \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} \end{pmatrix} u \\ y &= (\gamma_0 \gamma_1 \cdots \lambda_{n-1}) x \end{aligned} \tag{3.6.5}$$

where λ_i are district. Equation (3.6.5) is said to be in diagonal form since the system matrix A is diagonal. The characteristic polynomial of this system is

$$p(s) = \det(sI - A) = (s - \lambda_1)(s - \lambda_2)\cdots(s - \lambda_n)$$

Thus the diagonal elements are the system eigenvalues. The transfer function of Eq. (3.6.5) is

$$g(s) = C(sI - A)^{-1}B =$$

$$(\gamma_0 \gamma_1 \cdots \gamma_{n-1}) \begin{pmatrix} \frac{1}{s-\lambda_1} & & 0 & \\ & \frac{1}{s-\lambda_2} & & \\ 0 & & \ddots & \\ & & & \frac{1}{s-\lambda_n} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{pmatrix}$$

or

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$g(s)=\frac{\gamma_0\beta_0}{s-\lambda_1}+\frac{\gamma_1\beta_1}{s-\lambda_2}+\cdots+\frac{\gamma_{n-1}\beta_{n-1}}{s-\lambda_n} \tag{3.6.6}$$

Note that in order to transform the state equation into a standard form, certain conditions must hold. For example, when the system eigenvalues are distinct, it is always possible to transform the system matrix into the diagonal form.

Hessenberg form. The transformation of system equations into controllable canonical form can be a numerically unstable procedure, particularly for a high-order system. A special form of system equations, called Hessenberg form, can be obtained using numerically stable methods and is useful for many computer-aided design applications. The Hessenberg form for single-input single-output systems is given by

$$\dot{x}=\begin{bmatrix} x & x & x & & x & x & x \\ x & x & x & & x & x & x \\ 0 & x & x & & x & x & x \\ 0 & 0 & x & & x & x & x \\ . & . & . & & & x & x \\ . & . & . & \cdots & . & . & . \\ . & . & . & & . & . & . \\ 0 & 0 & 0 & & x & x & x \\ 0 & 0 & 0 & & 0 & x & x \end{bmatrix} x+\begin{bmatrix} x \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix} u$$

$$y=cx$$

where the column a_i, $i=1,2,\ldots,n$ of the system matrix has the property that the first $(i+1)$ elements of a_i are nonzero while the remaining elements are zero. A system can always be brought into the Hessenberg form by a sequence of orthogonal transformations (Jamshidi *et al.*, 1992).

3.6.2 CAD Examples

In this session a number of CAD examples will be used to help the reader use the computer as a tool.

CAD Example 3.6.1 An approximate linear model of the dynamics of an aircraft is

$$\begin{pmatrix} \dot{r}(t) \\ \dot{y}(t) \\ \dot{\beta}(t) \\ \dot{\gamma}(t) \end{pmatrix}=\begin{pmatrix} -10 & 0 & -10 & 0 \\ 0 & -0.7 & 9 & 0 \\ 0 & -1 & -0.7 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} r(t) \\ y(t) \\ \beta(t) \\ \gamma(t) \end{pmatrix}+\begin{pmatrix} 20 & 2.8 \\ 0 & -3.13 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \delta_a(t) \\ \delta_r(t) \end{pmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $r(t)$ and $y(t)$ are incremental roll and yaw rates, $\beta(t)$ and $\gamma(t)$ are incremental side slip and roll angles, and $\delta_a(t)$ and $\delta_r(t)$ are incremental changes in the aileron and rudder angles, respectively.

1. Is the aircraft controllable?
2. Is it controllable using only $\delta_a(t)$ as the control input?
3. Is it controllable through $\delta_r(t)$ only?

SOLUTION: A computer solution using LSSPAK follows.

1. <<CONOBS>> determines whether a multivariable linear time-invariant continuous-time system $dx/dt = Ax + Bu$, $y = Cx + Du$ or discrete-time system $x(k + 1) = Ax(k) + Bu(k)$, $y(k) = Cx(k) + Du(k)$ is completely controllable or completely observable.

Matrix dimensions are: A:$n \times n$, B: $n \times m$, C: $r \times n$, D: $r \times m$
Matrix D does not influence system controllability or observability

MATRIX DIMENSIONS: $n = 4$, $m = 2$, $r = 2$

Matrix A

- .100E + 02	0.000E + 00	-.100E + 02	0.000E + 00
0.000E + 00	-.700E + 00	0.900E + 01	0.000E + 00
0.000E + 00	-.100E + 01	-.700E + 00	0.000E + 00
0.100E + 01	0.000E + 00	0.000E + 00	0.000E + 00

Matrix B

0.200E + 02	0.280E + 01
0.000E + 00	-.313E + 01
0.000E + 00	0.000E + 00
0.000E + 00	0.000E + 00

CONTROLLABILITY Matrix Qc =
2E-01 2.8E+00 -2E+02 -2.8E+01 2E+03 2.487E+02 -2E+04 -2.443E+03
0.0E+00 -3.13E+00 0.0E+00 2.19E+00 0.0E+00 2.66E+01 0.0E+00 -5.8E+01

0.00E+00 0.00E-00 0.00E+00 3.13E+00 0.00E-00 -4.382E+00 0.00E+00 -2.357E+01
0.00E+00 0.00E+00 2.00E+01 2.80E+00 -2.00E+02 -2.80E+01 2.00E+03 2.487E+02

THE SYSTEM IS COMPLETELY CONTROLLABLE
The Controllability Matrix has Rank 4 = n

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

CONTROLLABILITY Matrix Qc =

2.000E + 01	- 2.000E + 02	2.000E + 03	- 2.000E + 04
0.000E + 00	0.000E + 00	0.000E + 00	0.000E + 00
0.000E + 00	0.000E + 00	0.000E + 00	0.000E + 00
0.000E + 00	2.000E + 01	- 2.000E + 02	2.000E + 03

THE SYSTEM IS NOT COMPLETELY CONTROLLABLE
The Controllability Matrix has Rank 2 < n

CONTROLLABILITY Matrix Qc =

2.800E + 00	-2.800E + 01	2.487E + 02	-2.443E + 03
-3.130E + 00	2.191E + 00	2.664E + 01	-5.808E + 01
0.000E + 00	3.130E + 00	-4.382E + 00	-2.357E + 01
0.000E + 00	2.800E + 00	-2.800E + 01	2.487E + 02

THE SYSTEM IS COMPLETELY CONTROLLABLE The Controllability Matrix has Rank 4 = n

CAD Example 3.6.2

Consider the linearized model of an aircraft given in CAD Example 3.6.1. In which of the following cases is the system observable?

1. Roll rate $r(t)$ is the only measurable quantity.
2. Yaw rate $y(t)$ and sideship angle $\beta(t)$ are measurable quantities.
3. Roll angle $y(t)$ is the only measurable quantity.

SOLUTION: An LSSPAK solution of this problem is shown below.

1. <<CONOBS>> determines whether a multivariable linear time-invariant continuous-time system $dx/dt = Ax + Bu$, $y = Cx + Du$ or discrete-time system $x(k + 1) = Ax(k) + Bu(k)$, $y(k) = Cx(k) + Du(k)$ is completely controllable or completely observable.

Matrix dimensions are: $A{:}n \times n$, B: $n \times m$, C: $r \times n$, D: $r \times m$
Matrix D does not influence system controllability or observability
MATRIX DIMENSIONS: $n = 4$, $m = 1$, $r = 1$

Matrix A

-.100E + 02	0.000E + 00	-.100E + 02	0.000E + 00
0.000E + 00	-.700E + 00	0.900E + 01	0.000E + 00
0.000E + 00	-.100E + 00	-.700E + 00	0.000E + 00
0.100E + 01	0.000E + 00	0.000E + 00	0.000E + 00

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Matrix B

0.200E + 02	0.280E + 01
0.000E + 00	-0.313E + 01
0.000E + 00	0.000E + 00
0.000E + 00	0.000E + 00

Matrix C

0.100E+01	0.000E+00	0.000E+00	0.000E+00

OBSERVABILITY Matrix Qo =

1.000E + 00	0.000E + 00	0.000E + 00	0.000E + 00
-1.000E + 01	0.000E + 00	- 1.000E + 01	0.000E + 00
1.000E + 02	1.000E + 01	1.070E + 02	0.000E + 00
-1.000E + 03	-1.140E + 02	- 9.849E + 02	0.000E + 00

THE SYSTEM IS NOT COMPLETELY OBSERVABLE The Observability Matrix has Rank 3 < n

OBSERVABILITY Matrix Qo =

0.000E + 00	1.000E + 00	0.000E + 00	0.000E + 00
0.000E + 00	0.000E + 00	1.000E + 00	0.000E + 00
0.000E + 00	-7.000E - 01	9.000E + 00	0.000E + 00
0.000E + 00	1.000E + 00	-7.000E - 01	0.000E + 00
0.000E + 00	-8.510E + 00	-1.260E + 01	0.000E + 00
0.000E + 00	1.400E + 00	-8.510E + 00	0.000E + 00
0.000E + 00	1.856E + 01	-6.777E + 01	0.000E + 00
0.000E + 00	7.530E + 00	1.856E + 01	0.000E + 00

CAD Example 3.6.3. Determine stability of the following systems.

a. $$\dot{x}(t) = \begin{pmatrix} 1 & 0.2 & 0 & -1 & 2 \\ 1.1 & 2 & -3 & 0 & 0 \\ 0 & -0.7 & 2 & 0.2 & 0.6 \\ 0.1 & 0 & 0 & 0.7 & 1.8 \\ 3.5 & 12 & 1 & 0 & 0 \end{pmatrix} x(t)$$

b. $$x(k+1) = \begin{pmatrix} 0 & 1 & 0 & 0.2 \\ 0.6 & 0 & 1 & 0 \\ -0.1 & 1 & 0 & 0 \\ 0 & 2 & 0.2 & 0.8 \end{pmatrix} x(k)$$

controlengineers.ir @controlengineers

SOLUTION: Below is a LSSTB solution of this problem.

```
>> A1 = [1.20-12;1.12-300; 0-.72.2.6; .100.7 1.8; 3.5 12 100;
>> EIG(A1)

        ANS    =
                3.7443 + 1.3541i
                3.7443  - 1.3541i
                1.2492  - 0.0000i
               -1.5189 + 0.8417i
               -1.5189  - 0.8417i

>> SYSTEM 1 is unstable
>> a2 = [0  1  0  .2;  .6  0  1  0;  -.1  1  0  0;  0  2  .2  .8];
>>eig(a2)

        ANS   =
               -1.2485
                1.3830
                0.0614
                0.6041

>>SYSTEM 2 is unstable
```

CAD Example 3.6.4. Determine asymptotic stability of the system

$$\dot{x} = \begin{pmatrix} 1 & 2 & 0.1 \\ 7 & -0.3 & 2 \\ 1 & 0 & -1.5 \end{pmatrix} x(t)$$

using the Lyapunov method. An LSSTB solution for this problem follows:

```
>>a = [12.1; 7-.32; 10-1.5];
>>q = eye(3);
>>p = lyapu(a,q)

WARNING
UNSTABLE EIGENVALUE - SOLUTION MAY NOT BE POSITIVE-DEFINITE.
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

P =

-0.3772	-0.0595	-0.0375
-0.0595	1.3705	0.1638
-0.0375	0.1638	0.3084

```
>> SYSTEM is unstable. Verify by eigenvalue of A.
>> eig(a)

   ANS =
          4.2484 - 0.0000i
         -3.1358 + 0.0000i
         -1.9126 - 0.0000i
```

3.7 Discussion and Conclusions

In this chapter three of the most fundamental issues of large-scale systems, i.e., stability, controllability, and observability, have been briefly discussed. In this section a comparison between the two main techniques of large-scale systems stability—Lyapunov and input-output—is made first, followed by a brief rundown on researchers' efforts since 1966. The second part of the discussion is devoted to the controllability and observability of large-scale systems. In the final segment of this section, attention is focused on controllability based on decentralized structure.

3.7.1 Discussion of the Stability of Large-Scale Systems

The discussion on the stability of large-scale systems begins by pointing out that Lyapunov and IO stability methods imply each other (Araki, 1978a). In order to facilitate this discussion, consider a large-scale interconnected system represented by

$$\dot{x}_i(t) = A_i x_i(t) + b_i\left\{-\psi_i(y_i(t),t) + \tilde{h}_i(y_i(t),\ldots,y_N(t),t) + z_i(t)\right\} \qquad (3.7.1)$$

$$y_i(t) = c_i^T x_i(t), \quad i = 1,\ldots,N \qquad (3.7.2)$$

where x_i is $n_i \times 1$ state vector of ith subsystem, A_i, b_i, and c_i are a constant matrix and vectors with the triplet (b_i, A_i, c_i^T) being both controllable and observable, and real-valued functions $\psi_i(\cdot, t)$ and $h_i(\cdot, \cdot, \ldots, t)$ are assumed to satisfy

$$\gamma_i \tau^2 \le \psi_i(\tau,t)\tau \le \delta_i \tau^2, \;\; t, \tau \in R \qquad (3.7.3)$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\left|\tilde{h}_i(\tau_1,\ldots,\tau_N,t)\right| \le \sum_{j=1}^{N} b_{ij}\left|\tau_j\right|, \quad t,\tau_j \in R \tag{3.7.4}$$

where γ_i, δ_i, and b_{ij} are nonnegative constants with $\delta_i \ge \gamma_i$. It is noted that this system is a special case of composite input-output system (3.3.4) with $D = I$, $F = 0$, $\mu = N$, and μ_i, $i = 1, \ldots, N$.

Araki (1978a) has stated a corollary, given below, which provides conditions for the stability of system (3.7.1)–(3.7.4) via the Nyquist criterion.

Corollary 3.2. Let a_i, $i = 1, \ldots, N$, be positive constants and $E = (e_{ij})$ defined by (3.2.13) be an M-matrix where $e_{ii} = a_i - b_{ii}$ and $e_{ij} = -b_{ij}$, $i \neq j$ and b_{ij} are given in *(3.7.4)*, when $\delta_i = \infty$ for $i = 1, \ldots, N$; then the system of (3.7.1)–(3.7.2) is L_2-stable if the Nyquist diagram of each transfer function $\bar{g}_i(s)$

1. does not intersect or encircle a disk with center at $c_i = \left\{\frac{1}{2}\left[-(\gamma_i - a_i)^{-1} + \varepsilon_i\right], j0\right\}$ and radius of $r_i = \frac{1}{2}\left((\gamma_i - a_i)^{-1} + \varepsilon_i\right)$ for some $\varepsilon > 0$ and if $\gamma_i < a_i$. Note that *j0* represents a zero imaginary quantity on the s-plane and is not be confused with the index j above;
2. lies inside the disk with center at c_i and radius $r_i = \frac{1}{2}\left((\gamma_i - a_i)^{-1} + \varepsilon_i\right)$ for some ε_i; satisfying $\left|(\gamma_i - a_i)^{-1}\right| > \varepsilon_i > 0$ and if $\gamma_i < a_i$;
3. and lies to the right of a line $s = \varepsilon_i$ parallel to the imaginary axis for some $\varepsilon_i > 0$ and if $\gamma_i = a_i$.

Typical transfer function $\bar{g}_i(s)$ is given by

$$\bar{g}_i(s) = c_i^T (sI - A_i)^{-1} b_i \tag{3.7.5}$$

Furthermore, let the disturbance inputs $u_i(t) = 0$ for $i = 1, \ldots, N$. The system would thus become a special case of (3.2.2) where

$$f_i(x_i,t) = A_i x_i - b_i \psi_i\left(c_i^T x_i\right) \tag{3.7.6}$$

$$g_i(x,t) = b_i \tilde{h}_i\left(c_1^T x_i,\ldots,c_N^T x_N,t\right) \tag{3.7.7}$$

For the sake of simplicity assume that condition 3 of Corollary 3.2 is the

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

only one holding for this unforced system. In other words, every $\bar{g}_i(s)$ satisfies condition 3. Then by utilizing Lefschetz's form of the Kalman-Yakubovich Lemma (Lefschetz, 1965), there are positive-definite matrices P_i and Q_i satisfying

$$A_i^T P_i + P_i A_i + Q_i = 0 \tag{3.7.8}$$

$$P_i b_i - c_i = 0 \tag{3.7.9}$$

If one would let $\upsilon_i(x_i,t) = x_i^T P_i x_i$, then through (3.7.3), (3.7.6), (3.7.8), and (3.7.9) the time derivative of υ_i (Araki, 1978) would be

$$\begin{aligned}\dot{\upsilon}_i(x_i,t)\Big|_{(Iss)} &= x_i^T\left(A_i^T P_i + P_i A_i\right)x_i - 2x_i^T P_i b_i \psi_i\left(c_i^T x_i, t\right)\\ &\le -x_i^T Q_i x_i - x_i^T\left(-2P_i b_i + 2c_i\right)\psi_i\left(c_i^T x_i, t\right) - 2\gamma_i\left(c_i^T x_i\right)^2\end{aligned} \tag{3.7.10}$$

$$= -\, x_i^T Q_i x_i - 2\gamma_i\left(c_i^T x_i\right)^2 \tag{3.7.11}$$

Moreover, from (3.7.4), (3.7.7), and (3.7.9), one would get

$$\begin{aligned}\left(\partial\upsilon_i(x_i,t)/\partial x_i\right)^T g_i(x,t) &\le 2x_i^T P_i b_i \tilde{h}_i\left(c_1^T x_i, \ldots, c_N^T x_N, t\right)\\ &\le \sqrt{2}\left|c_i^T x_i\right| \sum_{j=1}^{N} b_{ij}\sqrt{2}\left|c_j^T x_j\right|\end{aligned} \tag{3.7.12}$$

Now by choosing $\omega_i(x_i) = \sqrt{2}\left|c_i^T x_i\right|$, $a_i = \gamma_i$, and $z_i(x_i) = x_i^T Q_i x_i$, it becomes clear that relations (3.7.11) and (3.7.12) are identical to conditions (3.2.22) and (3.2.23) of Theorem 3.2, respectively. Therefore it has been shown that condition 3 of Corollary 3.2 (IO stability) implies conditions 1 and 2 of Theorem 3.2 (Lyapunov stability).

The converse of the above can also be easily shown to hold. Let A_i be a Hurwitz matrix and that conditions of Theorem 3.2 hold where

$$\upsilon_i\left(x_j,t\right) = x_i^T P_i x_i;\;\; \omega_i(x_i) = \alpha_i\left|c_i^T x_i\right|,\;\; \alpha_i > 0 \tag{3.7.13}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where P_i is the positive-definite solution of (3.7.8). The condition (3.7.3) implies (3.7.10). Now by assuming that

$$a_i = \gamma_i, \quad \alpha_i = \sqrt{2}, \quad z_i(x_i) = x_i^T Q_i x_i \tag{3.7.14}$$

the right-hand side of (3.7.10) can be bounded similar to (3.2.22) of Theorem 3.2 and that (3.7.9) is required. Then again by the Kalman-Yakubovich Lemma (Lefschetz, 1965), condition 3 of Corollary 3.2 is satisfied. Thus, in a limited case, it has been shown that conditions of Corollary 3.2 and Theorem 3.2 imply each other. Table 3.1 provides a comparison between IO and Lyapunov stability.

The notion of connective stability, introduced by Siljak (1972b, 1978), takes into account the structural perturbations of a large-scale composite system on the overall stability. The scheme is applicable to both linear and nonlinear systems, and a simple algebraic test would normally be sufficient for the required criterion. There are two conceptually important conclusions on connective stability. One is that this stability notion establishes an ability for the system to withstand sudden perturbations, hence a dynamic reliability property. The second is that the stability of the system (3.2.27) would still hold even for nonlinear interconnection functions $g_{ij}(t, x)$ whose actual shapes are not necessary as long as they are continuous and bounded between zero and one. The latter point would indicate that connective stability implies that the system would remain stable in spite of perturbations such as inaccurate measurements, parameter settings, computations, nonlinear characteristics, etc.

In spite of the treatments in this chapter, the fact remains that it is very improbable that a practical large-scale system is asymptotically stable in the large. Therefore, one would have to rely on local analysis using Lyapunov functions in most practical considerations. Even so, the construction of Lyapunov functions for subsystems with the "best" set of parameters is a rather laborious task. The following steps, suggested by Araki (1978a), offer a workable procedure: (1) Consider a nominal (equilibrium) point and extrapolate a system equation around it; (2) continue with an IO stability which is normally somewhat simpler than constructing Lyapunov functions; and (3) interpret the results of IO stability to form Lyapunov functions similar to the development following Equations (3.7.1)–(3.7.13) for a particular system. It should, however, be mentioned that this development is not complete, and further work on the derivations of Lyapunov

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 3.1 A Comparison of IO and Lyapunov Stability Methods

Basic Requirements	Inputs-Output Stability	Lyapunov Stability
main approach	small gain theorem (and circle criteria in L2 case)	second method of Lyapunov
subsystems knowledge	gains (and sector conditions in L2 case)	Lyapunov functions (stable) or positive-definite decrescent, radially unbounded functions (unstable)
interconnections knowledge	(functional) bounds	(instantaneous) bounds
derivation of composite system stability conditions	determination of weights c_i which guarantee small gain condition with respect to the norm	determination of weights c_i which guarantees negative-definiteness of
	$\left\{\sum_{i=1}^{N}\left(c_i\lVert u_i\rVert^2\right)\right\}^{1/2}$ or	$\sum_{i=1}^{N} c_i v_i(x_i,t)$
	circle condition with respect to the inner product $\sum_{i=1}^{N}\left\{c_i^2\langle u_i, y_i\rangle\right\}$	
test	Metzler matrix test	

functions from frequency domain conditions are necessary.

Finally a few comments regarding a historical viewpoint on the major contributions of large-scale composite systems stability is due. In order to facilitate this, Table 3.2 summarizes the development on the application of Lyapunov method to composite systems. As shown, the method was first considered by Bailey (1966). The basic assumptions made here were quadratic order Lyapunov functions for subsystems, linear interconnections,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 3.2 A Summary of Eight Approaches in Lyapunov Stability of Large-scale Composite Systems

Contributors	Subsystem Lyapunov Functions	Assumptions on Interactions	Stability Criterion Technique	Required Tests	Test Involving Arbitary Parameters
Bailey (1966)	quadratic	linear	vector Lyapunov	stability of linear time-invariant comparison system	no
Thompson (1970)	quadratic	linear	scalar Lyapunov	positive-definiteness of a matrix or other condition	yes
Michel and Porter (1972)	quadratic	Linear bounds	scalar Lyapunov	positive-definiteness of a matrix	yes
Araki and Kondo(1972) Araki (1978b)	quadratic	linear bounds	scalar Lyapunov	Metzler matrix	no
Šiljak (1972b,1978) (connective stability)	linear	linear bounds	vector Lyapunov	Metzler matrix	no
Grujić and Šiljak(1973b)	linear	linear bounds	scalar Lyapunov	Metzler matrix	no
Suda (1973)	quadratic	mean-square bounds	scalar Lyapunov	Metzler matrix	no
Moylan and Hill (1978) (dissipative Systems)	quadratic	linear	scalar Lyapunov	quasi-dominancy of a matrix	no

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and vector Lyapunov functions. Following the initial attempt by Bailey (1966), several authors have extended and improved it by altering the above three points as shown in Table 3.2.

Another line of research on the stability of large-scale systems has been that of the so-called dissipative systems. A system with m inputs and r outputs is said to be (C, Q, P)-dissipative if the truncated inner product relation

$$\langle u, Pu \rangle_\tau + 2\langle y, Qu \rangle_\tau + \langle y, Cy \rangle_\tau \geq 0 \tag{3.7.15}$$

for all $\tau \varepsilon R$ and C, Q, and P are $r \times r$, $r \times m$, and $m \times m$ matrices, with C and Q being symmetric as well. Moylan and Hill (1978) have assumed linear interconnections for dissipative systems and have presented a criterion based on the quasi-dominancy condition of a single matrix within the context of both Lyapunov and IO stability. The main characteristics of this is summarized in Table 3.2.

3.7.2 Discussion of the Controllability and Observability of Large-Scale Systems

The controllability and observability of composite systems were considered in Section 3.4. The notion of connectability was introduced in Section 3.4.1. The graph-theoretic approach utilized here is rather simple However, the generation of large graphs for realistic applications with the aid of digital computers must be utilized. It has been shown that a composite system with arbitrary interconnection and any number of inputs and outputs is controllable and observable for almost all interconnection gains between subsystems, provided that the system is connectable. The application of this result to structural controllability was given in Section 3.5.3.

The structural controllability and observability of large-scale systems were discussed in Section 3.5. The original proposal of the structural controllability was due to Lin (1974) and used a graph-theoretic scheme for SISO systems. The results of Lin (1974) were extended to MIMO systems through an algebraic approach which is more tractable and simpler to use. The generic rank determination through Algorithm 3.3 is a useful computational tool in evaluation the rank of a matrix. This algorithm cannot only be used in checking structural controllability, as demonstrated in Section 3.5.2 but also in composite systems considered earlier: Theorem 3.9 due to Shields and Pearson (1976) with the aid of Algorithm 3.3 provides a simple scheme for checking the structural controllability of large scale systems. The rank determination of usual controllability and observability

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

matrices along with input-output connectability (Definition 3.17) are used by Davison (1976) to set up ground rules for structural controllability and observability in Section 3.5.3. The numerical Example 3.5.5 demonstrated that these criteria (Theorem 3.10 and Corollary 3.1) are fairly simple to use. Either Algorithm 3.3 or the generic rank determination scheme by Davison (1977) can be used to satisfy one of the two conditions in the proposed criteria.

Problems

3.1. Consider a third-order system decomposed into second- and first-order subsystems:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{cc:c} -1 & 0.1 & 0.1 \\ 0.2 & -1 & 0.5 \\ \hdashline 1 & 0.8 & -2 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix}$$

Check whether this system is asymptotically stable in the sense of Lyapunov using Theorem 3.1.

3.2. Repeat Example 3.2.2. for the following nonlinear system:

$$\dot{x}_1 = -\tfrac{4}{5} x_1^5 + g_1(x,t)$$

$$\dot{x}_2 = -\tfrac{1}{5} x_2^5 + g_2(x,t)$$

where $g_i(x, t)$, $i = 1, 2$, should de defined in the process.

3.3. For the nonlinear interconnected system

$$\dot{x}_1 = 0.2x_1 - x_2 \cos x_1 + 0.1x_1 \sin x_1 x_2$$

$$\dot{x}_2 = x_2 - x_2 \cos x_1 + 0.25x_2 \cos x_1 x_2$$

$$\dot{x}_3 = x_3 - x_2 \sin x_1 + 0.125x_1 \cos x_1 x_2$$

Check its stability in the sense of Lyapunov.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

3.4. Consider the system of Figure 3.3 and let the forward path's diagonal transfer functions $G_i(x)$, $i = 1, 2$, be

$$G_1(s) = 0.5/((1+2s)(1+3s)), \quad G_2(s) = 1/((1+s)(1+6s))$$

and the output functions be

$$\tilde{y}_1(t) = y_1(t) + 0.5\psi(y_2(t))$$
$$\tilde{y}_2(t) = 0.4y_2(t) + 0.8\psi(y_1(t))$$

Check for the IO stability of this system.

3.5. Repeat Example 3.3.2 for

$$G_1(s) = 1/((s+5)(s+5)), \quad G_2(s) = 5/((s+2)(s+4))$$
$$H_1 = 1, \quad H_2 = K_1 = 2, \quad K_2 = 0.5$$

3.6. Consider a system

$$\dot{x}_1 = -x_1 + 0.08x_2$$
$$\dot{x}_2 = 0.5x_1 - x_2$$

Check the stability of this system i.s.L.

3.7. CADLAB Repeat Problem 3.6 on LSSPAK or LSSTB, or your favorite CACSD package.

3.8. Using the development in the proof of Theorem 3.1, prove Theorem 3.2.

3.9. Consider a sixth-order system

$$\begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{ccc|ccc} -1.5 & 0.2 & 0.1 & 0.2 & 0.1 & 0.1 \\ 0.1 & -3 & 0.3 & 0.1 & 0.1 & 0.2 \\ 0.2 & -1 & -2 & 0.2 & 0.3 & 0.4 \\ \hline 1 & 0 & 1 & -4 & 1 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0.2 \\ 0.5 & 0 & 1 & -0.5 & 0.2 & -5 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix}$$

Is the system asymptotically connectively stable?

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

3.10. CADLAB Consider a fifth-order parallel-series composite system

$$\begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{ccc|cc} -1 & 0.1 & 0.5 & 0 & 0 \\ 0 & -0.5 & 0.1 & 0 & 0 \\ 0.1 & 0 & -0.1 & 0 & 0 \\ \hline 0.1 & 0 & 0.1 & 0.1 & -0.2 \\ 0 & 0.2 & 0 & -0.1 & -0.25 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix} + \left[\begin{array}{c} 1 \\ 0.2 \\ 0.4 \\ \hline 0 \\ 1 \end{array}\right] u$$

$$y = \left[\begin{array}{ccc|cc} 1 & 0.5 & 1 & 1 & 1 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix} + [1]u$$

Determine whether this system is controllable and/or observable. Use a computer package for this purpose.

3.11. CADLAB For the system of Problem 3.10, determine the controllability and observability of its closed-loop form.

3.12. A second-order composite system is given by

$$\dot{x} = \begin{bmatrix} -2 & b \\ a & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 1 \end{bmatrix} x$$

with unknown interconnections a and b. Determine a region in the $(a - b)$ plane where the system is both controllable and observable.

3.13. A closed-loop composite system is given by

$$\begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{c|c} -d & c \\ \hline -1 & d+1 \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix} + \begin{bmatrix} c \\ d \end{bmatrix} u$$

$$y = (-1 - d) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + du$$

Determine whether the system is controllable or observable for $d \geq 2$ and $c \leq 3$.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

3.14. Find the generic rank of the following matrix by Algorithm 3.3.

$$A = \begin{bmatrix} 0 & x & 0 & x & x & x & 0 \\ x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 & x \\ x & x & x & 0 & 0 & 0 & x \\ x & x & 0 & 0 & x & x & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

where an x represents a nonzero entry.

3.15. Write a MATLAB file "generic.m" to implement the generic rank Algorithm 3.3.

3.16. Check whether a system described by the matrices

$$(A,B) = \left(\begin{bmatrix} x & x & 0 \\ 0 & x & x \\ x & 0 & x \end{bmatrix}, \begin{bmatrix} 0 \\ x \end{bmatrix} \right)$$

is structurally controllable.

3.17. Determine the structural controllability and observability of the system

$$(C,A,B) = \left(\begin{bmatrix} x & 0 & x & 0 & x \\ x & x & 0 & 0 & x \end{bmatrix}, \begin{bmatrix} x & 0 & 0 & x & x \\ 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x \\ x & x & 0 & 0 & x \\ x & x & x & x & x \end{bmatrix}, \begin{bmatrix} 0 & x \\ x & x \\ 0 & 0 \\ x & x \\ x & 0 \end{bmatrix} \right)$$

by using Theorem 3.10 and Corollary 3.1.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

3.18. Determine the observability of the system

$$\dot{x} = \begin{pmatrix} 1 & 0.2 & -0.1 & 1.5 \\ 0 & -1.2 & -0.5 & -0.7 \\ 0.3 & 0.4 & 0.7 & 2.1 \\ 2.1 & 0 & 3.1 & 0.1 \end{pmatrix} x(t) + \begin{pmatrix} 0.1 & -1 \\ 0.3 & 0.4 \\ -0.7 & 3.2 \\ 1 & 0 \end{pmatrix} u(t)$$

$$y(t) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0.1 & 0 & 0.6 & 0.8 \end{pmatrix} x(t)$$

3.19. Determine controllability and observability of the system

$$\dot{x}(t) = \begin{pmatrix} 0 & 0 & -6 \\ 1 & 0 & -11 \\ 0 & 1 & -6 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u(t)$$

$$y(t) = (1 \quad 0 \quad 0)\, x(t)$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 4

Hierarchical Control of Large-Scale Systems

4.1 Introduction

The notion of a large-scale system, as it was briefly discussed in Chapter 1, may be described as a complex system composed of a number of constituents or smaller subsystems serving particular functions and shared resources and governed by interrelated goals and constraints (Mahmoud, 1977; Jamshidi, 1983). Although interaction among subsystems can take on many forms, a common one is hierarchical, which appears somewhat natural in economic, management, organizational, and complex industrial systems such as steel, oil, robotics, and paper. Within this hierarchical structure, the subsystems are positioned on levels with different degrees of hierarchy. A subsystem at a given level controls or "coordinates" the units on the level below it and is, in turn, controlled or coordinated by the unit on the level immediately above it. Figure 4.1 shows a typical hierarchical (multilevel) system. The highest level coordinator, sometimes called the supremal coordinator, can be thought of as the board of directors of a corporation, while another level's coordinators may be the president, vice-presidents, directors, etc. The lower levels can be occupied by plant managers, shop managers, etc., while the large-scale system is the corporation itself. In spite of this seemingly natural representation of a hierarchical structure, its exact behavior has not been well understood mainly due to the fact that little quantitative work has been done on these large-scale systems (March and Simon, 1958). Mesarovic *et al.* (1970) presented one of the earliest formal quantitative treatments of hierarchical (multilevel) systems. Since then, a great deal of work has been done in the field (Schoeffler and Lasdon, 1966; Benveniste *et al.*, 1976; Smith and

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

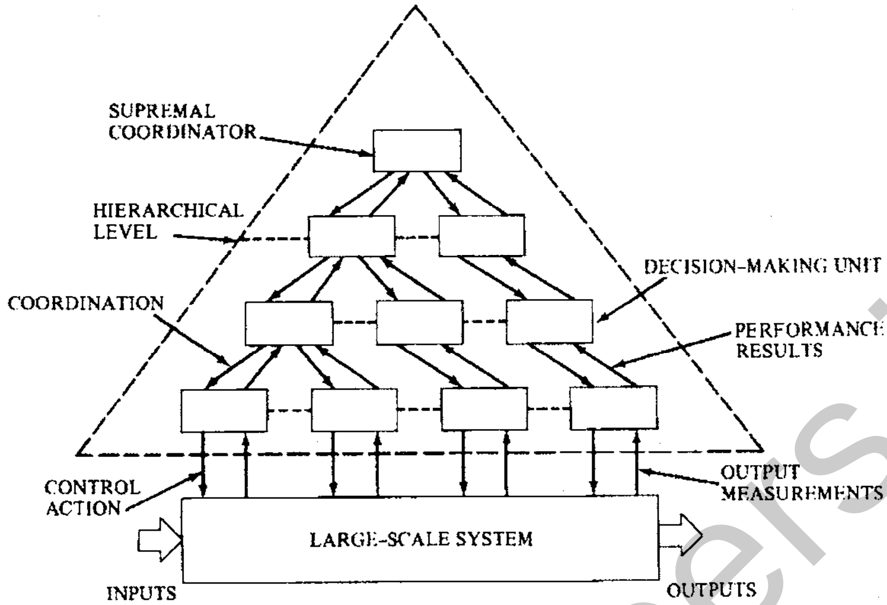


Figure 4.1 A hierarchical (multilevel) control strategy for a large-scale system.

Sage, 1973; Geoffrion. 1970; Schoeffler, 1971: Pearson. 1971: Cohen and Jolland, 1976; Sandell *et al.*, 1978; Singh, 1980; Jamshidi, 1983; Huang and Shao, 1994a,b). For a relatively exhaustive survey on the multilevel systems control and applications, the interested reader may see the work of Mahmoud (1977).

In this section, a further interpretation and insight of the notion of hierarchy, the properties and types of hierarchical processes, and some reasons for their existence are given. An overall evaluation of hierarchical methods is presented in Section 4.6.

There is no uniquely or universally accepted set of properties associated with the hierarchical systems. However, the following are some of the key properties:

1. A hierarchical system consists of decision-making components structured in a pyramid shape (Figure 4.1).
2. The system has an overall goal which may (or may not) be in harmony with all its individual components.
3. The various levels of hierarchy in the system exchange information (usually vertically) among themselves iteratively.
4. As the level of hierarchy goes up, the time horizon increases; i.e., the lower-level components are faster than the higher-level ones.

There are three basic structures in hierarchical (multilevel) systems de-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

pending on the model parameters, decision variables, behavioral and environmental aspects, uncertainties, and the existence of many conflicting goals or objectives.

1. *Multistrata Hierarchical Structure*: In this multilevel system in which levels are called strata, lower-level subsystems are assigned more specialized descriptions and details of the large-scale complex system than the higher levels.
2. *Multilayer Hierarchical Structure*: This structure is a direct outcome of the complexities involved in a decision-making process. The control tasks are distributed in a vertical division (Singh and Titli, 1978), as shown in Figure 4.2. For the multilayer structure shown here, regulation (first layer) acts as a direct control action, followed by optimization (calculation of the regulators' set points), adaptation (direct adaptation of the control law and model), and self-organization (model selection and control as a function of environmental parameters).
3. *Multiechelon Hierarchical Structure*: This is the most general structure of the three and consists of a number of subsystems situated in levels such that each one, as discussed earlier, can coordinate

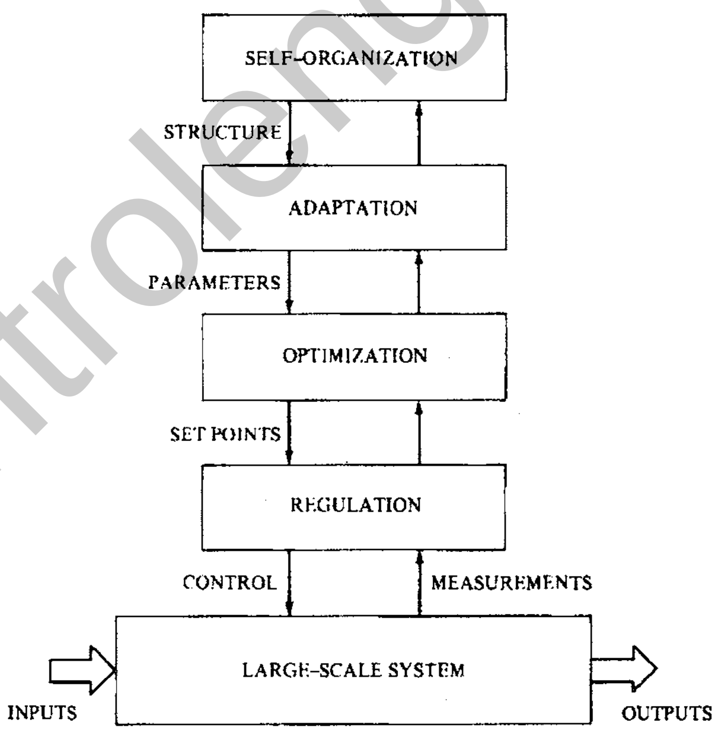


Figure 4.2 A multilayer control strategy for a large-scale system.

controlengineers.ir

controlengineers.ir @controlengineers

lower-level units and be coordinated by a higher-level one. This structure, shown in Figure 4.1, considers conflicting goals and objectives between decision subproblems. The higher-level echelons, in other words, resolve the conflict goals while relaxing interactions among lower echelons. The distribution of the control task in contrast to the multilayer structure described in Figure 4.2 is horizontal.

In addition to the vertical (multilayer) and horizontal (multiechelon) division of the control task, a third division called a time or functional division is possible (Singh and Titli, 1978). In this division a given subsystem's functional optimization problem is decomposed into a finite number of single-parameter optimization problems at a lower level and results in a considerable reduction in computational effort. This scheme has been used for hierarchical control of discrete-time systems by Jamshidi (1983).

Before the scope of the present chapter is given, based on the above discussion, one can make a tentative conclusion that a successful operation of hierarchical systems is best described by two processes known as *decomposition* and *coordination*. A pictorial representaiton of these two notions is shown in Figure 4.3 In summary, the basic notion behind *hierarchical control* is to: (a) *decompose* a given large-scale system into a number of small-scale subsystems, and (b) coordinate these subsystems' solutions until feasibility and optimality of the overall system are achieved through a multilevel iterative algorithm.

The coordination process as applicable to most hierarchical systems is described in Section 4.2. Section 4.3 is concerned with the open-loop control of continuous-time hierarchical systems where coordination between two levels is considered. The closed-loop hierarchical control of large-scale systems is discussed in Section 4.4, including the notions of "interaction prediction" and "structural perturbation." In Section 4.5, a hierarchical control approach based on Taylor Series (TS) and Chebyshev Series (SCSs) is presented. Here, the control problem is solved by a set of linear algebraic equations (Huang and Shao, 1994a, 1994b). A number of numerical examples illustrate the various techniques presented. The near-optimum design of linear and nonlinear hierarchical systems is discussed in Chapter 6. Section 4.6 is devoted to further discussion and the evaluation of hierarchical control techniques.

4.2 Coordination of Hierarchical Structures

It was mentioned in the previous section that a large-scale system can be

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

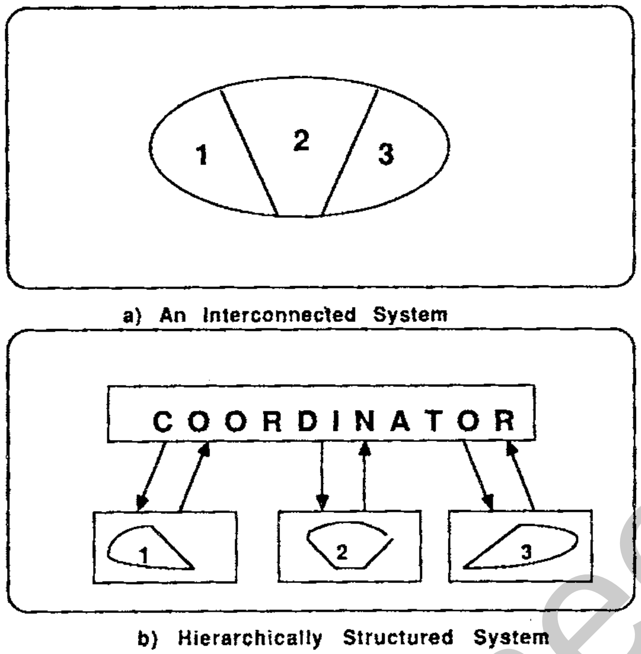


Figure 4.3 A pictorial representation of the notions of "decomposition" and "coordination" in a hierarchical control system.

controlengineers.ir

hierarchically controlled by decomposing it into a number of subsystems and then coordinating the resulting subproblems to transform a given integrated system into a multilevel one. This transformation can be achieved by a host of different ways. However, most of these schemes are essentially a combination of two distinct approaches: the *model-coordination method* (or "feasible" method) and *goal-coordination method* (or "dual-feasible" method) (Mesarovic *et al.*, 1969). In the next two sections, these methods are described for a two-subsystem static optimization (nonlinear programming) problem.

4.2.1 Model Coordination Method

Consider the following static optimization problem (Schoeffler, 1971):

minimize $J(x, u, y)$ (4.2.1)

subject to $f(x, u, y) = 0$ nonlinear constraint (4.2.2)

where x is a vector of system (state) variables, u is a vector of manipulated (control) variables, and y is a vector of interaction variables between subsystems. Let the problem and its objective function be decomposed into two subsystems, i.e.,

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$J(x,u,y)=J_1\left(x^1,u^1,y^1\right)+J_2\left(x^2,u^2,y^2\right) \tag{4.2.3}$$

and

$$f^i\left(x^i,u^i,y^1,y^2\right)=0,\quad i=1,2 \tag{4.2.4}$$

where x^i, u^i, and y^i are manipulated vectors of the system and interaction variables for ith subsystem, respectively. This decomposition has produced a performance function for each subsystem. However, through the vectors y^i, i = 1, 2, the subsystems are still interconnected. The objective of the model coordination method is to convert the integrated problem (4.2.1)–(4.2.2) into a two-level problem by fixing the interaction variables y^1 and y^2 at some value, say w^i, i = 1, 2, i.e.,

$$\text{Constrain } y^i = w^i,\ i = 1, 2 \tag{4.2.5}$$

Under this situation the problem (4.2.1)–(4.2.2) may be divided into the following two sequential problems:

First-Level Problem-Subsystem i

$$\text{Find } K_i(w) \min_{x^i,u^i} J_i\left(x^i, u^i, w^i\right) \tag{4.2.6}$$

$$\text{subject to } f^i(x^i, u^i, w^1, w^2) = 0 \tag{4.2.7}$$

Second-Level Problem

$$\underset{w}{\text{minimum}}\ K(w) = K_1(w) + K_2(w) \tag{4.2.8}$$

The above minimizations are to be done, respectively, over the following feasible sets:

$$S_1^i = \{(x^i, u^i): f^i(x^i, u^i, w) = 0\},\ i = 1, 2 \tag{4.2.9}$$

$$S_2^i = \{\ w^i: K_i(w^i) \text{ exists}\},\ i = 1, 2 \tag{4.2.10}$$

Figure 4.4 shows a two-level structure for the model coordination method. In this coordination procedure the variables w^i which fix interaction variables y^i are termed *coordinating variables.* Moreover, since certain internal interactions are fixed by adding a constraint to the mathematical model, this procedure is called model coordination. In other words, due to the fact

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

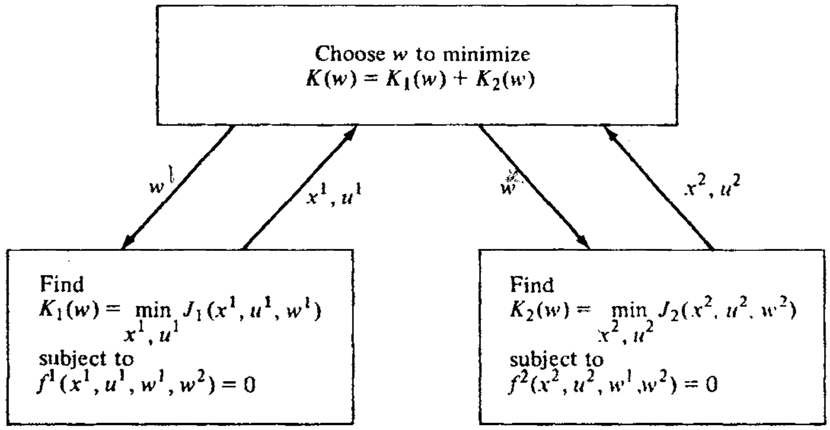


Figure 4.4 A two-level solution of a static optimization problem using model coordination.

that all intermediate variables x, u, and y are present, it is alternatively termed the "feasible decomposition method." Therefore, a system can operate with these intermediate values with a near-optimal performance. The first-level problems are constructed by fixing certain interacting variables in the original optimization problem, while assigning the task of determining these coordinating variables to the second level.

4.2.2 Goal Coordination Method

Consider the static optimization problem (4.2.1)–(4.2.2). In the goal coordination method the interactions are literally removed by cutting all the links among the subsystems. Let y^i be the outgoing variable from the ith subsystem, while its incoming variable is denoted by z^i. Due to the removal of all links between subsystems, it is clear that $y^i \neq z^i$. Under this condition, z^i acts as an arbitrary manipulated variable and should be chosen by the optimizing subsystems like x, u, and y. Moreover, the optimization problem considered in the previous section is completely decoupled into two subsystems due to the fact that their interactions are cut and their objective functions were already separated. In order to make sure the individual subproblems yield a solution to the original problem, it is necessary that the *interaction-balance principle* be satisfied, i.e., the independently selected y^i and z^i actually become equal (Mesarovic *et al.*, 1969; Schoeffler, 1971).

Here again, the procedure is to decompose the problem into a number of decoupled subproblems which constitute the first-level problem. The second-level problem is to force the first-level subproblems to a solution for which the interaction-balance principle holds. Mathematically, this multi-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

level formulation can be set up by introducing a weighting parameter α which penalizes the performance of the system when the interactions do not balance. Hence, to the objective function (4.2.3) a penalty term is added:

cost function

interaction variable

$$J(x, u, y, z, \alpha) = J_1(x^1, u^1, y^1) + J_2(x^2, u^2, y^2) + \alpha^T (y - z) \tag{4.2.11}$$

when $y = z$ you are done

where α is a vector of weighting parameters (positive or negative) which causes any interaction unbalance $(y - z)$ to affect the objective function. By introducing the z variables, the system's equations are given by

$$f_1(x^1, u^1, y^1, z^2) = 0 \tag{4.2.12}$$

$$f_2(x^2, u^2, y^2, z^1) = 0 \tag{4.2.13}$$

The set of allowable system variables is defined by

$$S_0 = \{(x, u, y, z): \ f_1(\cdot) = f_2(\cdot) = 0\} \tag{4.2.14}$$

Once the objective function (4.2.11) is minimized over the set S_0, it results in a function,

$$K(\alpha) = \min_{x,u,y,z \in S_0} J(x, u, y, z, \alpha) \tag{4.2.15}$$

After expanding the penalty term $\alpha^T(y-z) = \alpha_1^T(y^1 - z^1) + \alpha_2^T(y^2 - z^2)$ and considering the relations (4.2.11)–(4.2.13), the first-level problem is formulated as

Subsystem 1: $$\min_{x^1,u^1,y^1,z^2} J_1(x^1,u^1,y^1,z^2) + \alpha_1^T y^1 - \alpha_2^T z^2 \tag{4.2.16}$$

$$\text{subject to} \quad f_1(x^1,u^1,y^1,z^2) = 0 \tag{4.2.17}$$

Subsystem 2: $$\min_{x^2,u^2,y^2,z^1} J_2(x^2,u^2,y^2,z^1) + \alpha_1^T z^1 - \alpha_2^T y^2 \tag{4.2.18}$$

$$\text{subject to} \quad f_2(x^2,u^2,y^2,z^1) = 0 \tag{4.2.19}$$

The second-level problem is to manipulate the coordinating variable α in order to derive the two-subsystems interaction error to zero, i.e.,

$$\min_{\alpha} e = \min_{\alpha}(y - z) \tag{4.2.20}$$

decision variable

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It is clear from the second-level problem (Equation (4.2.20) that the coordinating variable (x is manipulated until the error e approaches zero; i.e., the interaction balance is held by manipulating the objective functions of the first-level problems (4.2.16) and (4.2.18) through variable α; hence, the name "goal coordination." Figure 4.5 shows the two-level solution via goal coordination. The reader should compare the two structures in Figures 4.4 and 4.5.

It will be seen later that the coordinating variable α can be interpreted as a vector of Lagrange multipliers and the second-level problem can be solved through well-known iterative search methods, such as the gradient, Newton's, or conjugate gradient methods.

4.3 Hierarchical Control of Linear Systems

In this section the goal coordination formulation of multilevel systems is applied to large-scale linear continuous-time systems within the context of open-loop control. In addition to the interaction-balance approach, another scheme known as the interaction prediction method is also discussed.

Let a large-scale dynamic interconnected system be represented by the following state equation:

$$\dot{x} = F(x, u, t), \quad x(t_o) = x_o \tag{4.3.1}$$

where x and u are $n \times l$ and $m \times l$ state and control vectors, respectively. It is assumed that the system can be decomposed into N interconnected subsystems s_i, $i = 1, 2, \ldots, N$, and the ith subsystem's state equation is

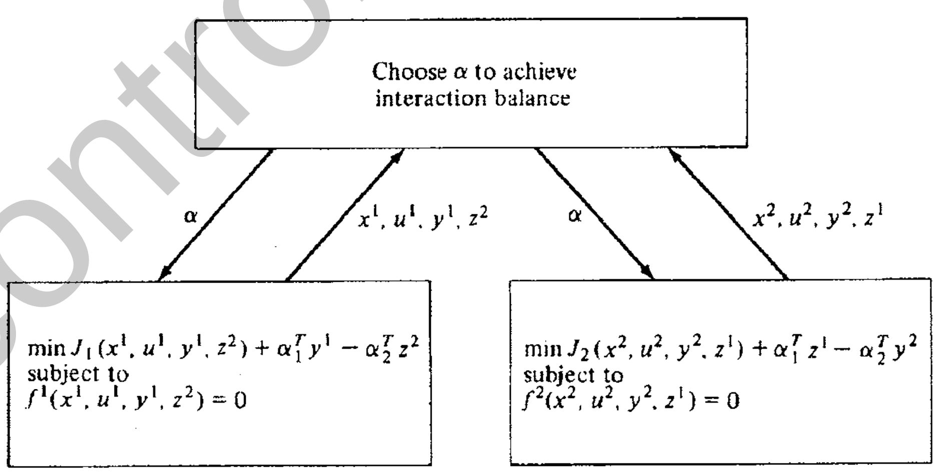


Figure 4.5 A two-level solution of a static optimization problem using goal coordination.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

given by

$$\dot{x}_i = f_i(x_i, u_i, t) + g_i(x, t), \quad x_i(t_o) = x_{io} \tag{4.3.2}$$

where x, u, x_i, u_i are respectively n-, m-, n_i-, and m_i-dimensional, g_i represents the ith subsystem interaction and

$$x^T(t) = \left(x_1^T(t), x_2^T(t), \ldots, x_N^T(t)\right) \tag{4.3.3}$$

$$u^T(t) = \left(u_1^T(t), u_2^T(t), \ldots, u_N^T(t)\right) \tag{4.3.4}$$

The objective, in an optimal control sense, is to find control vectors u_1, $u_2, \ldots, u_N$ such that a cost function

$$J = G\left(x\left(t_f\right)\right) + \int_{t_0}^{t_f} h(x(t), u(t), t)\, dt \tag{4.3.5}$$

is minimized subject to (4.3.1) and a feasible domain

$$u(t) \in U(x(t), t) = \left\{u \middle| v(x(t), u, t) \leq 0\right\} \tag{4.3.6}$$

Through the assumed decomposition of system (4.3.1) into N interconnected subsystems (4.3.2), a similar decomposition can be assumed to hold for the cost function constraint (4.3.6) and the interaction $g_i(x, t)$ in (4.3.2), i.e.,

$$J = \sum_i J_i = \sum_i \left\{G_i\left(x\left(t_f\right)\right) + \int_{t_0}^{t_f} h_i\left(x_i(t), z_i(t), u_i(t), t\right) dt\right\} \tag{4.3.7}$$

$$\upsilon(x, u, t) = \sum_j \upsilon_j\left(x_j, u_j, t\right) \tag{4.3.8}$$

$$g_i(x, t) = \sum_j g_{ij}\left(x_j, t\right) \tag{4.3.9}$$

where $z_i(t)$ is a vector consisting of a linear (or nonlinear) combination of the states of the N subsystems. Under the above assumption of separation, the large-scale system's optimal control problem (4.3.1), (4.3.5), and (4.3.6) can be rewritten as

minimize

$$\sum_i J_i = \sum_i \left\{G_i\left(x_i\left(t_f\right)\right) + \int_{t_0}^{t_f} h_i\left(x_i(t), u_i, (t), t\right) dt\right\} \tag{4.3.10}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

subject to

$$\dot{x}_i(t) = f_i\big(x_i(t), u_i(t), t\big) + \zeta_i\big(z_i(t), t\big), \quad x_i(t_o) = x_{io}, \quad i = 1, 2, \ldots, N \tag{4.3.11}$$

$$\zeta\big(z_i(t), t\big) = \sum_j g_{ij}\big(x_j(t), t\big), \quad i = 1, 2, \ldots, N \tag{4.3.12}$$

$$\sum_j \upsilon_j\big(x_j, u_j(t), t\big) \le 0 \tag{4.3.13}$$

The above problem, known as a hierarchical (multilevel) control, was demonstrated for a two-level optimization of a static problem in the previous section. The application of two-level goal-coordination to large-scale linear systems is given next.

4.3.1 Linear System Two-Level Coordination

Consider a large-scale linear time-invariant system:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \tag{4.3.14}$$

It is assumed that (4.3.14) can be decomposed into

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + C_i z_i(t), \ \ x_i(0) = x_{io} \tag{4.3.15}$$

and the $k_i \times 1$ interaction vector

$$z_i(t) = \sum_{\substack{j=1 \\ j \ne i}}^{N} G_{ij} x_j \tag{4.3.16}$$

is a linear combination of the states of the other N - 1 subsystems, and G_{ij} is an $n_i \times \ n_j$ matrix. The original system's optimal control problem is reduced to the optimization of N subsystems which collectively satisfy (4.3.15)–(4.3.16) while minimizing

$$J = \sum_{i=1}^{N} \Big\{ \underbrace{\tfrac{1}{2} x_i^T(t_f) Q_i x_i(t_f)}_{\text{penalty term (min. cost you have to pay)}} + \tfrac{1}{2}\int_0^{t_f} \Big[x_i^T(t) Q_i x_i(t) + \underbrace{u_i^T(t) R_i u_i(t)}_{\text{minimizing control effort (energy)}}$$
$$+ \underbrace{z_i^T(t) V_i z_i(t)}_{\text{minimizing interaction between subsystems}} \Big] dt \Big\} \tag{4.3.17}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where Q_i are $n_i \times n_i$ positive semidefinite matrices, R_i and V_i are $m_i \times m_i$ and $k_i \times k_i$ positive definite matrices with

$$n = \sum_{i=1}^{N} n_i, \quad m = \sum_{i=1}^{N} m_i, \quad k = \sum_{i=1}^{N} k_i, \quad k_i \le n_i \tag{4.3.18}$$

The physical interpretation of the last term in the integrand of (4.3.17) is difficult at this point. In fact, the introduction of this term, as will be seen later, is to avoid singular controls. The "goal coordination" or "interaction balance" approach of Mesarvic *et al.* (1970) as applied to the "linear-quadratic" problem by Pearson (1971) and reported by Singh (1980) and Jamshidi (1983) is now presented.

In this decomposition of a large interconnected linear system, the common coupling factors among its N subsystems are the "interaction" variables $z_i(t)$, which, along with (4.3.15)–(4.3.16), constitute the "coupling" constraints. This formulation is called "global" and is denoted by S_G. The following assumption is considered to hold. The global problem S_G is replaced by a family of N subproblems coupled together through a parameter vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ and denoted by $s_i(\alpha)$. $i = 1, 2, \ldots, N$. In other words, the global system problem S_G is "imbedded" or "enclosed" into a family of subsystem problems $s_i(\alpha)$ through an imbedding parameter (Sandell *et al.*, 1978) in such a way that for a particular value of a*, the subsystems $s_i(\mathrm{a}^*)$, $i = 1,2,\ldots,N$, yield the desired solution to S_G. In terms of hierarchical control notation, this imbedding concept is nothing but the notion of coordination, but in mathematical programming problem terminology, it is denoted as the "master" problem (Geoffrion, 1970). Figure 4.6 shows a two-level control structure of a large-scale system. Under this

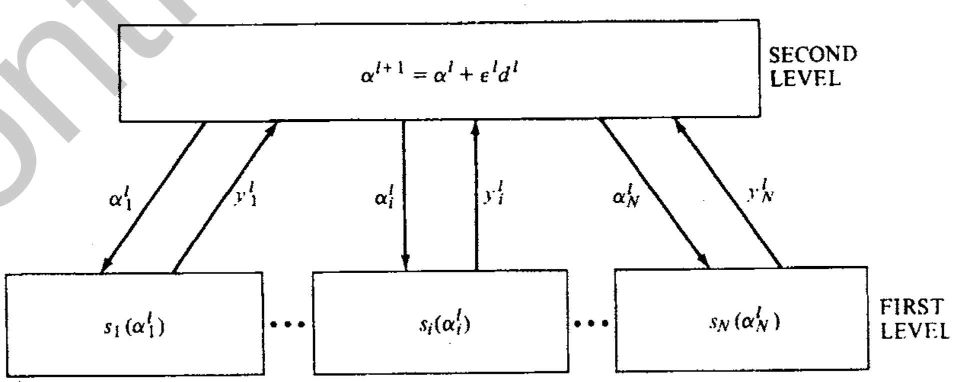


Figure 4.6 The two-level goal-coordination structure for dynamic systems.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

strategy, at ith iteration, each local controller i receives α_i^l from the coordinator (second-level hierarchy), solves $s_i(\alpha_i^l)$, and transmits (reports) some function y_i^l of its solution to the coordinator.

The coordinator, in turn, evaluates the next updated value of α, i.e.,

$$\alpha^{l+1} = \alpha^l + \varepsilon^l d^l \tag{4.3.19}$$

where ε^l is the lth iteration step size, and the update term d^l, as will be seen shortly, is commonly taken as a function of "interaction error":

$$e_i(\alpha(t),t) = z_i(\alpha(t),t) - \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij} x_j(\alpha(t),t) \tag{4.3.20}$$

The imbedded interaction variable $z_i(\cdot)$ in (4.3.20) can be considered as part of the control variable available to controller i, in which case the parameter vector $\alpha(t)$ serves as a set of "dual" variables or Langrange multipliers corresponding to interaction equality constraints (4.3.16). The fundamental concept behind this approach is to convert the original system's minimization problem into an easier maximization problem whose solution can be obtained in the two-level iterative scheme discussed above.

Let us introduce a dual function

$$q(\alpha) = \min_{x,u,z}\{L(x,u,z,\alpha)\} \tag{4.3.21}$$

subject to (4.3.15), where the Lagrangian $L(\cdot)$ is defined by

$$L(x,u,z,\alpha) = \sum_{i=1}^{N}\Bigg\{\tfrac{1}{2}x_i^T(t_f)Q_i x_i(t_f) + \tfrac{1}{2}\int_0^{t_f}\Bigg[x_i^T(t)Q_i x_i(t) + u_i^T(t)R_i u_i(t) + z_i^T(t)V_i z_i(t) + \underbrace{2\alpha_i^T\Bigg(z_i(t) - \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij}x_j(t)\Bigg)}_{\text{constraint eq. 20}}\Bigg]dt\Bigg\} \tag{4.3.22}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where the parameter vector α consists of k Lagrange multipliers. In this way the original constrained (subsystems interactions) optimization problem is changed to an unconstrained one, in other words, the constraint (4.3.16) is satisfied by determining a set of Lagrange multipliers α_i, $i = 1, 2, \ldots, k$. Under such cases, when the constraints are convex, the theorem of strong Lagrange duality (Geoffrion, 1971a,b; Singh, 1980) shows that

$$\underset{\alpha}{\text{Maximize}}\ q(\alpha) \equiv \underset{u}{\text{Minimize}}\ J \tag{4.3.23}$$

indicating that minimization of J in (4.3.17) subject to (4.3.15)–(4.3.16) is equivalent to maximizing the dual function $q(\alpha)$ in (4.3.21) with respect to α. To facilitate the solution of this problem, it is observed that for a given set of Lagrange multipliers $\alpha = \alpha^*$, the Lagrangian (4.3.22) can be rewritten as

$$\begin{aligned} L(x,u,z,\alpha^*) = \sum_{i=1}^{N} \Bigg\{ & \tfrac{1}{2} x_i^T(t_f) Q_i x_i(t_f) + \tfrac{1}{2}\int_0^{t_f} \Bigg[x_i^T(t) Q_i x_i(t) \\ & + u_i^T(t) R_i u_i(t) + z_i^T V_i z_i(t) + 2\alpha_i^{*T} z_i(t) \\ & - 2\sum_{\substack{j=1 \\ j\neq i}}^{N} \alpha_j^{*T} G_{ji} x_i(t) \Bigg] dt \Bigg\} \stackrel{\Delta}{=} \sum_{i=1}^{N} L_i(\cdot) \end{aligned} \tag{4.3.24}$$

which reveals that the decomposition is carried on to the Lagrangian in such a way that a sub-Lagrangian exists for each subsystem. Each subsystem would intend to minimize its own sub-Lagrangian L_i as defined by (4.3.24) subject to (4.3.15) and using the Lagrange multipliers α^* which are treated as known functions at the first level of hierarchy. The result of each such minimization would allow one to determine the dual function $q(\alpha^*)$ in (4.3.21). At the second level, where the solutions of all first-level subsystems are known, the value of $q(\alpha^*)$ would be improved by a typical unconstrained optimization such as the Newton's method, the gradient method, or the conjugate gradient method. The reason for a gradient-type method is due to the fact that the gradient of $q(\alpha)$ is defined by

$$\nabla_{\alpha} q(\alpha)\Big|_{\alpha_i=\alpha_i^*} = z_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij} x_j \stackrel{\Delta}{=} e_i, \quad i = 1,2,\ldots \tag{4.3.25}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is nothing but the subsystems' interaction errors, which are known through first-level solutions, and $\nabla_x f$ defines the gradient of f with respect to x. At the second level, the vector a is updated as indicated by (4.3.19) and Figure 4.6. If a gradient (steepest descent) method is employed, the vector d^l in (4.3.19) is simply the lth iteration's interaction error $e^l(t)$. However, a superior technique from a computational accuracy point of view is the conjugate gradient defined by

$$d^{l+1}(t) = e^{l+1}(t) + \gamma^{l+1} d^l(t), \quad 0 \le t \le t_f \tag{4.3.26}$$

where

$$\gamma^{l+1} = \frac{\int_0^{t_f} \left(e^{l+1}(t)\right)^T e^{l+1}(t) dt}{\int_0^{t_f} \left(e^l\right)^T e^l dt} \tag{4.3.27}$$

eq 19, 26, 27 Conjugate Gradient Approach

and $d^\circ = e^\circ$. Once the error vector $e(t)$ approaches zero, the optimum hierarchical control s results. Below, a step-by-step computational procedure for the goal coordination method of hierarchical control is given.

Algorithm 4.1. Goal Coordination Method

Step 1. For each first-level subsystem, minimize each sub-Lagrangian L_i using a known Lagrange multiplier $\alpha = \alpha^*$. Since the subsystems are linear, a Riccati equation formulation** can be used here. Store solutions.

Step 2. At the second level, a conjugate gradient iterative method similar to (4.3.26)–(4.3.27) is used to update $\alpha^*(t)$ trajectories like (4.3.19). Once the total system interaction error in normalized form

$$Error = \left(\sum_{j=1}^{N} \int_0^{t_f} dt \left\{ z_i - \sum_{\substack{j=1 \\ j \ne i}}^{N} G_{ij} x_j \right\}^T \left\{ z_i - \sum_{\substack{j=1 \\ j \ne i}}^{N} G_{ij} x_j \right\} dt \right) \Big/ \Delta t \tag{4.3.28}$$

is sufficiently small, an optimum solution has been obtained for the system. Here Δt is the step size of integration.

* Readers unfamiliar with the Riccati formulation can consult Section 4.3.2 on the Interaction Prediction Method.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Two examples follow that help illustrate the goal coordination or interaction balance approach. The first example, which was first suggested by Pearson (1971) and further treated by Singh (1980) and Jamshidi (1983), is used in a modified form. The second example represents the model of a multireach river pollution problem (Beck, 1974; Singh, 1975). The overall evaluation of the multilevel methods is deferred until Section 4.6, and the treatment of nonlinear multilevel systems is given in Chapter 6. Two alternatives to the solutions of this hierarchical control based on Taylor and Chebyshev series expansions are discussed in Section 4.6.

Example 4.3.1. Consider a 12th-order system introduced by Pearson (1971) and shown in Figure 4.7 with a state equation

$$\dot{x} = \left[\begin{array}{ccc|ccc|ccc|ccc}
0 & 1 & 0 & & & & & & & 0 & 0 & 0 \\
0 & 0 & 1 & & 0 & & & 0 & & 0 & 0 & 0 \\
-3 & -2 & -1 & & & & & & & 1 & 0 & \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & & & \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & & 0 & \\
1 & 0 & 0 & -1 & -3 & -2 & 0 & 1 & 0 & & & \\
\hline
 & & & 0 & 0 & 0 & 0 & 1 & 0 & & & \\
 & 0 & & 0 & 0 & 0 & 0 & 0 & 1 & & 0 & \\
 & & & 0 & 1 & 0 & -1 & -2 & -3 & & & \\
\hline
0 & 0 & 0 & & & & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & & 0 & & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & & & & 1 & 0 & 0 & -3 & -2 & -1
\end{array}\right] x
+ \left[\begin{array}{cc}
0 & 0 \\ 0 & 0 \\ 1 & 0 \\ \hline
0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline
0 & 0 \\ 0 & 0 \\ 0 & 1 \\ \hline
0 & 0 \\ 0 & 0 \\ 0 & 0
\end{array}\right] u \tag{4.3.29}$$

and a quadratic cost function

$$J = \int_0^{10} \left\{ x^T(t)Qx(t) + u^T(t)Ru(t) \right\} dt \tag{4.3.30}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

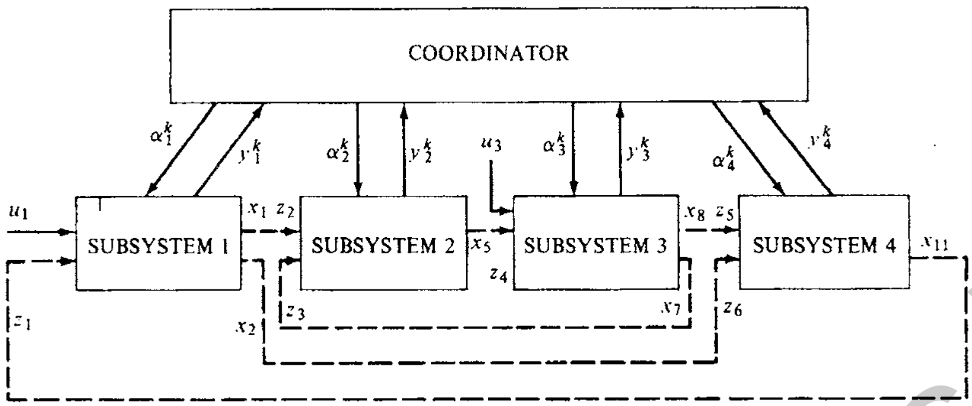


Figure 4.7 A block diagram for system of Example 4.3.1.

with

$$Q = diag\{Q_1, Q_2, Q_3, Q_4\}, \quad R = diag\{R_1, R_2, R_3, R_4\}$$

where

$$Q_i = diag(1,1,0), \quad R_i = 1, \quad i = 1,2,3,4$$

The system output vector is given by

$$y = Cx = \left[\begin{array}{ccc:ccc:ccc:ccc}
1 & 0 & 0 & & 0 & & & 0 & & & 0 & \\
0 & 1 & 0 & & & & & 0 & & & 0 & \\
\hdashline
& 0 & & 1 & 0 & 0 & & 0 & & & 0 & \\
& & & 0 & 1 & 0 & & & & & & \\
\hdashline
& 0 & & & 0 & & 1 & 0 & 0 & & 0 & \\
& & & & & & 0 & 1 & 0 & & & \\
\hdashline
& 0 & & & 0 & & & 0 & & 1 & 0 & 0 \\
& & & & & & & & & 0 & 1 & 0
\end{array}\right] x \qquad (4.3.31)$$

It is desired to find a hierarchical control strategy through the interaction balance (goal coordination) approach.

SOLUTION: From the schematic of the system shown in Figure 4.7 (dotted lines) and the state matrix in (4.3.29), it is clear that there are four third-order subsystems coupled together through six equality (number of dotted lines in Figure 4.7) constraints given by

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$e=(e_1,e_2,e_3,e_4,e_5,e_6)$$
$$=\left[(z_1-x_{11}),(z_2-x_1),(z_3-x_7),(z_4-x_5),(z_5-x_8),(z_6-x_2)\right] \quad (4.3.32)$$

where e_i, i = 1,2,...,6, represents the interaction errors between the four subsystems. The first-level subsystem problems were solved through a set of four third-order matrix Riccati equations final value problem

don't have initial conditions

Riccati eqn

$$\dot{K}_i(t)=-A_i^T K_i(t)-K_i(t)A_i+K_i(t)S_iK_i(t)-Q_i, \quad K_i(10)=0 \quad (4.3.33)$$

solve problem in negative time

where $K_i(t)$ is an $n_i \times n_i$ positive-definite symmetric Riccati matrix and S_i = $S_i = B_iR_i^{-1}B_i^T$. The "integration-free," or "doubling," method of solving the differential matrix Riccati equation proposed by Davison and Maki (1973) and overviewed by Jamshidi (1980) was used for computer solution of (4.3.33). The subsystems' state equations were solved by a standard fourth-order Runge-Kutta method, while the second-level iterations were performed by the conjugate gradient scheme (4.3.19), (4.3.26)–(4.3.27) utilizing the cubic spline interpolation (Hewlett-Packard, 1979) to evaluate appropriate numerical integrals. The step size was chosen to be Δt = 0.1 as in earlier treatments of this example (Pearson, 1971; Singh, 1980). The conjugate gradient algorithm resulted in a decrease in error from I to about 10^{-3} in six iterations, as shown in Figure 4.8, which was in close agreement with previously reported results of a modified version of system (4.3.29) by Singh (1980). Now let us consider the second example.

Example 4.3.2. Consider a two-reach model of a river pollution control problem,

$$\dot{x}=\left[\begin{array}{cc|cc}-1.32 & 0 & 0 & 0\\ -0.32 & -1.2 & 0 & 0\\ \hline 0.90 & 0 & -1.32 & 0\\ 0 & 0.9 & -0.32 & -1.2\end{array}\right]x+\left[\begin{array}{c|c}0.1 & 0\\ 0 & 0\\ \hline 0 & 0.1\\ 0 & 0\end{array}\right]u \quad (4.3.34)$$

where each reach (subsystem) of the river has two states— x_1 is the concentration of biochemical oxygen demand (BOD),* and x_2 is the concentration of dissolved oxygen (DO)—and its control u_1 is the BOD of the

*The biochemical oxygen demand represents the rate of absorption of oxygen by decomposing organic matter.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

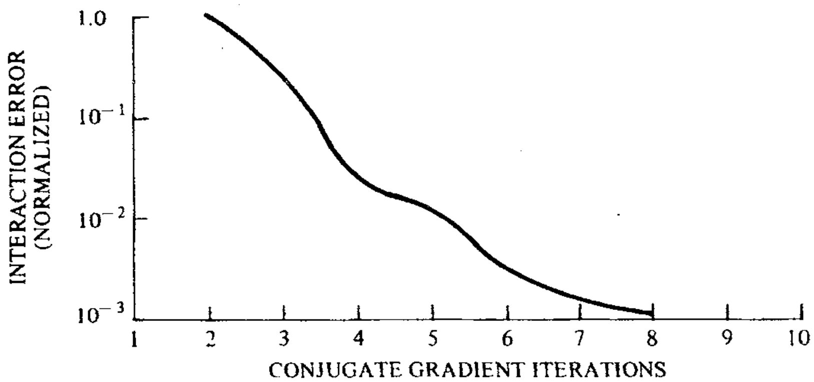


Figure 4.8 Normalized interaction error vs. congugate gradient iterations for Example 4.3.1.

effluent discharge into the river. For a quadratic cost function

$$J = \tfrac{1}{2}\int_0^5 \left(x^T Q x + u^T R u\right) dt \tag{4.3.35}$$

with $Q = \text{diag}\{2,4,2,4\}$ and $R = \text{diag}\ \{2,2\}$, it is desired to find an optimal control which minimizes (4.3.35) subject to (4.3.34) and $x(0) = (1\ 1\ \text{-}\ 1\ 1)^T$.

SOLUTION: As seen from (4.3.34)–(4.3.35), the two first-level problems are identical, and a second-order matrix Riccati equation is solved by integrating (4.3.33) using a fourth-order Runge-Kutta method for $\Delta t = 0.1$. The interaction error for this example reduced to about 10^{-5} in 15 iterations, as shown in Figure 4.9. The optimum BOD and DO concentrations of the two reaches of the river are shown in Figure 4.10.

4.3.2 Interaction Prediction Method

An alternative approach in optimal control of hierarchical systems which has both open- and closed-loop forms is the interaction prediction method based on the initial work of Takahara (1965), which avoids second-level gradient-type iterations. Consider a large-scale linear interconnected system which is decomposed into N subsystems, each of which is described by

Dynamic Constraint

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + C_i z_i(t), \quad x_i(0) = x_{io}, \quad i = 1,2,\ldots,N \tag{4.3.36}$$

where the interaction vector z_1 is

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$z_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} x_j(t) \qquad \text{static constraint} \tag{4.3.37}$$

The optimal control problem at the first level is to find a control $u_i(t)$ which satisfies (4.3.36)–(4.3.37) while minimizing a usual quadratic cost function

$$J_i = \underbrace{\tfrac{1}{2} x_i^T(t_f) Q_i x_i(t_f)}_{\text{penalty}} + \tfrac{1}{2} \int_0^{t_f} \underbrace{\left\{ x_i^T(t) Q_i x_i(t) + u_i^T(t) R_i(t) u_i(t) \right\}}_{\text{cost}} dt \tag{4.3.38}$$

This problem can be solved by first introducing a set of Lagrange multipliers $\alpha_i(t)$ and costate vectors $p_i(t)$ to augment the "interaction" equality constraint (4.3.37) and subsystem dynamic constraint (4.3.36) to the cost function's integrand; i.e., the ith subsystem Hamiltonian is defined by

$$\begin{aligned} H_i = {} & \tfrac{1}{2} x_i^T(t) Q_i x_i(t) + \tfrac{1}{2} u_i^T(t) R_i u_i(t) + \alpha_i^T z_i \\ & - \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_j^T G_{ji} x_i + p_i^T (A_i x_i + B_i u_i + C_i z_i) \end{aligned} \tag{4.3.39}$$

co-state variable

Then the following set of necessary conditions can be written:

$$\dot{p}_i = -\partial H_i / \partial x_i = -Q_i x_i - A_i^T p_i + \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ji}^T \alpha_j(t) \tag{4.3.40}$$

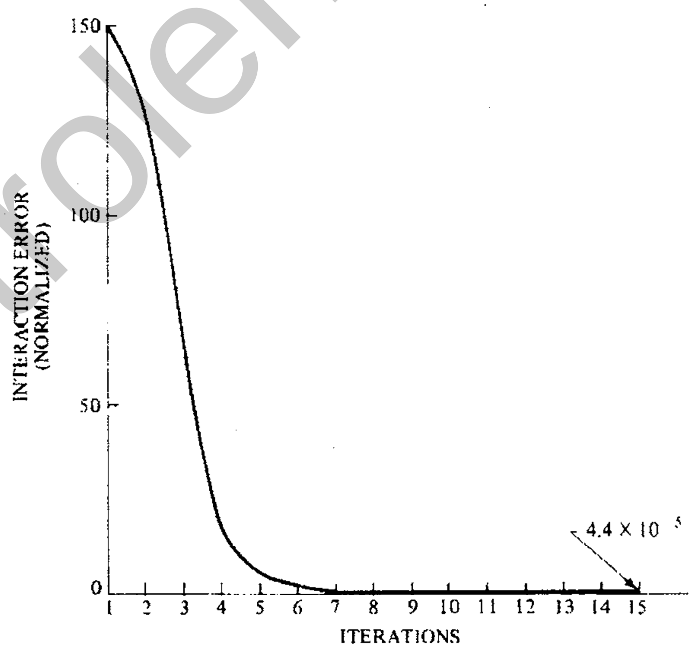


Figure 4.9 Interaction error behavior for river pollution system of Example 4.3.2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

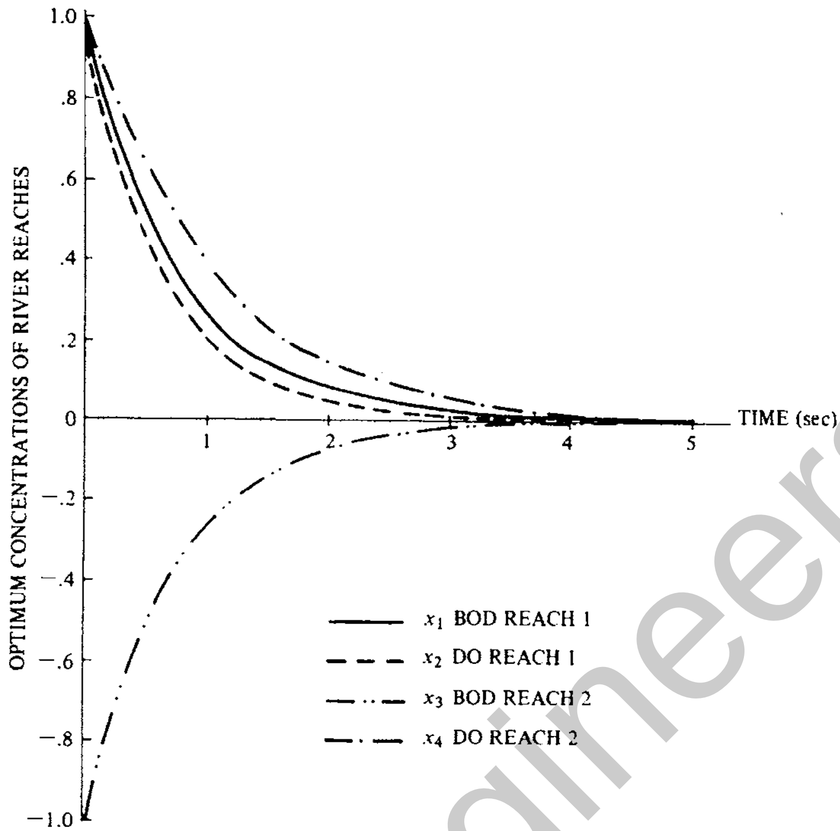


Figure 4.10 Optimum BOD and DO concentrations for the two-reach model of a river pollution control problem in Example 4 3.2.

$$p_i(t_f) = \partial\left(\tfrac{1}{2} x_i^T(t_f) Q_i x_i(t_f)\right) / \partial x_i(t_f) = Q_i x_i(t_f) \tag{4.3.41}$$

$$\dot{x}_i(t) = \partial H_i / \partial p_i = A_i x_i(t) + B_i u_i(t) + C_i z_i, \quad x_i(0) = x_{io} \tag{4.3.42}$$

$$0 = \partial H_i / \partial u_i = R_i u_i(t) + B_i^T p_i(t) \tag{4.3 43}$$

minimizing Hamiltonian

where the vectors $\alpha_i(t)$ and $z_i(t)$ are no longer considered as unknowns at the first level, and in fact $z_i(t)$ is augmented with $\alpha_i(t)$ to constitute a higher-dimensional "coordination vector," which will be obtained shortly. For the purpose of solving the first-level problem, it suffices to assume $\left(\alpha_i^T(t) \,|\, z_i^T(t)\right)^T$ as known. Note that $u_i(t)$ can be eliminated from (4.3.43),

$$u_i(t) = -R_i^{-1} B_i^T p_i(t) \tag{4.3.44}$$

must be pd

and substituted into (4.3.40)–(4.3.42) to obtain

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{x}_i(t) = A_i x_i(t) - S_i p_i(t) + C_i z_i(t), \quad x_i(0) = x_{io} \tag{4.3.45}$$

$$\dot{p}_i(t) = -Q_i x_i(t) - A_i^T p_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ji}^T \alpha_j(t), \quad p_i(t_f) = Q_i x_i(t_f) \tag{4.3.46}$$

which constitute a linear two-point boundary-value (TPBV) problem and, as in (4.3.33), $S_i \triangleq B_i R_i^{-1} B_i^T$. It can be seen that this TPBV problem can be decoupled by introducing a matrix Riccati formulation. Here it is assumed that

$$p_i(t) = K_i(t) x_i(t) + g_i(t) \tag{4.3.47}$$

where $g_i(t)$ is an n_i-dimensional open-loop "adjoint," or "compensation," vector. If both sides of (4.3.47) are differentiated and $\dot{p}_i(t)$ and $\dot{x}_i(t)$ from (4.3.46) and (4.3.45) are substituted into it, making repeated use of (4.3.47) and equating coefficients of the first and zeroth powers of $x_i(t)$, the following matrix and vector differential equations result:

$$\dot{K}_i(t) = -K_i(t) A_i - A_i^T K_i(t) + K_i(t) S_i K_i(t) - Q_i \tag{4.3.48}$$

$$\dot{g}_i(t) = -(A_i - S_i K_i(t))^T g_i(t) - K_i(t) C_i z_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ji}^T \alpha_j(t) \tag{4.3.49}$$

whose final conditions $K_i(t_f)$ and $g_i(t_f)$ follow from (4.3.41) and (4.3.47).

$$K_i(t_f) = Q_i, \quad g_i(t_f) = 0 \tag{4.3.50}$$

Following this formulation, the first-level optimal control (4.3.44) becomes

$$u_i(t) = -R_i^{-1} B_i^T K_i(t) x_i(t) - R_i^{-1} B_i^T g_i(t) \tag{4.3.51}$$

which has a partial feedback (closed-loop) term and a feedforward (open-loop) term. Two points are made here. First, the solution of the differential symmetric matrix Riccati equation which involves $n_i(n_i + 1)/2$ nonlinear scalar equations is independent of the initial state $x_i(0)$. The second point is that unlike $K_i(t)$, $g_i(t)$ in (4.3.49), by virtue of $z_i(t)$, is dependent on $x_i(0)$. This property will be used in Section 4.4 to obtain a completely closed-loop control in a hierarchical structure.

The second-level problem is essentially updating the new coordination vector $\left(\alpha_i^T(t) \,\vert\, z_i^T(t)\right)^T$. For this purpose, define the additively

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

separable Larangian

$$L=\sum_{i=1}^{N}L_i=\sum_{i=1}^{N}\Bigg(\tfrac{1}{2}x_i^T(t_f)Q_ix_i(t_f)+\int_0^{t_f}\Big\{\tfrac{1}{2}x_i^T(t)Q_ix_i(t)+\tfrac{1}{2}u_i^T(t)R_iu_i(t)+\alpha_i^T(t)z_i(t)-\sum_{\substack{j=1\\j\neq i}}^{N}\alpha_j^T(t)G_{ji}x_i(t)+p_i^T(t)\big[-\dot{x}_i(t)+A_ix_i(t)+B_iu_i(t)+C_iz_i(t)\big]\Big\}dt\Bigg) \tag{4.3.52}$$

The values of $\alpha_i(t)$ and $z_i(t)$ can be obtained by

$$0=\partial L_i(\cdot)/\partial z_i(t)=\alpha_i(t)+C_i^Tp_i(t) \tag{4.3.53}$$

$$0=\partial L_i(\cdot)/\partial\alpha_i(t)=z_i(t)-\sum_{\substack{j=1\\j\neq i}}^{N}G_{ij}x_j(t) \tag{4.3.54}$$

which provide

$$\alpha_i(t)=-C_i^Tp_i(t),\quad z_i(t)=\sum_{\substack{j=1\\j\neq i}}^{N}G_{ij}x_j(t) \tag{4.3.55}$$

The second-level coordination procedure at the $(l+1)$th iteration is simply

$$\begin{bmatrix}\alpha_i(t)\\ z_i(t)\end{bmatrix}^{l+1}=\begin{bmatrix}-C_i^Tp_i(t)\\ \sum_{\substack{j=1\\j\neq i}}^{N}G_{ij}x_j(t)\end{bmatrix}^{l} \tag{4.3.56}$$

The interaction prediction method is formulated by the following algorithm.

Algorithm 4.2. Interaction Prediction Method for Continuous-Time Systems

Step 1. Solve N independent differential matrix Riccati equations (4.3.48) with final condition (4.3.50) and store $K_i(t)$, $i=1, 2, \ldots, N$. Initialize an arbitrary value for $\alpha_i(t)$ and find corresponding value for $z_i(t)$.

Step 2. At the lth iteration use values of $\alpha_i^l(t)$, $z_i^l(t)$ to solve the "ad-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

joint" equation (4.3.49) with final condition (4.3.50). Store $g_i(t)$, $i = 1, 2, \ldots, N$.

Step 3. Solve the state equation

$$\dot{x}_i(t) = \left(A_i - S_i K_i(t)\right)x_i(t) - S_i g_i(t) + C_i z_i(t), \quad x_i(0) = x_{io} \tag{4.3.57}$$

and store $x_i(t)$, $i = 1, 2, \ldots, N$.

Step 4. At the second level, use the results of Steps 2 and 3 and (4.3.56) to update the coordination vector

$$\left(\alpha_i^T(t) \mid z_i^T(t)\right)^T$$

Step 5. Check for the convergence at the second level by evaluating the overall interaction error

$$e(t) = \sum_{i=1}^{N} \int_0^{t_f} \left\{ z_i(t) - \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} x_j(t) \right\}^T \left\{ z_i(t) - \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} x_j \right\} dt/\Delta t \tag{4.3.58}$$

Step 6. If a desired convergence is achieved, stop. Otherwise, set $l = l + 1$ and go to Step 2.

It must be noted that depending on the type of digital computer and its operating system, subsystem calculations may be done in parallel and that the N matrix Riccati equations at Step 1 are independent of $x_i(0)$, and hence they need to be computed once regardless of the number of second-level iterations in the interaction prediction algorithm (4.3.56). It is further noted that unlike the goal coordination methods, no $z_i(t)$ term is needed in the cost function, which was intended, as is discussed in next section, to avoid singularities.

The interaction prediction method, originated by Takahara (1965), has been considered by many researchers who have made significant contributions to it. Among them are Titli (1972), who called it the "mixed method" (Singh 1980), and Cohen *et al.* (1974), who have presented more refined proofs of convergence than those originally suggested. Smith and Sage (1973) have extended the scheme to nonlinear systems which will be considered in Chapter 6. A comparison of the interaction prediction, goal coordination methods, and integration-free approaches of Section 4.4

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

will be discussed in Section 4.5. The following two examples, followed by a CAD example, illustrate the interaction prediction method.

Example 4.3.3. Consider a fourth-order system

$$\dot{x} = \left[\begin{array}{cc|cc} 2 & 0.1 & 0.01 & 0 \\ 0.2 & -1 & -.10 & -0.5 \\ \hline 0.05 & 0.15 & 1 & 0.05 \\ 0 & -0.2 & -0.25 & -1.2 \end{array}\right] x + \left[\begin{array}{c|c} 1 & 0 \\ 0.1 & 0 \\ \hline 0 & 0.5 \\ 0 & 0.25 \end{array}\right] u \tag{4.3.59}$$

with $x(0) = (-1,0.1,1.0,-0.5)^T$ and a quadratic cost function with $Q = \text{diag}(2,1,1,2)$, $R = \text{diag}(1, 2)$ and no terminal penalty. It is desired to use the interaction prediction method to find an optimal control for $t_f = 1$.

SOLUTION: The system was divided into two second-order subsystems, and steps outlined in Algorithm 4.2 were applied. At the first step, two independent differential matrix Riccati equations were solved by using both the doubling algorithm of Davison and Maki (1973) and the standard Runge-Kutta methods. The elements of the Riccati matrix were fitted in by quadratic polynomial in the Chebyschev sense (Newhouse, 1962) for computational convenience:

$$K_1(t) = \begin{bmatrix} 4.44 + 0.32t - 1.26t^2 & 0.09 + 0.007t - 0.027t^2 \\ 0.09 + 0.007t - 0.027t^2 & 0.5 + 0.034t - 0.141t^2 \end{bmatrix}$$

$$K_2(t) = \begin{bmatrix} 2.87 - 5.26t + 2.42t^2 & -0.1 + 0.16t - 0.054t^2 \\ -0.1 + 0.16t - 0.054t^2 & 0.73 + 0.118t - 0.83t^2 \end{bmatrix} \tag{4.3.60}$$

At the first level, a set of two second-order adjoint equations of the form (4.3.49) and two subsystem state equations as in Step 3 of Algorithm 4.2 using the fourth-order Runge-Kutta method and initial values

$$\alpha_1(t) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad x_1(0) = \begin{bmatrix} -1 \\ 0.1 \end{bmatrix}, \quad z_1(t) = G_{12}x_2(0) = \begin{bmatrix} 0.01 \\ 0.35 \end{bmatrix}$$

$$\alpha_2(t) = \begin{bmatrix} 0.75 \\ 0.75 \end{bmatrix}, \quad x_2(0) = \begin{bmatrix} 1.0 \\ -0.5 \end{bmatrix}, \quad z_2(t) = G_{21}x_1(0) = \begin{bmatrix} -0.035 \\ -0.02 \end{bmatrix} \tag{4.3.61}$$

were solved. At the second level, the interaction vectors $[\alpha_{11}(t), \alpha_{12}(t), z_{11}(t), z_{12}(t)]$ and $[\alpha_{21}(t), \alpha_{22}(t), z_{21}(t), z_{22}(t)]^T$ were predicted using the recursive relations (4.3.56), and at each information exchange iteration the total interaction error (4.3.58) was evaluated for $\Delta t = 0.1$ and a cubic spline interpolator program. The interaction error was reduced to 3.5113456×10^{-6} in

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

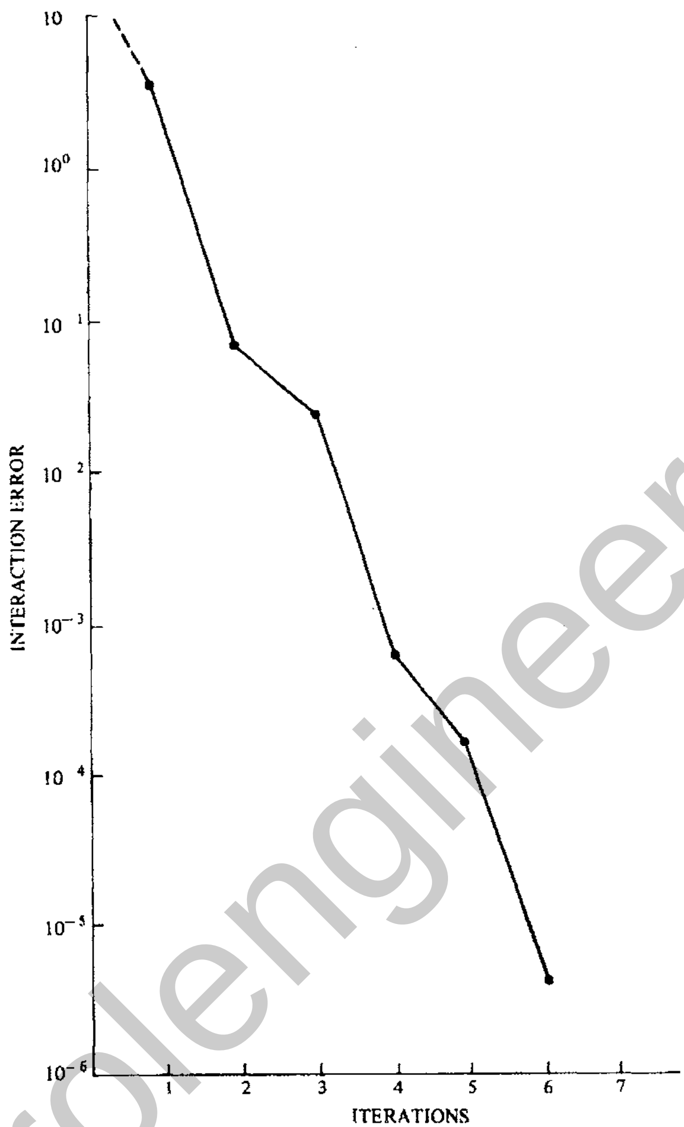
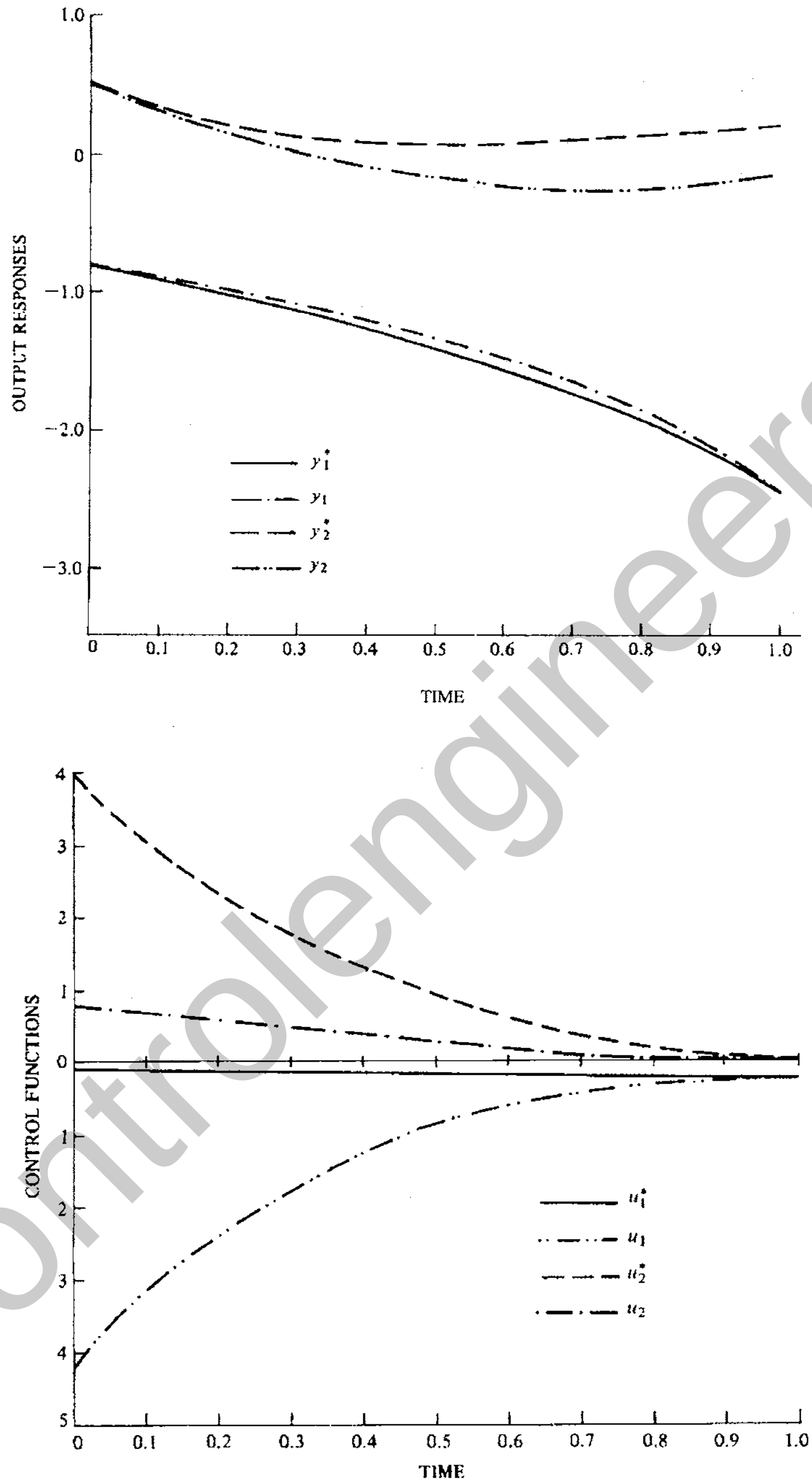


Figure 4.11 Interaction error vs. iterations for the interaction prediction in Example 4.3.3

six iterations, as shown in Figure 4.11. The optimum outputs for $C_i = (1\ 1)$ and control signals were obtained. Next, for the sake of comparison, the original system (4.3.59) was optimized by solving a fourth-order time-varying matrix Riccati equation by backward integration and solving for $x_i(t)$, $i = 1,2,3,4$; $y_j(t)$ and $u_j(t)$, $j = 1,2$. The outputs and control signals for both hierarchical and exact centralized cases are shown in Figure 4.12. Note the relatively close correspondence between the outputs for the original coupled and hierarchical decoupled systems. However, as one would expect, the two controls are different.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a)

(b)

Figure 4.12 The optimal (centralized) and suboptimal (iteraction prediction) responses for Example 4.3.3: (a) outputs, (b) controls.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Now let us consider the second example.

Example 4.3.4. Consider an eighth-order system.

$$\dot{x} = \begin{bmatrix} -5 & 0 & 0 & 0 & 0.1 & -0.5 & -0.009 & 3 \\ 0 & -2 & 0 & 0 & -0.29 & 0 & -0.3 & 0.48 \\ -0.08 & -0.11 & -3.99 & -0.93 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 1.32 & -1.39 & -1 & -0.4 & 0 & 0 \\ 0 & 0 & -0.1 & -0.4 & -0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.17 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & -0.11 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u$$

It is desired to use interaction prediction approach to find u*.

SOLUTION: The system was decoupled into two fourth-order subsystems and $t_f = 2$, $\Delta t = 0.1$, $Q_1 = Q_2 = I_4$, $R_1 = R_2 = 1$ were chosen. The initial values of $\alpha_i^0(t)$, $i = 1,2$, and state $x(0)$ were assumed to be $\alpha_1^0(t) = (0.5 \quad 1 \quad -1 \quad 0)^T$, $\alpha_2^0(t) = (1 \quad 0 \quad -1 \quad 0)^T$, and $x(0) = (-1 \;\; -0.5 \;\; 1 \;\; 0.5 \;\; 1 \;\; -1 \;\; 0.5 \;\; 0.5)^T$. The convergence was very rapid, as shown in Figure 4.13. In just four second-level iterations the interaction error reduced to 2×10^{-4}. In fact there was excellent convergence for a variety of $x(0)$ and $\alpha_i^0(t)$.

CAD Example 4.3.1 Consider a fourth-order system of Example 4.3.1 in (4.3.59),

$$\dot{x} = \left(\begin{array}{cc|cc} 2 & 0.1 & 0.01 & 0 \\ 0.2 & -1 & -.10 & -0.5 \\ \hline 0.05 & 0.15 & 1 & 0.05 \\ 0 & -0.2 & -0.25 & -1.2 \end{array}\right) x + \left(\begin{array}{c|c} 1 & 0 \\ 0.1 & 0 \\ \hline 0 & 0.5 \\ 0 & 0.25 \end{array}\right) u$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

with $x(0) = (-1, 0.1, 1.0, -0.5)^T$ and a quadratic cost function with $Q =$ diag (2, 1, 1, 2), $R =$ diag (1, 2) and no terminal penalty. It is desired to use LSSPAK or similar software and the interaction prediction method to find an optimal control with $t_f = 2$.

SOLUTION: As before, the system is divided into two second-order subsystems and the subsystem's Riccati equations are solved using RICRKUT of LSSPAK/PC and their solutions are fitted into fourth-order polynomi-

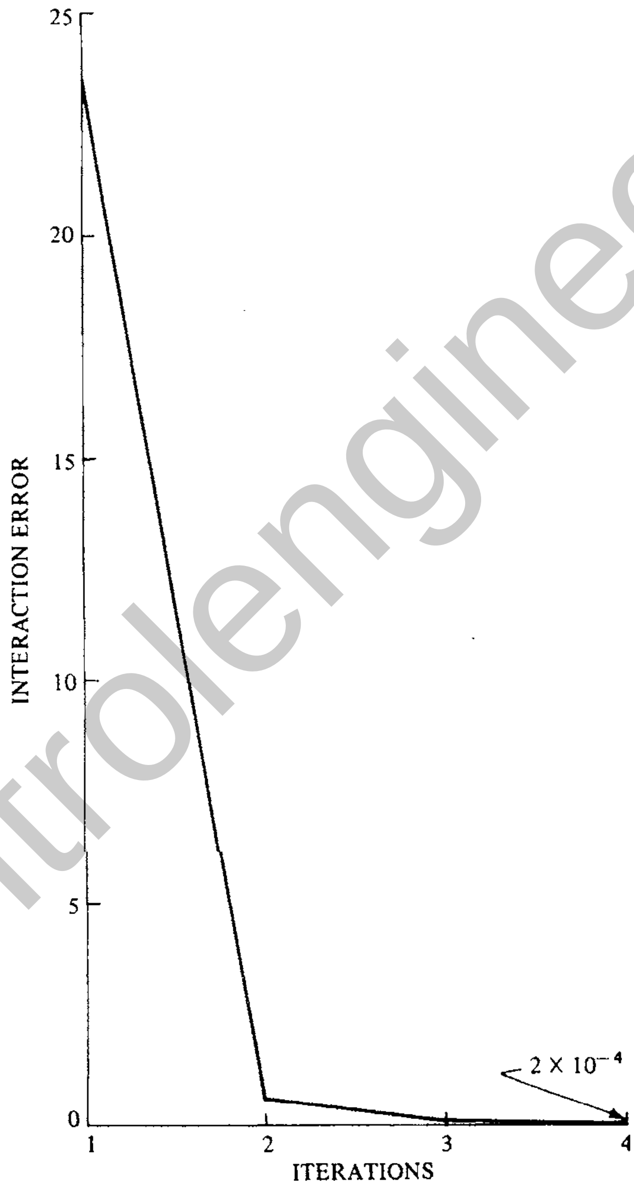


Figure 4.13 The interaction error vs. iterations for the eighth-order system of Example 4.3.4.

controlengineers.ir

controlengineers.ir
@controlengineers

als for computational convenience. Using program INTRPRD* of LSSPAK/PC, the interaction prediction algorithm is realized and converged in five iterations. The exact excerpts from running this CAD example follow. Instructions for plotting with the interaction prediction program will appear when you get a plot on the screen; hit return to return to the menu.

If you plan to dump plots to the printer, you must run the DOS file GRAPHICS prior to running this program. Then, when you wish to dump a plot, hit shift-PrtSc.

```
Optimization via the interaction prediction method.
Initial time (to): 0
Final time (tf): 2
Step size (Dt): .1
Total no. of 2nd level iterations = 6
Error tolerance for multi-level iterations - .00001
Order of overall large scale system = 4
Order of overall control vector (r) = 2
Number of subsystems in large scale system = 2

Matrix Subsystem state orders-n sub i
        0.200D+01
        0.200D+01

Matrix Subsystem input orders-r sub i
        0.100D+01
        0.100D+01

Polynomial approximation for the Ricatti matrices to be used.
Matrix Ricatti coefficients for SS# 1
        0.453D+01   -.259D+01   0.794D+01   -.762D+01   0.186D+01
        0.978D-01   -.793D-01   0.252D+00    .233D+00   0.571D-01
        0.490D+00   0.759D-02   -.109D+00   0.975D-01   -.531D-01

Matrix Ricatti coefficients for SS# 2
        0.112D+01   -.815D+01    .361D+01   0.455D+01    .105D+01
        -.149D+00   -.322D-01   0.697D-01   0.284D-01    .183D-01
        0.815D+00   0.642D-01   -.295D+00   0.305D+00   -.138D+00

System Matrix A
        0.200D+01   0.100D+00   0.100D-01   0.000D+00
        0.200D+00    .100D+01   0.100D+00    .500D+00
        0.500D-01   0.150D+00   0.100D+01   0.500D-01
        0.000D+00   -.200D+00    .250D+00   -.120D+01
```

*INTRPRD solves a multi-subsystem hierarchical control system using the method of interaction prediction. ~~See Appendix C for further information on large-scale systems software.~~

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Matrix Input Matrix B

0.100D+01	0.000D+00
0.100D+00	0.000D+00
0.000D+00	0.250D+00

Matrix Input Cost Function R

0.100D+01	0.000D+00
0.000D+00	0.200D+01

Matrix Lagrange Multiplier Initial Values

0.100D+01
0.100D+01
0.100D+01
0.100D+01

Matrix Initial conditions vector x0

-.100D+01
0.100D+00
0.100D+01
-.500D+00

Subsystem no. 1 at 2nd level iteration no. 1

Subsystem no. 2 at 2nd level iteration no. 1

At ~~second~~ first level iteration no. 1 interaction error = 0.347D+00

Subsystem no. 1 at 2nd level iteration no. 2

Subsystem no. 2 at 2nd level iteration no. 2

At ~~second~~ first level iteration no. 2 interaction error = 0.771D - 03

Subsystem no. 1 at 2nd level iteration no. 3

Subsystem no. 2 at 2nd level iteration no. 3

At second level iteration no. 3 interaction error = 0.507D - 03

Subsystem no. 1 at 2nd level iteration no. 4

Subsystem no. 2 at 2nd level iteration no. 4

At second level iteration no. 4 interaction error = 0.323D - 04

Subsystem no. 1 at 2nd level iteration no. 5

Subsystem no. 2 at 2nd level iteration no. 5

At second level iteration no. 5 interaction error = 0.310D - 05

Optimum responses are shown in Figure 4.14, while the convergence is depicted in Figure 4.15.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

More applications of the interaction prediction method are given in the problems section.

4.3.3 Goal Coordination and Singularities

When the goal coordination method was discussed earlier in (4.3.15)–(4.3.17), it was mentioned that the positive definite matrices S_i were introduced in the cost function (4.3.17) to avoid singularities. To see that this is in fact the case, let us reconsider the problem of minimizing L_i in (4.3.24) subject to (4.3.15). Let the ith subsystem's Hamiltonian be

$$H_i\left(x_i,u_i,z_i,p_i,\alpha_i^*\right)=1/2x_i^T(t)Q_ix_i(t) \\ +1/2u_i^T(t)R_iu_i(t)+1/2z_i^T(t)S_iz_i(t) \\ +\alpha_i^{*T}z_i-\sum_{\substack{j=1\\ j\neq i}}^{N}\alpha_j^{*T}G_{ji}x_i+p_i^T\left(A_ix_i+B_iu_i+C_iz_i\right) \tag{4.3.62}$$

As one of the necessary equations for the solution of the ith subsystem problem at the first level, we have

$$\partial H_i(\cdot)/\partial z_i=S_iz_i(t)+\alpha_i^*(t)+C_i^Tp_i(t)=0 \tag{4.3.63}$$

or

$$z_i(t)=-S_i^{-1}\left(C_i^Tp_i(t)+\alpha_i^*(t)\right) \tag{4.3.64}$$

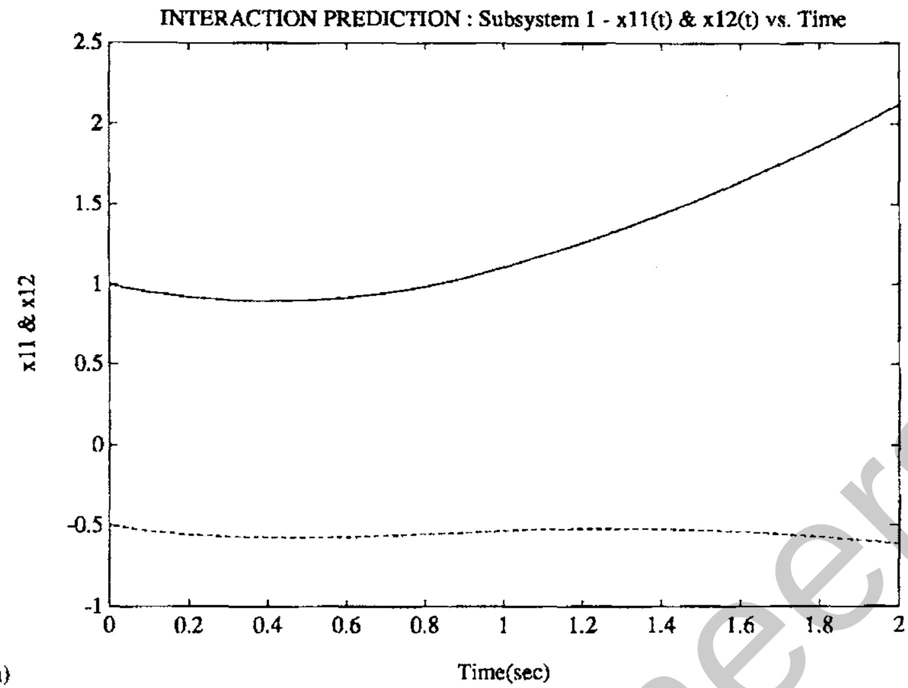
where a singular solution arises if the $z_i^T(t)S_iz_i(t)$ term does not appear in the cost function. Here two alternative approaches are given to avoid singularities at the first level. The following example illustrates the two approaches.

Example 4.3.5. Consider the following system:

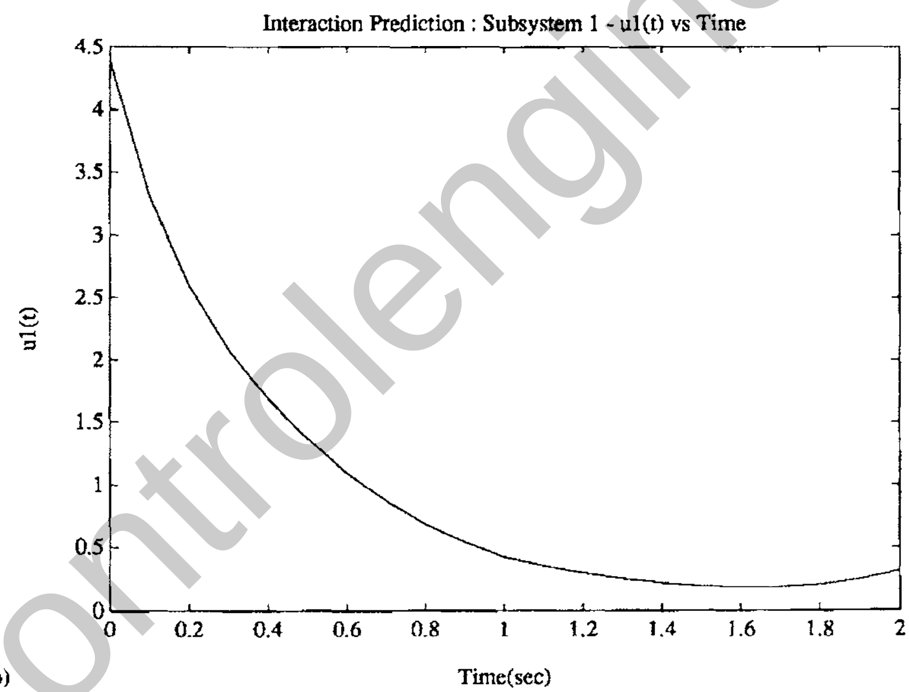
$$\dot{x}_1=-x_1+x_2+u_1,\quad x_1(0)=x_{10} \\ \dot{x}_2=-x_2+u_2,\quad x_2(0)=x_{20} \tag{4.3.65}$$

It is desired to find (u_1, u_2) such that (4.3.65) is satisfied while a quadratic cost function

controlengineers.ir
@controlengineers



(a)



(b)

Figure 4.14 Optimum time reponses for an interaction prediction solution of CAD Example 4.3.1.

Part (a) Subsystem No. 1 state variables.
Part (b) Subsystem No. 1 control variables.
Part (c) Subsystem No. 2 state variables.
Part (d) Subsystem No. 3 control variables.

controlengineers.ir
@controlengineers
اختصاصی مهندسی کنترل

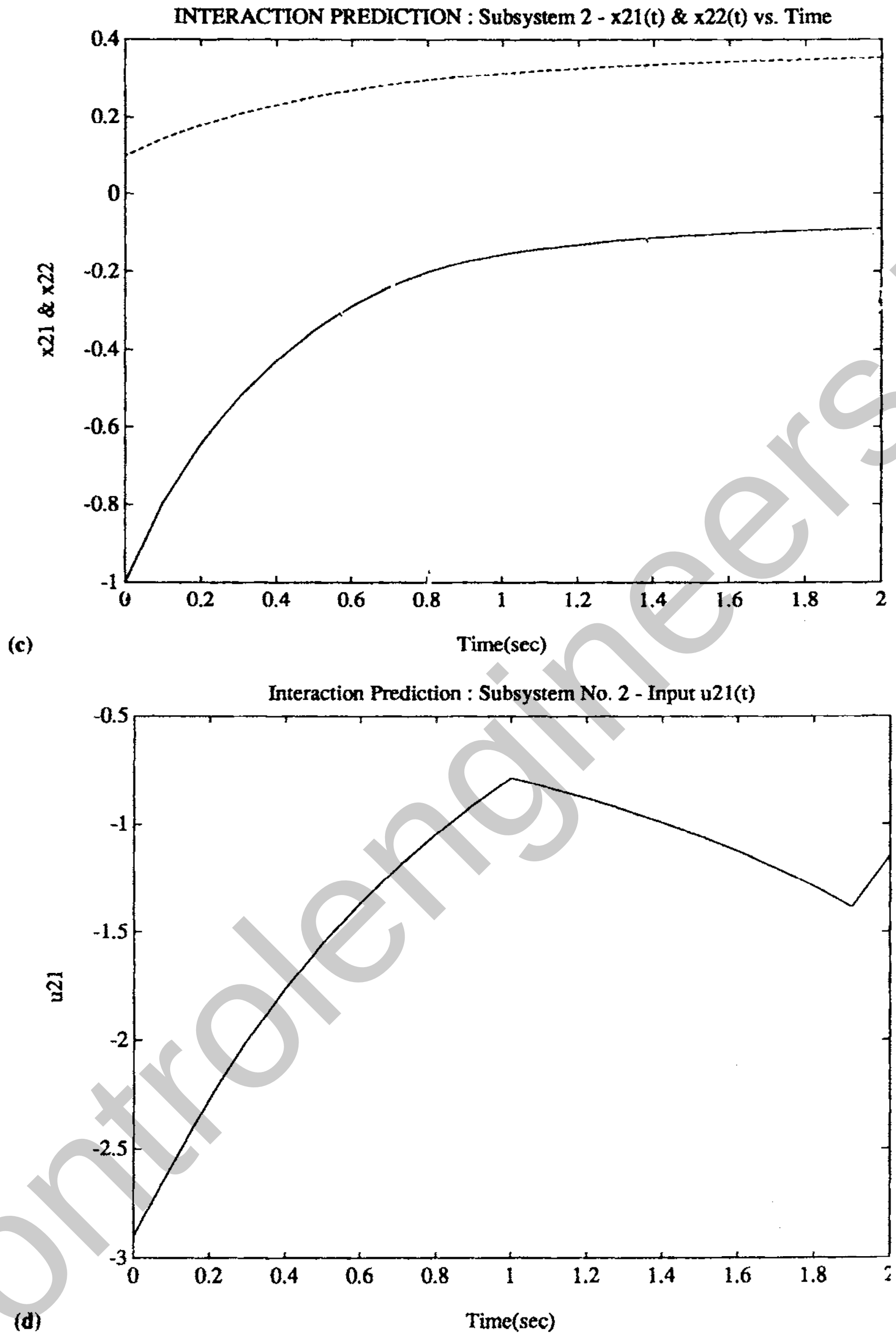


Figure 4.14 (continued)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

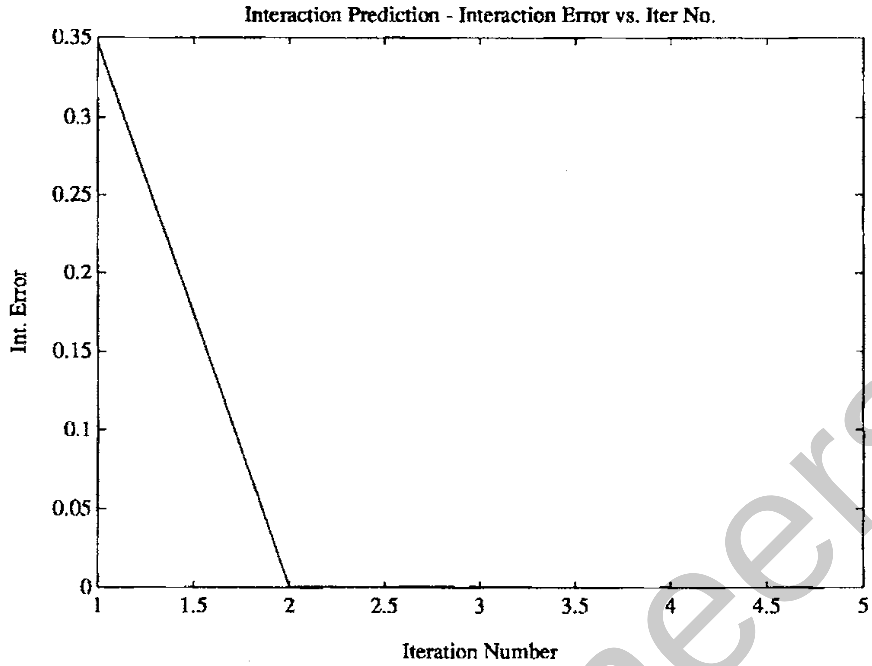


Figure 4.15 Interaction error vs. iteration for CAD Example 4.3.1.

$$J = \frac{1}{2}\int_0^1 \left(x_1^2 + x_2^2 + u_1^2 + u_2^2\right)dt \tag{4.3.66}$$

is minimized via the goal coordination method.

SOLUTION: From (4.3.65)–(4.3.66) it is seen that the system can be decomposed into two first-order subsystems.

$$\dot{x}_1 = -x_1 + u_1 + z_1, \quad x_1(0) = x_{10} \tag{4.3.67}$$

$$\dot{x}_2 = -x_2 + u_2, \quad x_2(0) = x_{20} \tag{4.3.68}$$

with interaction constraint

$$z_1 = x_2 \tag{4.3.69}$$

The problem in its present form has the following Hamiltonian:

$$\begin{aligned} H = &\left(\tfrac{1}{2}x_1^2 + \tfrac{1}{2}u_1^2 + \alpha z_1 - \alpha x_2 - p_1 x_1 + p_1 u_1 + p_1 z_1\right) \\ &+ \left(\tfrac{1}{2}x_2^2 + \tfrac{1}{2}u_2^2 - p_2 x_2 + p_2 u_2\right) \end{aligned} \tag{4.3.70}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

in which the interaction variable appears linearly. Thus, the application of goal coordination for the present formulation would lead to a singular problem, since z_1 appears linearly in (4.3.70). The following system reformulations of the problem would avoid singularities.

4.3.3.a Reformulation 1

Bauman (1968) suggests rewriting the interaction constraint (4.3.69) in quadratic form,

$$z_1^2 = x_2^2 \tag{4.3.71}$$

which would give the following necessary conditions for optimality at the first level:

$$\begin{aligned} 0 &= \partial H_1/\partial u_1 = u_1 + p_1, \quad 0 = \partial H_1/\partial z_1 = 2\alpha z_1 + p_1 \\ -\dot{p}_1 &= \partial H_1/\partial x_1 = x_1 - p_1, \quad p_1(1) = 0 \\ \dot{x}_1 &= \partial H_1/\partial p_1 = -x_1 + u_1 + z_1, \quad x_1(0) = x_{10} \end{aligned} \tag{4.3.72}$$

for the first subsystem and

$$\begin{aligned} 0 &= \partial H_2/\partial u_2 = u_2 + p_2 \\ -\dot{p}_2 &= \partial H_2/\partial x_2 = x_2 - p_2, \quad p_2(1) = 0 \\ \dot{x}_2 &= \partial H_2/\partial p_2 = -x_2 + u_2, \quad x_2(0) = x_{20} \end{aligned} \tag{4.3.73}$$

for the second subsystem. After the introduction of the Riccati formulation (4.3.72) and (4.3.73) will lead to

$$\dot{x}_1(t) = -\left\{1 + \left(1 + \frac{1}{2\alpha}\right)k_1(t)\right\}x_1(t), \quad x_1(0) = x_{10}$$

$$z_1(t) = -\left(\tfrac{1}{2}\alpha\right)k_1(t)x_1(t)$$

$$u_1(t) = -k_1(t)x_1(t)$$

controlengineers.ir

controlengineers.ir
@controlengineers

and

$$\dot{x}_2(t) = -(1+k_2(t))x_2(t), \quad x_2(0) = x_{20}$$
$$u_2(t) = -k_2(t)x_2(t)$$

where $k_i(t)$ is the ith subsystem scalar time-varying Riccati matrix. The coordination at the second level is achieved through the following iterations:

$$\alpha^{l+1}(t) = \alpha^l(t) + \varepsilon^l d^l(e(t))$$
$$e(t) = z_1^2(t) - x_2^2(t)$$

This reformulation would avoid singularities but makes the second-level iterations convergence very slow.

4.3.3.b Reformulation 2

Singh (1980) suggest an alternative formulation which not only avoids singularities but also gives good convergence. The procedure is based on solving for x in terms of the interaction vector z and substituting it into the cost function; i.e., z can be written in general as

$$z = Gx \text{ or } x = G^{-1}z \triangleq Vz$$

where G is assumed to be nonsingular and the reformulated Hamiltonian is

$$H(\cdot) = \tfrac{1}{2}z^T(t)V^TQVz(t) + \tfrac{1}{2}u^T(t)Ru(t) + \alpha^T z$$
$$-\alpha^T Gx + p^T(Ax + Bu + Cz)$$

For the example under consideration, the matrix G is singular, but a solution can still be found. Here the Hamiltonian is

$$H(\cdot) = \tfrac{1}{2}x_1^2 + \tfrac{1}{2}u_1^2 + \alpha z_1 - \alpha x_2 - p_1x_1 + p_1u_1 + p_1z_1$$
$$+ \tfrac{1}{2}z_1^2 + \tfrac{1}{2}u_2^2 - p_2x_2 + p_2u_2$$

with first-level subsystem problems being

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$0 = \partial H/\partial u_1 = u_1 + p_1, \quad 0 = \partial H/\partial z_1 = z_1 + \alpha + p_1$$
$$-\dot{p}_1 = \partial H/\partial x_1 = x_1 - p_1, \quad p_1(1) = 0$$
$$\dot{x}_1 = \partial H/\partial p_1 = -x_1 + u_1 + z_1, \quad x_1(0) = x_{10}$$

and

$$0 = \partial H/\partial u_2 = u_2 + p_2$$
$$-\dot{p}_2 = \partial H/\partial x_2 = \alpha - p_2, \quad p_2(1) = 0$$
$$\dot{x}_2 = \partial H/\partial p_2 = -x_2 + u_2, \quad x_2(0) = x_{20}$$

The second subsystem can be solved immediately, since the p_2-costate equation is decoupled from x_2 and can be solved backward in time and substituted into the x_2 equation, which would, in effect, mean that the solution of a Riccati equation has been avoided for this particular example. For the first subsystem, however, following the formulation of first-level problems in interaction prediction (4.3.40)–(4.3.51), both a Riccati and an open-loop adjoint (compensation) vector equation such as (4.3.48) and (4.3.49) need to be evaluated. For this example, the first subsystem problem is

$$\dot{x}_1(t) = -(1 + 2k_1(t))x_1(t) - 2g_1(t) - \alpha(t), \quad x_1(0) = x_{10}$$
$$\dot{k}_1(t) = 2k_1(t) + 2k_1^2(t) - 1, \quad k_1(1) = 0$$
$$\dot{g}_1(t) = (1 + 2k_1(t))g_1(t) + k_1(t)\alpha(t), \quad g_1(1) = 0$$
$$u_1(t) = -k_1(t)x_1(t) - g_1(t)$$

where two differential equations for $k_1(t)$ and $g_1(t)$ must be solved backward in time. Thus, while no auxiliary equation needs to be solved for the second subsystem, two such equations should be solved for the first subsystem. In general, this reformulation would require the solution of

$$\begin{aligned} \dot{x} &= Ax - Sp - GQ^{-1}(p + \alpha), \quad x(0) = x_0 \\ \dot{p} &= -G^T\alpha + A^T p, \quad p(t_f) = 0 \\ u &= -R^{-1}B^T p, \quad z = -GQ^{-1}(p + \alpha) \end{aligned} \tag{4.3.74}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which indicates that the costate vector p equation is decoupled from x and can be solved backward in time (eliminating a Riccati equation) and substituted in the top equation to find x. Since the A, B, Q, and R matrices are block-diagonal, problem (4.3.74) can be decomposed into N subsystem problems, provided that the term $(x^T(t)V^TQVz(t))$ is separable in z where $V = G^{-1}$.

4.4 Closed-Loop Hierarchical Control of Continuous-Time Systems

The last section dealt with open-loop hierarchical control of continuous-time systems in which the control depended on the system's initial condition $x(t_0)$. The scheme nearest to a closed-loop structure was the interaction prediction method which resulted in a partially closed-loop structure with an open-loop component which still depended on $x(t_0)$. Although one may always be able to measure $x(t_0)$, by the time an open-loop control is calculated and applied to the system, the initial state has most likely changed, thus resulting in unpredictable and undesirable responses. It is therefore worthwhile to construct closed-loop control laws which are independent of the initial state (Singh, 1980). The most likely case in which such a control structure is possible, as in nonhierarchical systems, is the linear quadratic regulator problem. Many authors have considered the problem of closed-loop control of hierarchical systems. Mesarovic *et al.* (1970) suggested a suboptimal structure which had an adaptive feature as the system evolves. Sage and his associates (Smith and Sage, 1973; Arafeh and Sage, 1974a, b) have used a similar suboptimal technique for the filter problem. Others (Cheneveaux, 1972; Cohen *et al.*, 1972, 1974) have considered the problem and suggested either a "partial" feedback controller or "complete" feedback controls which involve off-line calculations of the overall coupled large-scale system. An attractive extension of the partial feedback algorithm of Cohen *et al.* (1974) which provides "complete" closed-loop structure based on off-line calculations of feedback gains and their on-line implementation is due to Singh (1980).

In this section Singh's (1980) extension of the interaction prediction approach along with numerical examples are presented, in which the interaction prediction in its general sense was introduced with a partial closed-loop controller and an open-loop component given in (4.3.51). The main shortcoming of this structure is, as mentioned earlier, that the open-loop component depends on the initial state of the system and there is no apparent possibility for on-line implementation. To see this dependency of the open-loop component, let us consider the differential equation for the adjoint vector $g_i(t)$ in (4.3.49) and use (4.3.55) to eliminate $\alpha_i(t)$ and $z_i(t)$; i.e.,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{g}_i(t) = -\left(A_i - S_i K_i(t)\right)^T g_i(t) - K_i(t) C_i \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ij} x_j(t)$$

$$- \sum_{\substack{j=1 \\ j \neq i}}^{N} G_{ji}^T C_j^T \left(K_j(t) x_j(t) + g_j(t)\right) \tag{4.4.1}$$

which indicates that the vector $g_i(t)$ depends on the states of all the other subsystems and hence the initial state $x(t_0)$. The following theorem due to Singh (1980) relates the open-loop component

$$u^0(t) = -R^{-1} B^T g(t) \tag{4.4.2}$$

to the state $x(t)$ for the overall system which can be used in a hierarchical structure of a regulator problem.

Theorem 4.1. *The open-loop adjoint vector $g(t)$ and state $x(t)$ are related through the following transformation:*

$$g(t) = M(t_f, t)\, x(t) \tag{4.4.3}$$

PROOF: Rewriting the adjoint equations (4.4.1), the overall system's adjoint equation becomes

$$\begin{aligned} &\dot{g}(t) = -\left(A - SK(t) - CG\right)^T g(t) - \left(K(t)CG + G^T C^T K(t)\right) x(t) \\ &g(t_f) = 0 \end{aligned} \tag{4.4.4}$$

which can be represented in terms of its homogeneous and particular solutions.

$$g(t) = \Phi_1(t,t_0) g(t_0) - \int_{t_0}^{t_f} \Phi_1(t,\tau)\left(K(\tau)CG + G^T C^T K(\tau) x(\tau)\right) d\tau \tag{4.4.5}$$

where $\Phi_1(t, t_0)$ is the "state transition" matrix of $(A-SK(t)-CG)^T$. Note also that $K(t)$ is a block-diagonal matrix consisting of the subsystems, Riccati matrices, i.e., $K(t) = \mathrm{diag}\{K_1(t),...,K_i(t),...,K_N(t)\}$. However, for the composite system,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.4.6}$$

with standard quadratic cost

$$J = \tfrac{1}{2}\int_{t_0}^{t_f} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right]dt \tag{4.4.7}$$

the closed-loop optimal control system is well known:

$$\dot{x}(t) = \left(A - \tilde{S}P(t)\right)x(t) \tag{4.4.8}$$

or

$$x(t) = \Phi_2(t,t_0)x(t_0) \tag{4.4.9}$$

where $P(t)$ is the $n \times n$-dimensional time-varying Riccati matrix for the composite system and $\Phi_2(t, t_0)$ is the state transition matrix corresponding to the feedback system matrix $\left(A - \tilde{S}K\right)$ and $\tilde{S} = BR^{-1}B^T$. Now if we substitute $x(t)$ from (4.4.9) in (4.4.5),

$$\begin{aligned} g(t) = \Phi_1(t,t_0)g(t_0) - \int_{t_0}^{t_f} \Phi_1(t,\tau)\big(K(\tau)CG \\ + G^T C^T K(\tau)\big)\Phi_2(\tau,t_0)x(t_0)\,d\tau \end{aligned} \tag{4.4.10}$$

Using the final condition $g(t_f)$ in (4.4.4) at $t = t_f$ and making use of the properties of the state transition matrix, (4.4.10) can be used to solve for $g(t_0)$:

$$\begin{aligned} g(t_0) = \Phi_1\left(t_0,t_f\right)\int_{t_0}^{t_f} \Big[\Phi_1\left(t_f,\tau\right)\big(K(\tau)CG \\ + G^T C^T K(\tau)\big)\Phi_2(\tau,t_0)\Big]x(t_0)\,d\tau \end{aligned} \tag{4.4.11}$$

By moving the term $\Phi_1(t_0, t_f)$ inside the integral sign and taking advantage of product property of transition matrices, (4.4.11) is rewritten

$$\begin{aligned} g(t_0) = M\left(t_f,t_0\right)x(t_0) = \Big\{\int_{t_0}^{t_f} \Phi_1(t_0,\tau)\big(K(\tau)CG \\ + G^T C^T K(\tau)\Phi_2(\tau,t_0)\,d\tau\Big\}x(t_0) \end{aligned} \tag{4.4.12}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

or

$$g(t) = M(t_f, t)x(t) \tag{4.4.13}$$

which gives the desired relation. Q.E.D. ■

A corollary of the above theorem can be stated as follows. For the time-invariant case as $t_f \to \infty$, constant A, B, C, G, Q, and R matrices and time-invariant g and K, M become a constant transformation matrix (Sage, 1968). This corollary is used to find an approximate feedback law for the open-loop component.

The relation (4.4.13) is a sound theoretical property, but from an implementation point of view, it is not very desirable for a large-scale system because the overall system's Riccati differential equation must be solved and that defeats the original purpose of finding a control via hierarchical control. Singh (1980) has raised the point that for the time-invariant case, M can be computed easily, since near $t = 0$, M is constant while x and g are not. Therefore, it is suggested that if the first $n = \sum_{i=1}^{N} n_i$ values of $x(t_k)$ and $g(t_k)$ for $k = 0, 1, \ldots, n$ are evaluated and recorded, M can be approximately given by

$$M = \tilde{G}X^{-1} \tag{4.4.14}$$

where

$$\tilde{G} = \left[g(t_0) \middle| g(t_1) \middle| \cdots \middle| g(t_n) \right], \quad X = \left[x(t_0) \middle| x(t_1) \middle| \cdots \middle| x(t_n) \right] \tag{4.4.15}$$

and the inversion of X is done off-line. Note that if a time-varying M is desirable, it is possible to solve the problem with n initial conditions, i.e., $x(t_0), x(t_{0+1}), \ldots$, and form $n \times n$ time-dependent matrices $\tilde{G}(t)$ and $X(t)$ to find $M(t)$ for each integration step. In summary, the resulting control for the composite system can be formulated by

$$\begin{aligned} u &= -R^{-1}B^T Kx - R^{-1}B^T g \\ &= -R^{-1}B^T (K + M)x = -Fx \end{aligned} \tag{4.4.16}$$

It is noted that the above gains are all independent of $x(t_0)$, and the matri-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ces R, B, K, and M are obtained from decentralized calculations. The following example describes the above feedback law.

Example 4.4.1. Let us reconsider the system in Example 4.3.3 with $t_f = 4$, G_{12}, and G_{21}, matrices switched. It is desired to find a feedback gain matrix F.

SOLUTION: The decomposed system Riccati matrices at $t = 0$ are

$$K_1 = \begin{bmatrix} 4.440 & 0.093 \\ 0.093 & 0.498 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 2.9067 & -0.1010 \\ -0.1010 & 0.7522 \end{bmatrix}$$

The problem was simulated on LSSPAK. After six iterations of interaction prediction at the second level, $\tilde{G}$ and X were determined to be

$$\tilde{G} = 10^{-1} \begin{bmatrix} -0.78 & -2.13 & -1.38 & 1.24 \\ -0.73 & -2.18 & -1.20 & 1.28 \\ -0.69 & -2.20 & -1.02 & 1.29 \\ -0.66 & -2.16 & -0.85 & 1.24 \end{bmatrix}$$

$$X = \begin{bmatrix} 1.0 & -0.500 & -1.00 & 0.100 \\ 0.77 & -0.510 & -1.08 & -0.117 \\ 0.59 & -0.516 & -1.17 & 0.127 \\ 0.45 & -0.516 & -1.28 & 0.135 \end{bmatrix}$$

the partial feedback matrix M to be

$$M = \tilde{G}X^{-1} = \begin{bmatrix} 63.35 & -210.14 & 229.97 & -82.88 \\ 66.72 & -221.3 & 242.26 & -87.40 \\ 77.32 & -258.4 & 285.52 & -104.2 \\ 65.70 & -218.4 & 239.66 & -86.70 \end{bmatrix}$$

and the overall approximate feedback gain matrix based on hierarchical control to be

$$F = R^{-1}B^T(K + M) = \begin{bmatrix} 74.5 & -232.1 & 254.2 & -91.6 \\ 27.5 & -91.89 & 102.0 & -36.82 \end{bmatrix}$$

controlengineers.ir

controlengineers.ir
@controlengineers

4.5 Series Expansion Approach to Hierarchical Control

In the previous section, we discussed two approaches for optimal hierarchical control of linear large-scale systems—goal coordination and interaction prediction. The first-level (sybsystem) problem of both approaches constitutes the minimization of a cost function,

$$J_i = \tfrac{1}{2}\int_0^{t_f} \left(x_i^T Q_i x_i + u_i^T R_i u_i\right) dt \tag{4.5.1}$$

subject to a dynamic constraint,

$$\dot{x}_i = A_i x_i + B_i u_i + C_i z_i, \quad x_i(0) = x_{i0} \tag{4.5.2}$$

and a static interaction equality constraint,

$$z_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij} x_j \tag{4.5.3}$$

where all the matrices, their properties, and dimensions were appropriately defined in Section 4.3.1. The solution of this subsystem-level problem, as seen before, constitutes a Riccati formulation which can be computationally intensive for even low-order subsystems.

This section is devoted to an integration-free approach to the optimal hierarchical control approaches of Section 4.3. The approach is based on the work of Huang and Shao (1994a,b) in which, through the use of Taylor's series expansion, the first-level problem is reduced to a simple matrix algebraic one with immediate solution.

4.5.1 Problem Formulation

For the problem (4.5.1)–(4.5.3), redefine the dual function (4.3.21) as

$$q(\alpha) = \min_{x_i, u_i}\left\{L_i\left(x_i, u_i, z_i, \alpha_i\right)\right\}, \quad i = 1,2,\ldots,N \tag{4.5.4}$$

where $L_i(\cdot)$ is defined by

$$L_i\left(x_i, u_i, z_i, \alpha_i\right) = \tfrac{1}{2}\sum_{i=1}^{N}\int_0^{t_f}\left[x_i^T Q_i x_i + u_i^T R_i u_i + 2\alpha_i^T\left(z_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij} x_j\right)\right] dt \tag{4.5.5}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

with $\alpha = \left(\alpha_1^T \ldots \alpha_N^T\right)^T$ being the Lagrange multiplier vector which, as seen before, is a partition of the coordination vector. It is further noted that, as by Equation (4.3.23), we have

$$\max_{\alpha} q(\alpha) = \min_{u_i} J, \quad i = 1, \ldots, N \tag{4.5.6}$$

The advantage of the use of this strong duality is that $q(\alpha)$ is maximized in place of a decentralized minimization problem at the decomposed subsystem level.

At the lowlevel, the coordination vector components $\alpha(t)$and $z_i(t)$ are known trajectories. The problem is to minimize L_i in (4.5.5) subject to (4.5.2). Huang and Shao (1994a) have used a Taylor series expansion to convert the differential Equation (4.5.2) to an algebraic form. Let us approximate the ith subsystem's state vector $x_i(t)$ by

$$x_i(t) = \sum_{j=1}^{m} x_{ij} \phi_j(t) \tag{4.5.7}$$

where x_{ij} is the jth unknown coefficient of the basis function $\phi_j(t)$. The basis function is

$$\phi_j(t) = t^{j-1}, \quad j = 1, 2, \ldots, m \tag{4.5.8}$$

Thus, $x_i(t)$ in (4.5.7) can be written as

$$x_i(t) = X_i \Phi(t) \tag{4.5.9}$$

where

$$X_i = \left[x_{i1}, x_{12}, \ldots, x_{im}\right]; \ \Phi = \left[\phi_1, \phi_2, \ldots, \phi_m\right]^T$$

where X_i is a state parameter vector for the n_i -dimensional state variable. In a similar fashion we have

$$u_i(t) = U_i \Phi(t) = \left[u_{i1}, u_{i2}, \ldots, u_{im}\right] \Phi(t) \tag{4.5.11}$$

$$z_i(t) = Z_i \Phi(t) = \left[z_{i1}, z_{i2}, \ldots, z_{im}\right] \Phi(t) \tag{4.5.12}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where U_i and Z_i are similarly parameter vectors. Upon the substitutions of (4.5.9), (4.5.11) into (4.5.2) and integrating it from zero to t, one gets (Huang and Shao, 1994a)

$$X_i\Phi(t) - X_i(0)\Phi(t) = A_i X_i H\Phi(t) + C_i Z_i H\phi(t) \tag{4.5.13}$$

where H is the operational matrix of integration defined by the integration of shifted Chebyshev series approximation (Razzaghi and Arabshai, 1989) for a Taylor polynomial $\phi(z)$, i.e.,

$$\int_0^z \Phi(z)dz \doteq H\Phi(z) \tag{4.5.14a}$$

and

$$H = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2} & \cdots & 0 \\ & & & & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{m-1} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{4.5.14b}$$

Furthermore, $X_i(0)$ in (4.5.13) is defined as

$$X_i(0) = \left[x_i(0), 0, \ldots, 0\right] \tag{4.5.15}$$

and $x_i(0)$ is a pre-specified vector for each subsystem. In relation (4.5.13), the independence of individual column vectors $\phi_i(t)$ of $\Phi(t)$ reduces this equation to

$$X_i = A_i X_i H + B_i U_i H + C_i Z_i H + X_i(0) \tag{4.5.16}$$

which constitutes an $n_i \times m$ matrix algebraic equation. Through the use of the Kronecker product (Bellman, 1970), one can rearrange (4.5.15) as

$$\hat{X}_i = \left(H^T \otimes A_i\right)\hat{X}_i + \left(H^T \otimes B_i\right)\hat{U}_i + \left(H^T \otimes C_i\right)\hat{Z}_i + \hat{X}_i(0) \tag{4.5.17}$$

where

$$\hat{X}_i = \left[x_{i1}^T, x_{i2}^T, \ldots, x_{im}^T\right]^T \tag{4.5.18a}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\hat{U}_i = \left[u_{i1}^T, u_{i2}^T, \ldots, u_{im}^T\right]^T \tag{4.5.18b}$$

$$\hat{Z}_i = \left[z_{i1}^T, z_{i2}^T, \ldots, z_{im}^T\right]^T \tag{4.5.18c}$$

$$\hat{X}_i(0) = \left[x_i^T(0), 0, \ldots, 0\right]^T \tag{4.5.18d}$$

and denotes the Kronecker product. Equation (4.5.17) can also be used to solve for $\hat{U}_i$, i.e.,

$$\hat{U}_i = \hat{B}_i \hat{X}_i + \hat{C}_i \hat{Z}_i + \hat{D}_i \tag{4.5.19}$$

where

$$\hat{B}_i = \left(H^T \otimes B_i\right)^{-1}\left(I_i - H^T \otimes A_i\right)$$

$$\hat{C}_i = -\left(H^T \otimes B_i\right)^{-1}\left(H^T \otimes C_i\right)$$

$$\hat{D}_i = -\left(H^T \otimes B_i\right)^{-1} \hat{X}_i(0)$$

and I_i is an $mn_i \times mn_i$ identify matrix in lieu of the differential equation.

4.5.2 Performance Index Approximation

The performance index can also be approximated as follows:

$$x_i(t) = \hat{\Phi}_{n_i}^T(t)\hat{X}_i \tag{4.5.20a}$$

$$u_i(t) = \hat{\Phi}_{r_i}^T(t)\hat{U}_i \tag{4.5.20b}$$

$$z_i(t) = \hat{\Phi}_{q_i}^T(t)\hat{Z}_i \tag{4.5.20c}$$

$$d_i(t) = \hat{\Phi}_{q_i}^T(t)\hat{d}_i \tag{4.5.20d}$$

where $\hat{X}_i$, $\hat{U}_i$, and $\hat{Z}_i$ are already defined by (4.5.18), and

$$\hat{\Phi}_{n_i}(t) = \left[I_{n_i}\phi_1(t), I_{n_i}\phi_2(t), \ldots, I_{n_i}\phi_m(t)\right]^T_{n_i \times n_i m} \tag{4.5.21a}$$

$$\hat{\Phi}_{r_i}(t) = \left[I_{r_i}\phi_1(t), I_{r_i}\phi_2(t), \ldots, I_{r_i}\phi_m(t)\right]^T_{r_i \times r_i m} \tag{4.5.21b}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\hat{\Phi}_{q_i}(t) = \left[I_{q_i}\phi_1(t), I_{q_i}\phi_2(t), \ldots, I_{q_i}\phi_m(t)\right]^T_{q_i \times q_i m} \tag{4.5.21c}$$

$$\hat{\alpha}_i = \left[\alpha_{i1}^T, \alpha_{i2}^T, \ldots, \alpha_{im}^T\right]^T \tag{4.5.21d}$$

substituting (4.5.21) into the cost function (4.4.5), one gets

$$L_i = \tfrac{1}{2}\int_0^{t_f} \left[\hat{X}_i^T \hat{\Phi}_{n_i}(t) Q_i \hat{\Phi}_{n_i}^T(t)\hat{X}_i + \hat{U}_i^T \hat{\Phi}_{r_i}(t) R_i \hat{\Phi}_{r_i}^T(t)\hat{U}_i + 2(\hat{\alpha}_i^T \hat{\Phi}_{q_i}(t)\hat{\Phi}_{q_i}(t)\hat{Z}_i \right.$$
$$\left. -\sum_j \hat{\alpha}_j^T \hat{\Phi}_{q_i}(t) L_{ji} \hat{\Phi}_{n_i}^T(t)\hat{X}_i) \right] dt \tag{4.5.22}$$

Now, one can use a property of $p \times \; n$ matrix A, and then an n-degree extended Taylor series matrix $\Phi_n^T(z)$ can be expressed by

$$A\hat{\Phi}_n^T(z) = \hat{\Phi}_p^T(z)(I_m \otimes A) \tag{4.5.23}$$

where I_m is an $m \times \; m$ identity matrix. The property (4.5.23) can be applied to (4.5.22) to put it in a more compact form as,

$$L_i = \tfrac{1}{2}\left[\hat{X}_i^T \hat{Q}_i \hat{X}_i + \hat{U}_i^T \hat{R}_i \hat{U}_i + 2\left(\hat{\alpha}_i^T \hat{W}_{q_i} \hat{Z}_i - \sum_j \hat{\alpha}_j^T \hat{W}_{q_j} \hat{L}_{ji} \hat{X}_i \right)\right] \tag{4.5.24}$$

where

$$\hat{Q}_i = \hat{W}_{n_i}(I_m \otimes Q_i),$$
$$\hat{R}_i = \hat{W}_{r_i}(I_m \otimes R_i),$$
$$\hat{L}_{ji} = (I_m \otimes L_{ji}),$$

and $\hat{W}_{n_i}, \hat{W}_{r_i}, \hat{W}_{q_i}$, and $\hat{W}_{q_j}$ are given by

$$\hat{W}_n = \int_0^B \hat{\Phi}_n(z)\hat{\Phi}_n^T(z)dz = w \otimes I_n \tag{4.5.25}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

4.5.3 Optimal Control

With the above development and transformations accomplished, one can now redefine the optimal control problem at the subsystem level as min L_i in (4.5.24) subject to (4.5.19).

Now by using (4.5.19), the Lagrangian L_i in (4.5.24) becomes

$$L_i = \tfrac{1}{2}\Big[\hat{X}_i^T \hat{Q}_i \hat{X}_i + \left(\hat{B}_i \hat{X}_i + \hat{C}_i \hat{Z}_i + \hat{D}_i\right)^T \hat{R}_i \left(\hat{B}_i \hat{X}_i + \hat{C}_i \hat{Z}_i + \hat{D}_i\right)$$
$$+2\hat{\alpha}_i^T \hat{W}_{q_i} \hat{Z}_i - 2\sum_j \hat{\alpha}_j^T \hat{W}_{qj} \hat{L}_{ji} \hat{X}_i \Big] \tag{4.5.26}$$

Therefore, the lower-level problem is one of minimizing L_i to find X_i with z_i* and d_i* being known vectors. From the necessary conditions of optimality one gets

$$\frac{\partial L_i}{\partial X_i} = 0 .$$

Hence,

$$\left(\hat{Q}_i + \hat{B}_i^T \hat{R}_i \hat{B}_i\right) X_i + \hat{B}_i^T \hat{R}_i \hat{C}_i \hat{Z}_i^* + B_i^T \hat{R}_i \hat{D}_i - \sum_j \hat{\alpha}_j^{*T} \hat{W}_{qj} \hat{L}_{ji} = 0$$

From this expression, the optimal state is given by

$$\hat{X}_i = -\left(Q_i + B_i^T R_i B_i\right)^{-1} \left(B_i^T R_i C_i Z_i^* + \hat{B}_i^T \hat{R}_i \hat{D}_i - \sum_j \hat{\alpha}_j^{*T} \hat{W}_{q_j} \hat{L}_{ji} \right) \tag{4.5.27}$$

Substituting 4.5.27 into 4.5.19, the optimal control is obtained

$$U_i = B_i \left(Q_i + B_i^T R_i B_i\right)^{-1} \left(B_i^T R_i C_i Z_i^* + B_i^T R_i D_i - \sum_j \alpha_j^{*T} W_{q_j} L_{ji} \right)$$
$$+ C_i Z_i^* + D_i \tag{4.5.28}$$

An initial comparison of this approach to the standard interaction prediction indicates that this approach eliminates the need to solve a TPBV

controlengineers.ir

controlengineers.ir
@controlengineers

(two-point boundary value) problem and its associated problems such as the Ricatti or any other differential equations.

4.5.4 Coordinator Problem

As before, the problem at this level is for the coordinator to improve on the values of α_i and z_i, i.e., the coordination vector. In accordance with the interaction prediction (Jamshidi, 1983) and Equation (4.3.56), a necessary condition for maximizing $\phi(\alpha)$ is given

$$\frac{\partial L}{\partial \bar{Z}_i} = 0, \quad \frac{\partial L}{\partial \bar{\lambda}_i} = 0,$$

where L_i is defined in (4.5.5). The new prediction for Z_i and α_i is given by

$$\begin{pmatrix} \hat{Z}_i \\ \hat{\alpha}_i^T \overline{W}_{q_i} \end{pmatrix}^{k+1} = \begin{pmatrix} \sum\limits_{j=1} \hat{L}_{ij} \hat{X}_j \\ -\hat{C}_i^T \hat{R}_i \hat{D}_i - \hat{C}_i^T \hat{R}_i \hat{B}_i \hat{X}_i - \hat{C}_i^T \hat{R}_i \hat{C}_i \sum\limits_j \hat{L}_{ij} \hat{X}_j \end{pmatrix}^k \tag{4.5.29}$$

where X_i is the value of 4.5.27. The interaction error is normalized from (Huang and Shao, 1994a)

$$Error = \sum_{i=1}^{n} \left(\hat{Z}_i - \sum_j \hat{L}_{ij} \hat{X}_j \right)^T \hat{W}_{q_i} \left(\hat{Z}_i - \sum_{j=1} \hat{L}_{ij} \hat{X}_j \right) \tag{4.5.30}$$

when the total interaction error is small enough, the optimal trajectories are

$$x_i^*(t) = \overline{\Phi}_{n_i}^T(t) \overline{X}_i^*, \tag{4.5.31}$$

$$u_i^*(t) = \overline{\Phi}_{m_1}^T(t) \overline{U}_i^*, \tag{4.5.32}$$

where X_i^* and U_i^* have been given (4.5.27) and (4.5.28).

The above development on integration-free hierarchical control approach is summarized by the following algorithm.

Algorithm 4.3. Series Expansion Hierarchical Control

Step 1: Using intitial values of $\alpha_i(t)$ and $z_i(t)$, compute n-independent Equations (4.5.19) and store x_i, $i = 1, 2, \ldots, n$.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Step 2: Using the results of Step 1, at the second level use (4.5.29) to update the coordination vector.

Step 3: Evaluate the interaction error (4.5.31) and check for convergence. If converged, stop. Otherwise, go to Step 2.

Example 4.5.1. Consider the system of Example 4.3.3 which was solved using the standard interaction prediction whose convergence was shown in Figure 4.11. It is desired to resolve this system using Algorithm 4.3 and compare the two approaches.

SOLUTION: Use the identical initial values for $\alpha_i(t)$ and $x(0)$ matrices R and Q and final time $t_f = 1$. The system is fourth-order, consisting of two second-order subsystems. An $m = 5$ term Taylor series is used to solve the hierarchical control problem. The algorithm (Huang and Shao, 1994a) has been programmed using an IBM PC 486. As the first step, two independent algebraic equations are calculated by using (4.5.27) and (4.5.28) and initial values

$$\alpha_1(t) = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}; \qquad \alpha_2(t) = \begin{pmatrix} 0.75 \\ 0.75 \end{pmatrix}$$

At the second level, the interaction vectors were predicted using the recursive interaction prediction relations (4.5.29). The interaction error was reduced to 2.389×10^{-6} in 6 iterations, as shown in Figure 4.16.

This convergence is 32 percent better than the standard interaction prediction method whose error reduced to 3.511×10^{-6} for the same number of iterations—6.

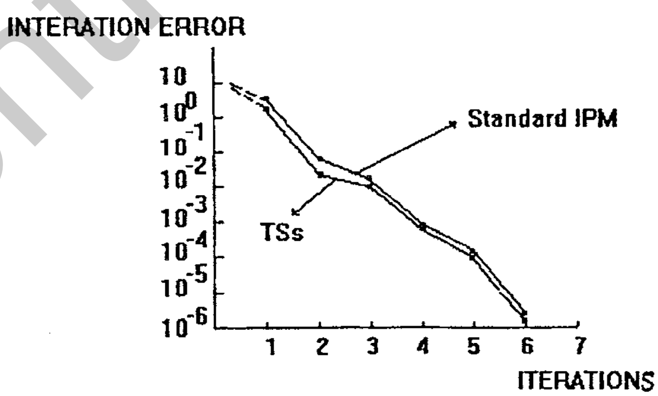


Figure 4.16 Interaction error versus iteration for Example 4.5.1 using standard interaction prediction and Taylor series methods.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

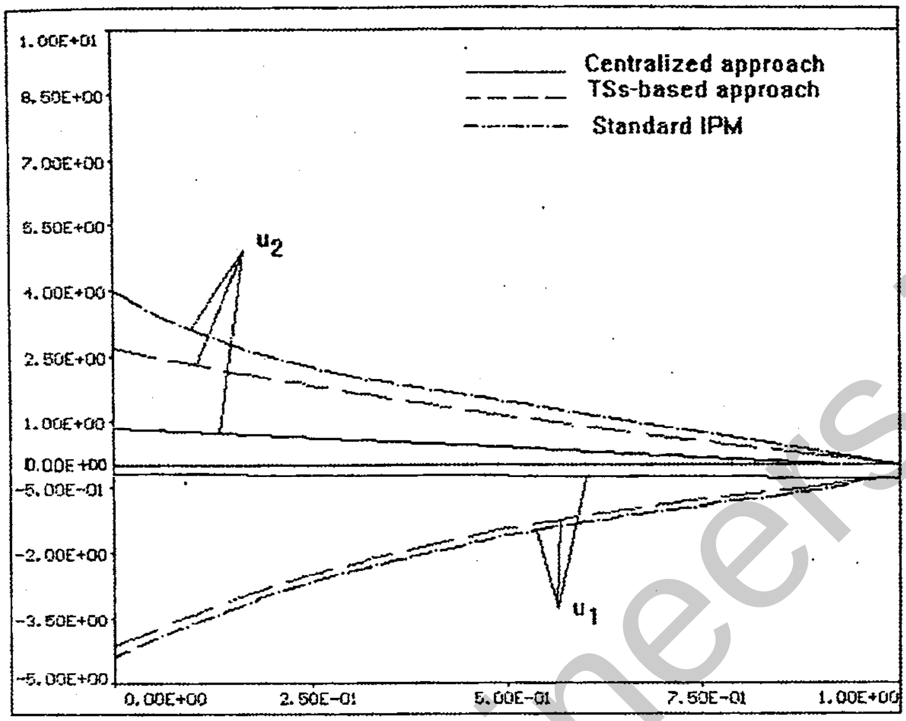


Figure 4.17 Optimal hierarchical control trajectories of Example 4.5.1 using three methods.

The optimum control signals are obtained:

$$\begin{pmatrix} u_1^*(t) \\ u_2^*(t) \end{pmatrix} = \begin{pmatrix} -4.151 & 8.052 & -6.721 & 2.910 & -0.394 \\ 2.708 & -3.852 & 1.658 & -0.593 & 0.097 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \\ t^4 \end{pmatrix}$$

Next, for the sake of comparison, the original system is optimized by using the stardard IPM method and the exact centralized solving method. These results are shown in Figure 4.17. Note that the TSs-based approach has better results than the standard IPM.

Huang and Shao (1994a) have noted that increasing the number of terms of the Taylor series, i.e., m beyond a certain value, say $m = 10$, may not be worth the resulting computational burdon. A value of $5 \le m \le 10$ may be appropriate for most prescribed convergence levels. The Taylor series scheme does have an attractive feature and that is the elimination of any TPBV problem at the first level. In a way, the present method has retained the computationally simple feature of the interaction prediction at the sec-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ond level and improved the overall problem by simplifying the first-level problem as well.

4.6 Computer-Aided Hierarchical Control Design Examples

One of the most relevant examples of large-scale systems is a power system. In this section two CAD examples for a ninth-order and a twentieth-order power system will be designed using interaction prdiction method using LSSPAK.

CAD Example 4.6.1. Consider a ninth-order system descending a mathematical model of a power system

$$\dot{x} = \begin{bmatrix} 0 & 0.545 & 0 & 0 & 0 & -0.545 & 0 & 0 & 0 \\ -6 & -0.05 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.33 & 3.33 & 0 & 0 & 0 & 0 & 0 \\ 0 & -5.21 & 0 & -12.5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.425 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & -0.05 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.33 & 3.33 & 0 \\ 0 & 0 & 0 & 0 & 0 & -5.21 & 0 & -12.5 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0.425 & 0 & 0 & 0 \end{bmatrix} x$$

$$+ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 12.5 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 12.5 \\ 0 & 0 \end{bmatrix} u, \quad J = \frac{1}{2}\int_0^3 \left(x^T Q x + u^T R u\right) dt$$

where $Q = I_q$, $R = I_2$, $\Delta t = 0.05$.

$$x(0) = \begin{bmatrix} 10 & -110 & -11 & 10 \end{bmatrix}^T \text{ and } \alpha(0) = \begin{bmatrix} 0.1 & 0.1 & 0.02 & -0.2 & 0.10 & 0.1 & 0.1 \end{bmatrix}^T$$

SOLUTION: The solution of this problem on LSSPAK is shown in Figures 4.18–4.20.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

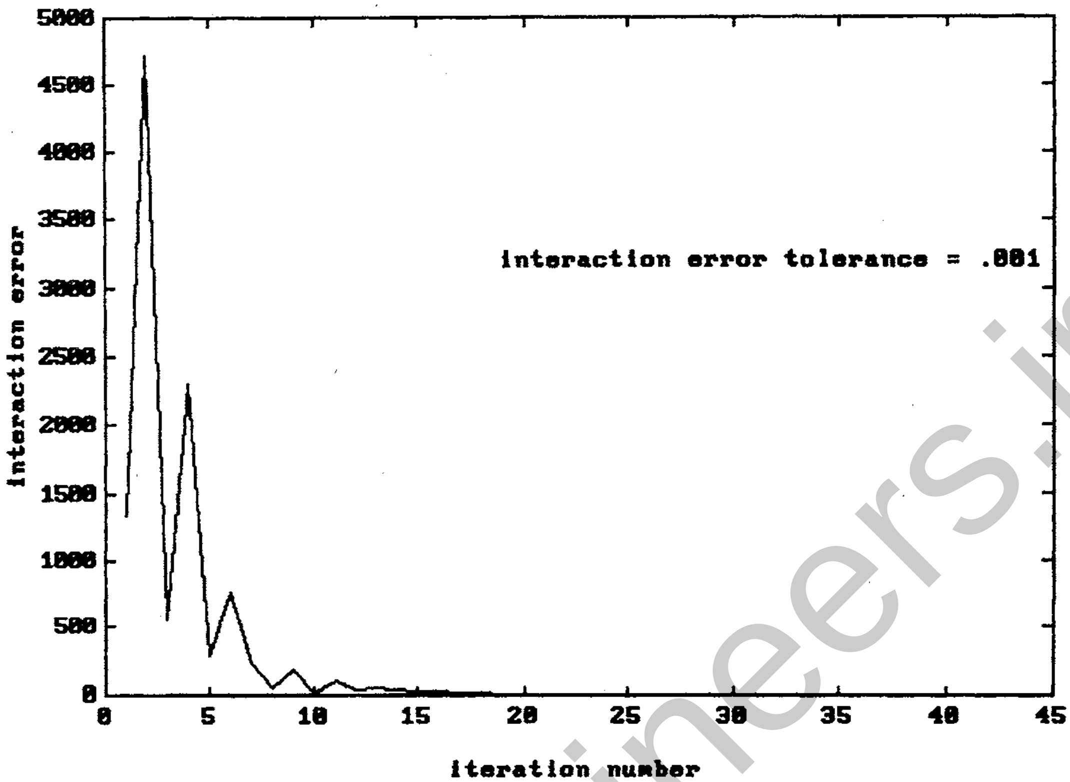


Figure 4.18 Interaction error vs. iterations for CAD Example 4.6.1.

Partial Listing of LSSPAK using "RICRKUT" and "INTPRD"

<<RICRKUT>> Solves Differential Matrix Riccati Equation via Runge-Rutta integration including a polynomial approximation for the Riccati matrix.

For an equation of the form:

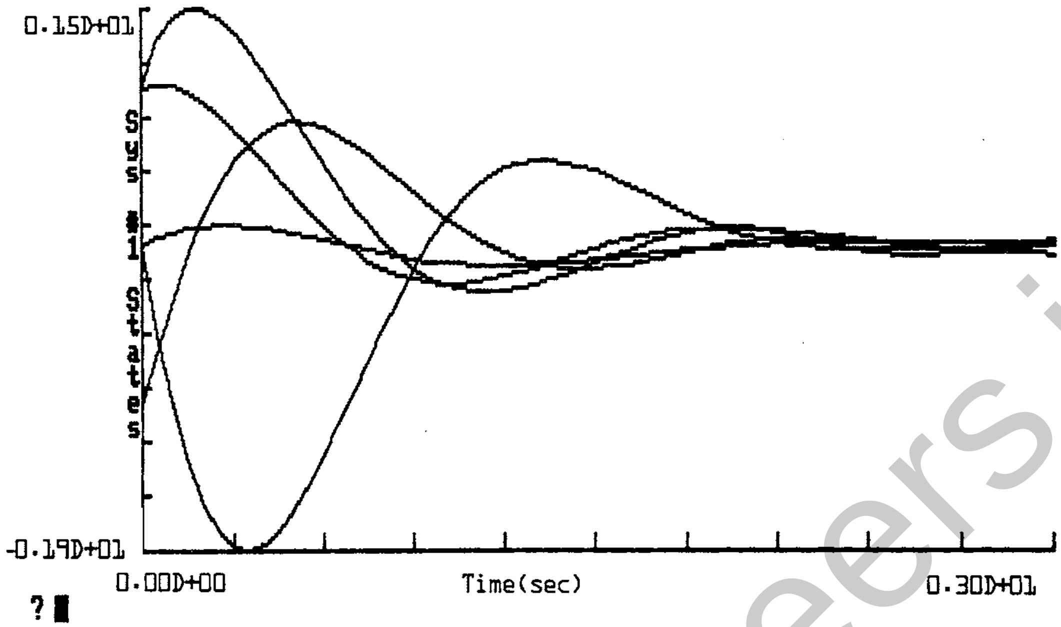
dR(t)/dt=A'K+KA-KBR^-1B'K+Q, with Final Condition K(tf)

Order of Matrix A - n = 5
Order of Matrix B - m = 2
Initial Time to = 0
Final Time tf = 3
Step Size dt = .05

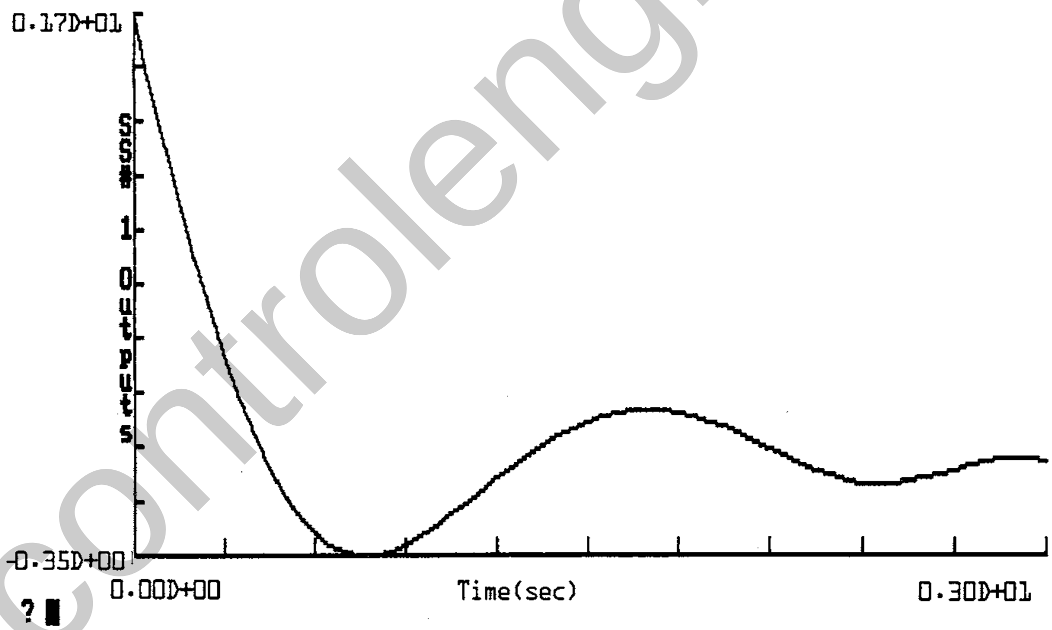
Matrix A

0.000E+00	0.545E+00	0.000E+00	0.000E+00	0.000E+00
–.600E+01	–.500E–01	0.600E+01	0.000E+00	0.000E+00
0.000E+00	0.000E+00	–.333E+01	0.333E+01	0.000E+00
0.000E+00	–.521E+01	0.000E+00	–.125E+02	0.000E+00
0.100E+01	0.425E+00	0.000E+00	0.000E+00	0.000E+00

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) States.

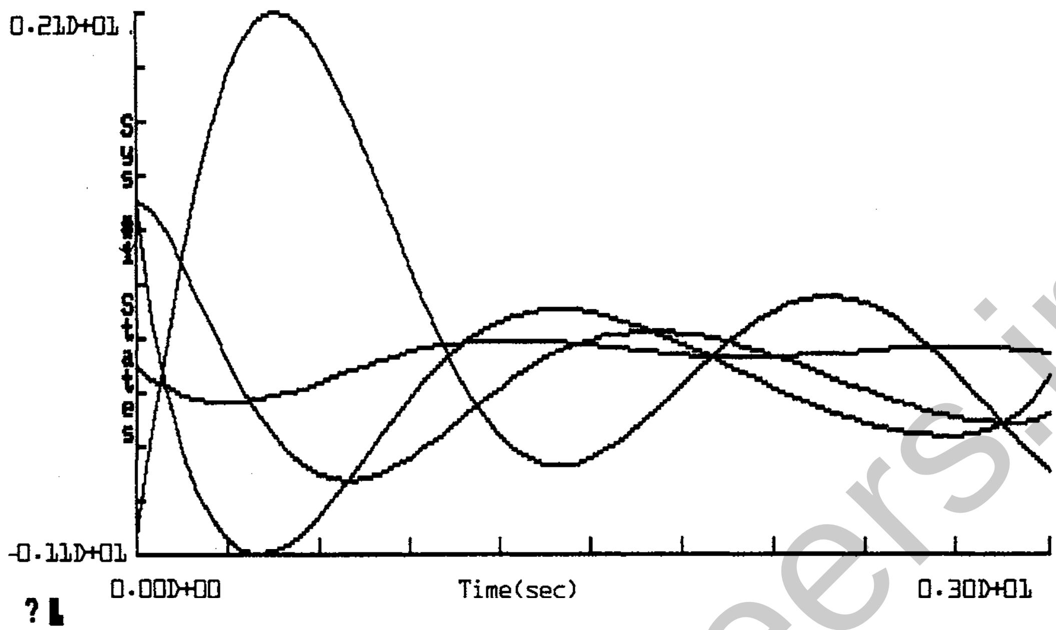


(b) Control input signal.

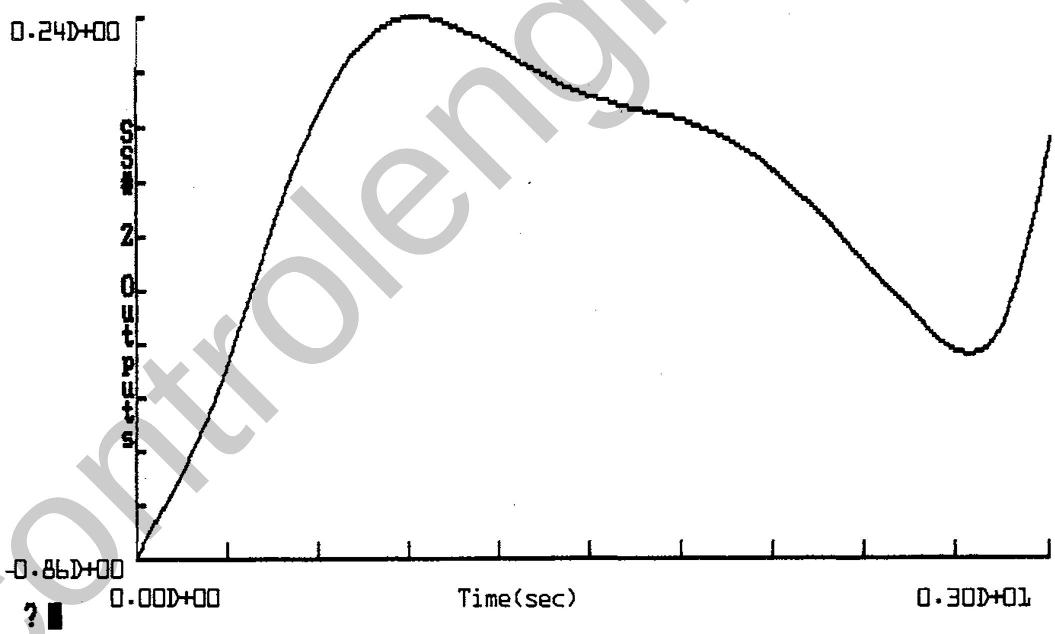
Figure 4.19 Optimal time responses for subsystem No. 1 of CAD Example 4.6.1.

controlengineers.ir

controlengineers.ir
@controlengineers



(a) States.



(b) Control input signal.

Figure 4.20 Optimal time responses for subsystem No. 2 of CAD Example 4.6.1.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Matrix B

0.000E+00	0.000E+00
0.000E+00	0.000E+00
0.000E+00	0.000E+00
0.125E+02	0.000E+00
0.000E+00	0.000E+00

Matrix Q-Diag
0.100E+01
0.100E+01
0.100E+01
0.100E+01
0.100E+01

Matrix R

0.100E+01	0.000E+00
0.000E+00	0.100E+01

K(tf):
1 0.000E+00
2 0.000E+00
•
•
•
15 0.000E+00

At Time t = 0 Riccati Matrix K(t):

3.719E+00	1.093E−01	−2.710E−01	−5.967E−02	4.435E−01
1.093E−01	5.991E−01	5.125E−01	6.796E−02	4.539E−01
−2.710E−01	5.125E−01	7.404E−01	1.186E−01	4.240E−01
−5.967E−02	6.796E−02	1.186E−01	5.364E−02	6.718E−02
4.435E−01	4.539E−01	4.240E−01	6.718E−02	2.1398+00

At Time t = .05 Riccati Matrix K(t):
•
•
•

<<RICRRUT>> Solves Differential Matrix Riccati Equation via Runge-Rutta integration including a polynomial approximation for the Riccati matrix.

For an equation of the form:

dK(t)/dt~A'K+KA-KBR^-lB'K+Q, with Final Condition K(tf)

Order of Matrix A - n = 4
Order of Matrix B - m = 2
Initial Time to = 0
Final Time tf = 3
Step Size dt = .05

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Matrix A

−.500E−01	0.600E+01	0.000E+00	0.000E+00
0.000E+00	−.333E+01	0.333E+01	0.000E+00
−.521E+01	0.000B+00	−.125E+02	0.000E+00
0.425E+00	0.000E+00	0.000E+00	0.000E+00

Matrix B

0.000E+00	0.000E+00
0.000B+00	0.000E+00
0.000E+00	0.125E+02
0.000E+00	0.000E+00

Matrix Q-Diag
0.100E+01
0.100E+01
0.100E+01
0.100E+01

Matrix R
0.100E+01
0.000E+00
0.000E+00
0.100E+01

K(tf):
1 0.000E+00
2 0.000E+00
•
•
•
10 0.000E+00

At Time t = 0 Riccati Matrix K(t):

5.200E-01	4.676E-01	6.300E-02	4.038E-01
4.676E-01	6.954E-01	1.120E-01	3.827E-01
6.300E-02	1.120E-01	5.257E-02	6.094E-02
4.038E-01	3.827E-01	6.094E-02	2.392E+00

At Time t = .05 Riccati Matrix K(t):
•
•
•

At Time t = .45 Riccati Matrix K(t):

5.124E-01	4.598E-01	6.172E-02	3.631E-01
4.598E-01	6.875E-01	1.107E-01	3.413E-01
6.172E-02	1.107E-01	5.236E-02	5.419E-02
3.631E-01	3.413E-01	5.419B-02	2.176E+00

•
•
•

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers


```
********************************************************************************
  INTRPRD solves a multi subsystem hierarchical control system using the method
  of interaction prediction. The algorthim may be found in LARGE SCALE SYS-
  TEMS MODELING AND CONTROL  M. Jamshidi, Elsevier North Holland,N.Y.,
  1983 pp 119-123.
********************************************************************************
  Instructions for plotting with interaction prediction program:
  When you get a plot on the screen, hit return to return to the menu.
  If you plan to dump plots to the printer,
  you must run the DOS file GRAPHICS prior to running this program.
  Then when you wish to dump a plot hit shift-PrtSc.

  Optimization via the interaction prediction method.
  Initial time (to):        0
  Final time (tf):          3
  Step size (Dt):           .05
  Total no. of 2nd level iterations =  50
  Error tolerance for multi-level iterations  =  .001
  Order of overall large scale system=      9
  Order of overall control vector (R)=      2
  Number of subsystems in large scale system= 2

  Matrix Subsystem state orders--n sub i
  0.500D+01
  0.400D+01

  Matrix Subsystem input orders--r sub i
  0.100D+01
  0. 100D+01

  Discrete values of Ricatti matrix to be used.
  Subsystem no. 1 at 2nd level iteration no. 1
  Subsystem no. 2 at 2nd level iteration no. 1
  At second level iteration no.  1 interaction error = 0.134D+04
  Subsystem no. 1 at 2nd level iteration no. 2
  Subsystem no. 2 at 2nd level iteration no. 2
  At second level iteration no.  2 interaction error = 0.471D+04
  Subsystem no. 1 at 2nd level iteration no. 3
  Subsystem no. 2 at 2nd level iteration no. 3
  •
  •
  •
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

At second level iteration no. 39 interaction error = 0.232D-02
Subsystem no. 1 at 2nd level iteration no. 40
Subsystem no. 2 at 2nd level iteration no. 40
At second level iteration no. 40 interaction error = 0.285D-02
Subsystem no. 1 at 2nd level iteration no. 41
Subsystem no. 2 at 2nd level iteration no. 41
At second level iteration no. 41 interaction error = 0.945D-03

CAD Example 4.6.2. In this example, a twentieth-order model of a power system whose model is given by $\dot{x} = Ax + Bu$ with dim $(A) = 20 \times 20$, dim $(B) = 20 \times 20$, $R = I_{20}$, $Q = I_{20}$, $\alpha(0) = (0.5 \;\; 0.5 \;\; \ldots \;\; 0.5)^T$. Due to lack of space, matrices A and B are not printed here. Interested readers can get them from the author or the Solutions Manual.

SOLUTION: The system was decomposed into five fourth-order subsystems. Five differential matrix Riccati equations were solved on RICRKUT for $0 \le t \le 4$ at $\Delta t = 0.1$. A partial listing of INTPRD is shown below.

```
******************************************************************************
INTRPRD solves a multi subsystem hierarchical control system using the method
of interaction prediction. The algorthim  may be found in LARGE SCALE SYS-
TEMS MODELING AND CONTROL  M. Jamshidi, Elsevier North Holland,N.Y.,
1983 pp 119-123.
******************************************************************************
Instructions for plotting with interaction prediction program:

When you get a plot on the screen, hit return to return to the menu.

If you plan to dump plots to the printer,
you must run the DOS file GRAPHICS prior to running this program.
Then when you wish to dump a plot hit shift-PrtSc.

Optimization via the interaction prediction method.

Initial time (to):        0
Final time (tf):          4
Step size (Dt):           .1
Total no. of 2nd level iterations =  31
Error tolerance for multi-level iterations  =  .001
Order of overall large scale system=       20
Order of overall control vector (R)=       20
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Number of subsystems in large scale system =5

•
•
•

Subsystem no. 1 at 2nd level iteration no. 25
Subsystem no. 2 at 2nd level iteration no. 25
Subsystem no. 3 at 2nd level iteration no. 25
Subsystem no. 4 at 2nd level iteration no. 25
Subsystem no. 5 at 2nd level iteration no. 25
At second level iteration no. 25 interaction error = 0.539D-02

•
•
•

Subsystem no. 1 at 2nd level iteration no. 1
Subsystem no. 2 at 2nd level iteration no. 1
Subsystem no. 3 at 2nd level iteration no. 1
Subsystem no. 4 at 2nd level iteration no. 1
Subsystem no. 5 at 2nd level iteration no. 1
At second level iteration no. 1 interaction error = 0.127D+05

Figure 4.21 shows the interaction prediction error versus iteration for the twentieth-order power system optimal hierarchical control problem.

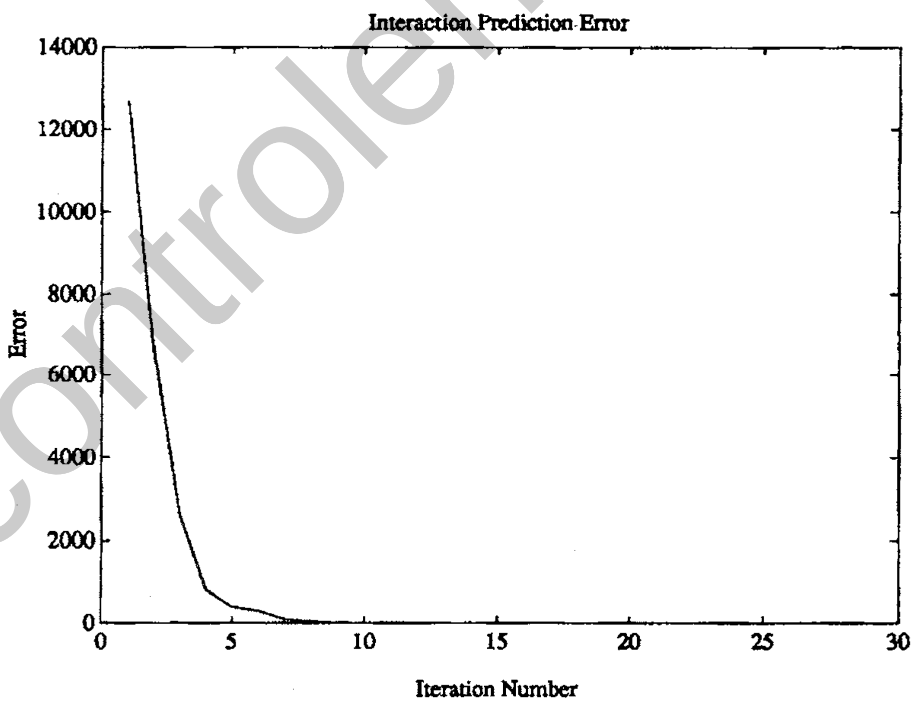


Figure 4.21 Interaction error performance for CAD Example 4.6.2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Problems

4.1. CADLAB

Consider a two-subsystem problem,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \left[\begin{array}{cc|cc} -1.5 & 0 & 0 & 0.1 \\ -0.5 & -1 & 0 & 0.2 \\ \hline 0.5 & 0 & -1.5 & 0 \\ 0 & 0.5 & -0.5 & -1 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \left[\begin{array}{c|c} 0.1 & 0 \\ 0.1 & 0 \\ \hline 0 & 0.1 \\ 0 & 0.1 \end{array}\right] \left[\begin{array}{c} u_1 \\ \hline u_2 \end{array}\right]$$

Use the two-level goal coordination Algorithm 4.1 to find an optimum control which minimizes

$$J = \int_0^5 \left(x^T Q x + u^T R u\right) dt$$

$Q = \text{diag}(1,2,2,1)$, $R = I_2$, and $\Delta t = 0.1$. Use your favorite computer language and an integration routine such as Runge-Kutta to solve the associated differential matrix Riccati equation and the state equation.

4.2. CADLAB

Consider an interconnected system

$$\dot{x} = \left[\begin{array}{cc|ccc} -1 & 0.5 & 0.1 & 0.5 & 0 \\ 0.1 & -2 & -0.5 & 0.2 & -0.1 \\ \hline 0.2 & 0 & -5 & 0.5 & -1 \\ 0.1 & -0.2 & 0 & -2 & 0 \\ 0.4 & 0.1 & -0.5 & 0 & -4 \end{array}\right] x + \left[\begin{array}{c|c} 1 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{array}\right] u$$

with cost function

$$J = \sum_{i=1}^{2} \tfrac{1}{2}\left[x_i^T(2) Q_i x_i(2) + \int_0^2 \left(x_i^T(t) Q_i x_i(t) + u_i^T(t) R_i u(t)\right) dt \right]$$

with $Q_i = \text{diag}(2, 1,1,1,1)$, $R_i = I_2$, and initial conditions $x^T(0) = [\,1 \;\; 0 \;\; 0.5 - 1 \;\; 0\,]$, $\Delta t = 0.1$. Use the interaction prediction Algorithm 4.2 to find the optimal control $u^*(t)$ for the above problem. Use LSSPAK or your favorite software.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

4.3. CADLAB Repeat problem 4.2 using the Taylor series expansion approach of Section 4.5.

4.4. CADLAB For a system with the matrices

$$A = \begin{bmatrix} -5 & 0.2 & 0.5 & 0.1 & 0 \\ 0 & -2 & 0.2 & 0 & 0 \\ 0.17 & 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & -0.5 & 0 \\ -1 & 0 & -0.5 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

with $Q =$ diag(1, 1, 2, 2, 2) and $R = I_2$, find a hierarchical control law based on the Taylor series expansion method of Algorithm 4.3. Use initial state $x(0) = [\ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ ]^T$.

4.5. Repeat Example 4.3.4.

4.6. Repeat Problem 4.5 using interaction prediction approach.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 5

Decentralized Control of Large-Scale Systems

controlengineers.ir

5.1 Introduction

In Chapter 1, the notion of decentralized control and its structure were briefly introduced. As it was mentioned then and is emphasized again here, the main motivation behind decentralized control is the failure of conventional methods of centralized control theory. Some fundamental techniques such as pole placement, state feedback, optimal control, state estimation, and like of the latter theory require complete information from all system sensors for the sake of feedback control. This scheme is clearly inadequate for feedback control of large-scale systems. Due to the physical configuration and high dimensionality of such systems, a centralized control is neither economically feasible nor even necessary. Therefore, in many applications of feedback control theory to linear large-scale systems some degree of restriction is assumed to prevail on the transfer of information. In some cases a total decentralization is assumed; i.e., every local control u_i is obtained from the local output y_i and possible external input v_i (Sandell *et al.*, 1978). In others, an intermediate restriction on the information is possible.

In this chapter, three major problems related to decentralized structure of large-scale systems are addressed. The first is decentralized stabilization. The problem of finding a state or an output feedback gain whereby the closed-loop system has all its poles on the left half-plane is commonly known as feedback "stabilization." Alternatively, the closed-loop poles of a controllable system may be preassigned through the state or output feedback. Clearly, the applications of these concepts in a decentralized fashion require certain extensions which are discussed in Section 5.3.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The second problem addressed in this chapter is the decentralized "robust" control of large-scale linear systems with or without a known plant. This problem, first introduced by Davison (1976a,b,c,d) and known as "general servomechanism," takes advantage of the tuning regulators and dynamic compensators to design a feedback which both stabilizes and regulates the system in a decentralized mode. The notion of "robust" feedback control will be defined and discussed in detail later; however, for the time being a control is said to be robust if it continues to provide asymptotic stability and regulation under perturbation of plant parameters and matrices.

The third problem is stochastic decentralized control of continuous- and discrete-time systems. The scheme discussed here is based on the assumption of one sensor for each controller (channel or node) whose information, processed with a local Kalman estimator, is shared with all other controllers.

In Section 5.2, the problem of decentralized stabilization is mathematically formulated. The section reviews some of the appropriate schemes for decentralized feedback stabilization, including the notions of "fixed modes," "fixed polynomials" (Wang and Davison, 1973), and their role in dynamic compensation. Also discussed are the dynamic stabilization of large-scale systems and the notion of "exponential stability" applied to decentralized systems. The stabilization of linear time-varying systems (Ikeda and Šiljak, 1980a,b) and the special case of time-invariant system stabilization will follow.

In Section 5.3, decentralized adaptive control is reviewed and a regulation problem is formulated. A theorem for the convergence of the decentralized adaptive iterations follows. Decentralized tracking and its design process are given next. A liquid-metal reactor (LMR) is used for decentralized control. Model-reference adaptive control (MRAC) is then defined, and the LMR is repeated for a design based on MRAC.

Discussion on the three main problems considered in this chapter and their potential use in real large-scale systems are presented in Section 5.4.

5.2 Decentralized Stabilization

In this section the problem of stabilizing large-scale linear time-invariant (TIV) and time-varying systems is presented. Conditions under which the overall system with decentralized control can be stabilized will be given. Decentralized stabilization has been an active field of research for large-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

scale systems. The discussions on this topic are restricted to linear time-invariant systems for the most part and are based on the works of Wang and Davison (1973), Davison (1976a,b, 1978, 1979), Šiljak (1978a,b, 1991), Corfmat and Morse (1976b), Saeks (1979), and Huang and Sundareshan (1980). Additional works on such subjects as large-scale linear systems with nonlinear interconnections (Sundareshan, 1977a) and the results of Ikeda and Šiljak (1980a) for the time-varying systems and the case of time-invariant systems (Anderson and Moore, 1981) will also be considered.

Consider a large-scale linear TIV (time-invariant) system with N local control stations (channels),

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^{N} B_i u_i(t) \tag{5.2.1}$$

$$y_i(t) = C_i x, \quad i = 1, 2, \ldots, N \tag{5.2.2}$$

where x is an $n \times 1$ state vector, and u_i and y_i are $m_i \times 1$ and $r_i \times 1$ control and output vectors associated with the ith control station, respectively. The original system control and output orders m and r are given by

$$m = \sum_{i=1}^{N} m_i, \qquad r = \sum_{i=1}^{N} r_i \tag{5.2.3}$$

The decentralized stabilization problem is defined as follows: Obtain N local output control laws, each with its independent dynamic compensator,

$$u_i(t) = H_i z_i(t) + K_i y_i(t) + L_i \upsilon_i(t) \tag{5.2.4}$$

$$\dot{z}_i(t) = F_i z_i(t) + S_i y_i(t) + G_i \upsilon_i(t) \tag{5.2.5}$$

so that the system (5.2.1)–(5.2.2) in its closed-loop form is stabilized. In (5.2.4)–(5.2.5) $z_i(t)$ is the $n_i \times 1$ output vector of the ith compensator, $\upsilon_i(t)$ is the $m_i \times 1$ external input vector for the ith controller, and matrices H_i, K_i, L_i, F_i, S_i, and G_i are $m_i \times n_i$, $m_i \times r_i$, $m_i \times m_i$, $n_i \times n_i$, $n_i \times r_i$, and $n_i, \times m_i$ respectively. Alternatively, the problem can be restated as follows: Find matrices H_i, K_i, L_i, F_i, S_i, and G_i, $i = I, 2, \ldots, N$ so that the resulting closed-loop system described by (5.2.1)–(5.2.2) has its poles in a set £, where £ is a nonempty symmetric open subset of complex s-plane (Davison, 1976a). It is clear that the membership of a closed-loop system pole $\lambda \in$ £ implies its complex conjugate $\lambda^* \in L$ in a prescribed manner.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

5.2.1 Fixed Polynomials and Fixed Modes

The notions of fixed polynomials and fixed modes are generalizations of the "centralized" systems' pole placement problem, in which any uncontrollable and unobservable mode of the system must be stable (Brasch and Pearson, 1970) to the decentralized case. The idea of fixed modes for decentralized control was introduced by Wang and Davison (1973) which leads to necessary and sufficient conditions for stabilizability of decentralized systems (5.2.1)–(5.2.2). A more refined characterization of fixed modes and the associated decentralized control were presented later on by Corfmat and Morse (1976a), Anderson and Clements (1981), Anderson (1982), Vidyasager and Viswanadham (1982), Davison and Özgüner (1983), and Tarokh and Jamshidi (1987). Some of these approaches have led to conditions which involve difficult rank and eigenvalue computations (Šiljak, 1991).

Consider the decentralized stabilization problem described by (5.2.1)–(5.2.5). The dynamic compensator (5.2.4)–(5.2.5) may be rewritten in compact form,

$$u(t) = Hz(t) + Ky(t) + Lv(t) \tag{5.2.6}$$

$$\dot{z}(t) = Fz(t) + Sy(t) + Gv(t) \tag{5.2.7}$$

where

$$\begin{aligned}
&H \triangleq \text{Block} - \text{diag}\{H_1, H_2, \ldots, H_N\}, \quad K \triangleq \text{Block} - \text{diag}\{K_1, K_2, \ldots, K_N\}\\
&L \triangleq \text{Block} - \text{diag}\{L_1, L_2, \ldots, L_N\}, \quad F \triangleq \text{Block} - \text{diag}\{F_1, F_2, \ldots, F_N\}\\
&S \triangleq \text{Block} - \text{diag}\{S_1, S_2, \ldots, S_N\}, \quad G \triangleq \text{Block} - \text{diag}\{G_1, G_2, \ldots, G_N\}
\end{aligned} \tag{5.2.8}$$

and

$$\begin{aligned}
&u^T(t) \triangleq \{u_1^T(t) \vdots \cdots \vdots u_N^T(t)\}, \quad z^T(t) \triangleq \{z_1^T(t) \vdots \cdots \vdots z_N^T(t)\}\\
&y^T(t) = \{y_1^T(t) \vdots \cdots \vdots y_N^T(t)\}, \quad v^T(t) = \{v_1^T(t) \vdots \cdots \vdots v_N^T(t)\}
\end{aligned}$$

If the controller (5.2.6)–(5.2.7) is applied to (5.2.1)–(5.2.2), the following augmented closed-loop system results:

$$\begin{bmatrix} \dot{x}(t) \\ \hline \dot{z}(t) \end{bmatrix} = \left[\begin{array}{c|c} A + BKC & BH \\ \hline SC & F \end{array}\right] \begin{bmatrix} x(t) \\ \hline z(t) \end{bmatrix} + \begin{bmatrix} BL \\ \hline G \end{bmatrix} v(t) \tag{5.2.9}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where

$$B = [\underbrace{B_1}_{m_1} \vdots \cdots \vdots \underbrace{B_N}_{m_N}] \; and \; C = \begin{bmatrix} C_1 \\ \cdots \\ \vdots \\ \cdots \\ C_N \end{bmatrix} \begin{matrix} \}r_1 \\ \\ \\ \\ \}r_N \end{matrix} \tag{5.2.10}$$

As mentioned earlier, the problem is to find the control laws (5.2.6)–(5.2.7) so that the overall augmented system (5.2.9) is asymptotically stable. In other words, by way of local output feedback, the closed-loop poles of the decentralized system are required to lie on the left half of the complex s-plane. The following definitions and theorem provide the ground rules for this problem.

Definition 5.1. Consider the system (C, A, B) describing (5.2.1)–(5.2.2) and integers m_i, r_i, $i = 1, 2, \dots, N$, in (5.2.3). Let the $m \times r$ gain in matrix K be a member of the following set of block-diagonal matrices:

$$\mathbf{K} = \left(K \middle| K = \begin{array}{c} m_1\{ \\ \\ \\ \\ \end{array} \begin{bmatrix} \overbrace{K_1}^{r_1} & & & \\ & K_2 & & \\ & & \ddots & \\ & & & \underbrace{K_N}_{r_N} \end{bmatrix} \begin{array}{c} \\ \\ \\ \}m_N \end{array} \right) \tag{5.2.11a}$$

where dim $(K_i) = m_i \times r_i$, $i = 1, 2, \dots, N$. Then the "fixed polynomial" of (C, A, B) with respect to $\mathbf{K}$ is the greatest common divisor (gcd) of the set of polynomials $|\lambda I - A - BKC|$ for all $K \in \mathbf{K}$ and is denoted by

$$\phi(\lambda; C, A, B, \mathbf{K}) = \text{gcd}\{|\lambda I - A - BKC|\} \tag{5.2.11b}$$

Definition 5.2. For the system (C, A, B) and the set of output feedback gains K given by (5.2.11), the set of "fixed modes" of (C, A, B) with respect to K is defined as the intersection of all possible sets of the eigenvalues of matrix (A + BKC), i.e.,

$$\Lambda(C, A, B, \mathbf{K}) = \bigcap_{K \in \mathbf{K}} \lambda(A + BKC) \tag{5.2.12}$$

where $\lambda(\cdot)$ denotes the set of eigenvalues of (A + BKC). Note also that K

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

can take on the null matrix; hence the set of "fixed modes" $\Lambda(\cdot)$ is contained in $\lambda(A)$. In view of Definition 5.1, the members of $\Lambda(\cdot)$, i.e., the "fixed modes," are the roots of the "fixed polynomials" $\phi(\cdot;\cdot)$ in (5.2.11b), i.e.,

$$\Lambda(C,A,B,K)=\left\{\lambda\middle|\lambda\in s \text{ and } \phi(\lambda,C,A,B,K)=0\right\} \tag{5.2.13}$$

where s denotes a set of values on the entire complex s-plane.

Even though the above definition and the theorems on decentralized stabilizability of large-scale systems, which will follow soon in this section, provide a complete characterization of "fixed modes," it is often impossible to identify such modes because of the inexact knowledge of the system's parameters (Šiljak, 1991). Due to this fact, structurally "fixed modes" are the consequence of disconnections between the system and some controllers or parts of a single controller. These modes remain fixed unless the control structure is changed.

Let a structured system $\tilde{S}=\left(\tilde{C},\tilde{A},\tilde{B}\right)$; then a system $S = (C, A, B)$ is called admissible if $(C,A,B)\in\left(\tilde{C},\tilde{A},\tilde{B}\right)$. Hence, one can now introduce the following definition.

Definition 5.3 A structured system $\left(\tilde{C},\tilde{A},\tilde{B}\right)$ and a set of output gains $\mathbf{K}$ is said to have a set of "structurally fixed modes" if every admissible system (C, A, B) has "fixed modes" with respect to $\mathbf{K}$.

The following algorithm, due to Davison (1976a), provides a quick way of finding the fixed modes of a system (C, A, B).

Algorithm 5.1. Evaluation of Fixed Modes

Step 1: Find all the eigenvalues of A, i.e., $\lambda(A)$.

Step 2: Choose an arbitrary $m \times r$-dimensional $K \in \mathbf{K}$ (by either a pseudo-random number generator or other means) so that the norm $\|A\| \simeq \|BKC\|$.

Step 3: Find $\lambda(A + BKC)$.

Step 4: Then $\Lambda(C, A, B, K) \subset \lambda(A + BKC)$ with respect to $\boldsymbol{K}$, i.e., the fixed modes of (C, A, B) are contained in those eigenvalues of $(A + BKC)$ which are common with the eigenvalues of A.

Step 5: Steps 2 through 4 may be repeated until the fixed modes of A are identified.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It turns out that the fixed modes of a centralized system $\left(C, A, B, \tilde{K}\right)$, where $\tilde{K}$ is $m \times r$, correspond to the uncontrollable and unobservable modes of the system (Wang and Davison, 1973). The following theorem provides the necessary and sufficient conditions for the stabilizability of a decentralized closed-loop system.

Theorem 5.1. *For the system (C,A,B) in* (5.2.1)–(5.2.2) *and the class of block-diagonal matrices* **K** *in* (5.3.6a), *the local feedback laws* (5.2.4)–(5.2.5) *would asymptotically stabilize the system if and only if the set of fixed modes of (C,A,B,K) is contained in the open left-half s-plane, i.e.,* $\Lambda(C,A,B,K) \in s^-$, *where* s^- *is the open LHP (left-half plane) s-plane.*

The proof of this theorem, which is based on Kalman's canonical structure theorem (Kalman, 1962) and three other lemmas, can be found elsewhere (Wang and Davison, 1973). An important corollary of Theorem 5.1 is that under the conditions of this theorem, a necessary and sufficient condition for a set of eigenvalues of $(A + BKC)$ to belong to a presubscribed set £ is that $\Lambda(C, A, B, K) \subset$ £. The following examples illustrate the evaluation of the fixed modes of the system and check on the stability of decentralized control.

Šiljak (1991) has presented a theorem which characterizes structurally fixed modes algebraically. The existence of such modes, in this case, reduces to either an algebraic condition or a rank condition. Interested readers may contact this reference. Example 5.2.4 will help illustrate the basic difference between "fixed modes" and "structurally fixed modes."

Example 5.2.1. Consider a system

$$\dot{x} = \begin{bmatrix} 0.5 & 0 & 1 & 0 \\ 0.1 & 1.2 & 0 & 0.1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0.4 & 0.75 \end{bmatrix} x + \begin{bmatrix} 0.85 & 0 \\ 0 & 1 \\ 0 & 1.25 \\ 1 & 0 \end{bmatrix} u \tag{5.2.14}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 1 \end{bmatrix} x \tag{5.2.15}$$

It is desired to find the fixed modes, if any, of this system.

SOLUTION: A program was written to simulate Algorithm 5.1. Here are the results of three iterations. The eigenvalues of A are $\lambda(A) = (0.5, 1.2, -1, 0.75)$, which are the diagonal elements of A. For three arbitrary $(m \times r) = (2 \times 2)$-dimensional K matrices,

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$K_i = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} = \begin{bmatrix} 2.248 & 0 \\ 0 & 32.458 \end{bmatrix}, \begin{bmatrix} 1.245 & 0 \\ 0 & 4.258 \end{bmatrix}$$
$$\begin{bmatrix} 2.3588 & 0 \\ 0 & -2.146 \end{bmatrix}, \quad i = 1,2,3$$

the respective eigenvalues of $(A + BKC)$ are

$$\lambda(A + BK_1C) = (1.0375 \pm j1.7866,\ 2.037,\ 1.20)$$
$$\lambda(A + BK_2C) = (1.966 \pm j1.196,\ 3.577,\ 1.20)$$
$$\lambda(A + BK_3C) = (1.66 \pm j0.759,\ -2.41,\ 1.20)$$

Clearly system (5.2.14)–(5.2.15) has a fixed mode $\lambda = 1.2$ and hence according to Theorem 5.1, this system *cannot* be stabilized by decentralized control with dynamic compensators.

Example 5.2.2. Consider a second-order system

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 1 & -2 \end{bmatrix} x + \begin{bmatrix} 0.2 & 0 \\ 0 & 0.5 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 & 5 \\ 2 & 0 \end{bmatrix} x$$

It is desired to check the stabilizability of this system under decentralized control.

SOLUTION: Utilizing Algorithm 5.1 it is found out that none of the eigenvalues of $(A+BKC)$ for several randomly chosen $K=diag\{k_1,k_2\}$ corresponded to $\lambda(A) = (0, -2)$. Thus it is concluded that this system can be stabilized through decentralized dynamic compensation which is discussed shortly and is illustrated by Example 5.2.5.

The following example makes an illustrative comparison between decentralized and centralized control structures.

Example 5.2.3. Consider a seventh-order system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & -2 & -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} u$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$y=\begin{bmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\end{bmatrix}x$$

It is desired to investigate the advantages of a decentralized versus a centralized control structure.

SOLUTION: The A matrix of this system has the following eigenvalues: $\lambda(A)=(-2.472, -0.322\pm j1.43, -1.0, 0.24, -0.562\pm j0.68)$. For the sake of output feedback gains, both decentralized and centralized structures are considered:

$$K_d=\begin{bmatrix}k_1&0&0&0&0\\0&k_2&0&0&0\\0&0&k_3&k_4&k_5\end{bmatrix},\quad K_c=\begin{bmatrix}k_{11}&k_{12}&k_{13}\\k_{21}&k_{22}&k_{23}\\k_{31}&k_{32}&k_{33}\end{bmatrix}$$

For the above gain matrices with several randomly chosen entries, a fixed mode $\lambda=1.0$ resulted. In fact, for several other possible decentralized structures like K_d, it was deduced that there is no particular advantage for using a more complex centralized controller. This result is most desirable for large-scale power systems, as reported for a 26th-order power system by Davison (1976a).

Example 5.2.4 Consider a third-order, two-input, and three-input system $S=(C,A,B)$ discussed by Šiljak (1991) given by

$$A=\begin{bmatrix}1&2&0\\3&4&3\\0&2&1\end{bmatrix},\quad B=\begin{bmatrix}0&0\\0&1\\1&0\end{bmatrix},\quad C=\begin{bmatrix}0&1&0\\1&0&0\\0&1&1\end{bmatrix} \tag{5.2.16}$$

It is desired to investigate the existance of fixed modes and structurally fixed modes.

SOLUTION: Let the feedback structure be given by

$$K=\begin{bmatrix}k_{11}&0&0\\0&k_{22}&k_{23}\end{bmatrix} \tag{5.2.17}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Then, the closed-loop matrix is given by

$$A-BKC=\begin{bmatrix} 1 & 2 & 0 \\ 3-k_{22} & 4 & 3-k_{23} \\ 0 & 2-k_{11} & 1 \end{bmatrix}$$

Using Algorithm 5.1 it is clear that the eigenvalue $\lambda = 1$ of this matrix is independent of k_{ij}, i = 1, 2 and j = 1, 2, 3. Hence, this value (λ = 1) represents a "fixed mode" and system (5.2.16) cannot be stabilized by a decentralized feedback gain (5.2.17). Now, if the a_{33} element in matrix A in (5.2.16) is perturbed by a small amount of ε to get

$$A_\varepsilon = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 3 \\ 0 & 2 & 1+\varepsilon \end{bmatrix}$$

With this structural change, it can be easily seen that the resulting system (C, A_ε, B) has no fixed modes with respect to the decentralized feedback gain (5.2.17). Therefore, the mere fact that the system (C, A, B) in (5.2.16) has a fixed mode is a result of cancellation of nonzero elements of A and a small structural perturbation would eleiminate the fixed mode. To reinforce the difference between the fixed mode and the structurally fixed mode of the system, assume that the elements $a_{11} = a_{33} = 0$ result in

$$A_o = \begin{bmatrix} 0 & 2 & 0 \\ 3 & 4 & 3 \\ 0 & 2 & 0 \end{bmatrix} \tag{5.2.18}$$

Under this new structure, the system (C, A_0, B) has a fixed mode $\lambda = 0$ *regardless* of the values of *all* nonzero elements of A_0 in (5.2.18). In other words, the system (C, A_0, B) has a "structurally fixed mode" regardless of any perturbations that the system ($C + \Delta C$, $A_0 + \Delta A_0$, $B + \Delta B$) may undergo.

5.2.2 Stabilization via Dynamic Compensation

One of the earliest efforts in dynamically compensating centralized systems is due to Brasch and Pearson (1970) and Jamshidi (1983) using output feedback. The problem can be briefly stated as follows: Consider a linear

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

TIV system

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.2.19}$$
$$y(t) = Cx(t) \tag{5.2.20}$$

It is desired to find a dynamic compensator

$$\dot{z}(t) = Fz(t) + Sy(t) \tag{5.2.21}$$
$$u(t) = Hz(t) + Ky(t) \tag{5.2.22}$$

so that the closed-loop system

$$\dot{x}(t) = (A + BKC)x(t) + BHz(t) \tag{5.2.23}$$

has a prescribed set of poles.

For the case of finding the dynamic compensator, let n_c and n_o be the smallest integers such that

$$\text{rank}\left[B, AB, \ldots, A^{n_c}B\right] = n, \quad \text{rank}\left[C^T, A^TC^T, \ldots, A^{Tn_o}C^T\right] = n \tag{5.2.24}$$

Now for convenience, let $\eta = \min.(n_c, n_o)$, and $\Lambda_\eta = \{\lambda_1, \lambda_2, \ldots, \lambda_{n+\eta}\}$ be a set of arbitrary complex numbers with the only restriction being that for each λ_i with $\text{Im}(\lambda_i) \neq 0$, a complex conjugate pair $\lambda_i = \text{Re}(\lambda_i) \pm j\,\text{Im}(\lambda_i)$ is contained in Λ_η. Let us define the following augmented triplet (C_η, A_η, B_η):

$$A_\eta = \begin{array}{c} \left[\begin{array}{c|c} A & 0 \\ \hline 0 & 0 \end{array}\right] \begin{array}{l} \}n \\ \}\eta \end{array} \\ \begin{array}{cc} \underbrace{}_{n} & \underbrace{}_{\eta} \end{array} \end{array}, \quad B_\eta = \begin{array}{c} \left[\begin{array}{c|c} B & 0 \\ \hline 0 & I \end{array}\right] \begin{array}{l} \}n \\ \}\eta \end{array} \\ \begin{array}{cc} \underbrace{}_{m} & \underbrace{}_{\eta} \end{array} \end{array}, \quad C_\eta = \begin{array}{c} \left[\begin{array}{c|c} C & 0 \\ \hline 0 & I \end{array}\right] \begin{array}{l} \}r \\ \}\eta \end{array} \\ \begin{array}{cc} \underbrace{}_{n} & \underbrace{}_{\eta} \end{array} \end{array} \tag{5.2.25}$$

The following theorem determines the existence of an output feedback gain for proper pole placement.

Theorem 5.2. *Let* (C,A,B) *be a controllable and observable system, and let the triple* (C_η, A_η, B_η) *be defined by* (5.2.25) *with* $\eta = \min(n_c, n_o)$ *and a set of prescribed poles* $\Lambda_\eta = (\lambda_1, \lambda_2, \ldots, \lambda_{n+\eta})$. *Then there exists a gain matrix* K *such that the eigenvalues of* $A_\eta + B_\eta KC_\eta$ *are exact ele-*

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ments of Λ_η.

The proof of this theorem with the aid of the properties of cyclic matrices can be found elsewhere (Brasch and Pearson, 1970). The above theorem, the canonical structure theorem of Kalman (1962), and the decentralized stabilization problem of (5.2.1)–(5.2.5) have been used by Wang and Davison (1973) and Davison (1976a) to find a dynamic stabilizing compensator for a large-scale system under decentralized control. Consider that the set of N dynamic compensators (5.2.4)–(5.2.5) and the triplet (C_η, A_η, B_η) in (5.2.25) defines a real constant $(m+\eta)\times(r+\eta)$ K_η matrix,

$$K_\eta = \begin{array}{c} \left[\begin{array}{cccc|cccc} K_1 & & 0 & & H_1 & & 0 & \\ \hline & K_2 & & & & H_2 & & \\ & & \ddots & & & & & \\ & & & K_N & & & & H_N \\ \hline S_1 & & & & F_1 & & & \\ & S_2 & & & & F_2 & & \\ & 0 & \ddots & & & & \ddots & \\ & & & S_N & & & & F_N \end{array}\right] \begin{array}{l} \}\, m_1 \\ \}\, m_2 \\ \vdots \\ \}\, m_N \\ \}\, \eta_1 \\ \}\, \eta_2 \\ \vdots \\ \}\, \eta_N \end{array} \\ \begin{array}{cccccccc} \underbrace{\quad}_{r_1} & \underbrace{\quad}_{r_2} & & \underbrace{\quad}_{\dots r_N} & \underbrace{\quad}_{\eta_1} & \underbrace{\quad}_{\eta_2 \dots} & & \underbrace{\quad}_{\eta_N} \end{array} \end{array} \tag{5.2.26}$$

where K_i, H_i, S_i, F_i are $m_i \times r_i$, $m_i \times \eta_i$, $\eta_i \times r_i$, and $\eta_i \times \eta_i$ submatrices, respectively, defined in (5.2.4)–(5.2.5); m and r are defined in (5.2.3); and $\eta = \sum_{i=1}^{N} \eta_i$ The following proposition summarizes the decentralized control pole placement problem.

Proposition 5.1. *Considering the triplets* (C,A,B) *and* (C_η, A_η, B_η) *defined above and the set of block-diagonal* $\mathbf{K}$ *in* (5.2.6a), *for any set of integers* $\eta_1, \eta_2, \dots, \eta_N$ with $\eta_i \geq 0$, *the following two "fixed polynomials" are identical:*

$$\phi(\lambda; C, A, B, K) = \phi\left(\lambda; C_\eta, A_\eta, B_\eta, K_\eta\right) \tag{5.2.27}$$

where C_η, A_η, B_η, *and* K_η, *are as defined earlier. In other words, the greatest common divisor of* $\det(\lambda I - A - BKC)$ *and* $\det(\lambda I - A_\eta - B_\eta K_\eta C_\eta)$ *are the same.*

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The proof of this proposition is given by Wang and Davison (1973). The result of this proposition and a matrix identity are used to place poles in decentralized output feedback controllers through dynamic compensation, which is illustrated by the following example.

Example 5.2.5. Let us consider the system in Example 5.2.2. It is desired to find a decentralized stabilizing output control such that a prescribed set of eigenvalues are achieved.

SOLUTION: For this example $\lambda(A) = (0, -2)$ and the matrix $A + BKC$ is

$$A+BKC=\begin{bmatrix} 0 & k_1 \\ 1+k_2 & -2 \end{bmatrix} \tag{5.2.28}$$

and $\eta = \min(n_c, n_o) = 1$; hence, A_η, B_η, K_η, and C_η are

$$A_\eta = A_1 = \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 1 & -2 & 0 \\ \hline 0 & 0 & 0 \end{array}\right], \quad B_\eta = B_1 = \left[\begin{array}{cc|c} 0.2 & 0 & 0 \\ 0 & 0.5 & 0 \\ \hline 0 & 0 & 1 \end{array}\right]$$

$$C_\eta = C_1 = \left[\begin{array}{cc|c} 0 & 5 & 0 \\ 2 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array}\right], \quad K_\eta = K_1 = \left[\begin{array}{cc|c} k_1 & 0 & h_1 \\ 0 & k_2 & 0 \\ \hline s_1 & 0 & f_1 \end{array}\right] \tag{5.2.29}$$

where k_1, k_2, s_1, h_1, and f_1 are unknowns which can be found for a desired pole placement of the two decentralized controllers. The closed-loop matrix $A + BKC$ is given by (5.2.28) and $A_1 + B_1K_1C_1$ is

$$A_1+B_1K_1C_1 = \left[\begin{array}{cc|c} 0 & k_1 & 0.2h_1 \\ 1+k_2 & -2 & 0 \\ \hline 0 & 5s_1 & f_1 \end{array}\right] \tag{5.2.30}$$

which, for the special case of $\eta = \eta_1 + \eta_2 = 1 + 0 = 1$, (5.3.29) is expressed by

$$A_1+B_1K_1C_1 = \left[\begin{array}{c|c} A+BKC & B_1h_1 \\ \hline s_1C_1 & f_1 \end{array}\right]$$

which is in agreement with (5.2.9) for $\upsilon(t) = 0$ and a comparison between

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

(5.2.28) and (5.2.30). The problem is to find k_1, k_2, s_1, h_1, and f_1 such that the augmented system has a preassigned set of poles for the decentralized system and compensator, say,

$$\Lambda_\eta = \Lambda_1 = \{-1 \pm j2, -1\} \tag{5.2.31}$$

To achieve this, one notes the following matrix identity:

$$\det\left[\begin{array}{c|c} M_1 & M_{12} \\ \hline M_{21} & M_2 \end{array}\right] = \det(M_1)\cdot\det(M_2 - M_{21}M_1^{-1}M_{12})$$

and applies it to

$$\det(\lambda I - A_1 - B_1K_1C_1) = \det(\lambda I - A - BKC)$$
$$\cdot \det\left[(\lambda - f_1) - (0 \quad 5s_1)(\lambda I - A - BKC)^{-1}\begin{pmatrix} 0.2h_1 \\ 0 \end{pmatrix}\right] \tag{5.2.32}$$

Choosing k_1, and k_2 such that the first two poles are appropriately placed, $k_1 = 1$ and $k_2 = -6$. The remaining unknowns may be obtained by setting the latter part of (5.2.32) equal to zero for $\lambda = \lambda_3 = -1$, i.e.,

$$\det\left[(\lambda - f_1) - (0 \quad 5s_1)(\lambda I - A - BKC)^{-1}\begin{pmatrix} 0.2h_1 \\ 0 \end{pmatrix}\right]$$
$$= \det\left(\lambda - f_1 - \frac{-5h_1s_1}{\lambda^2 + 2\lambda + 5}\right)\Bigg|_{\lambda=-1} = 0$$

or for arbitrary $s_1 = h_1 = 1$, $f_1 = 0.25$. Thus, the decentralized dynamic compensator controllers are

$$\dot{z}_1 = 0.25z_1 + y_1, \qquad z_1(0) = 0$$
$$u_1 = z_1 + y_1$$
$$\dot{z}_2 = 0, \qquad z_2(0) = 0$$
$$u_2 = -6y_2$$

Figure 5.1 shows a block diagram for the two-controller decentralized structure. Note that while Station 1 represents a dynamic compensator, Station 2 is a simple-static, proportional-output feedback compensator. Note also that the eigenvalues of the combined system are not those specified by (5.2.31) for the system with no compensation. In fact, for this decentral-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

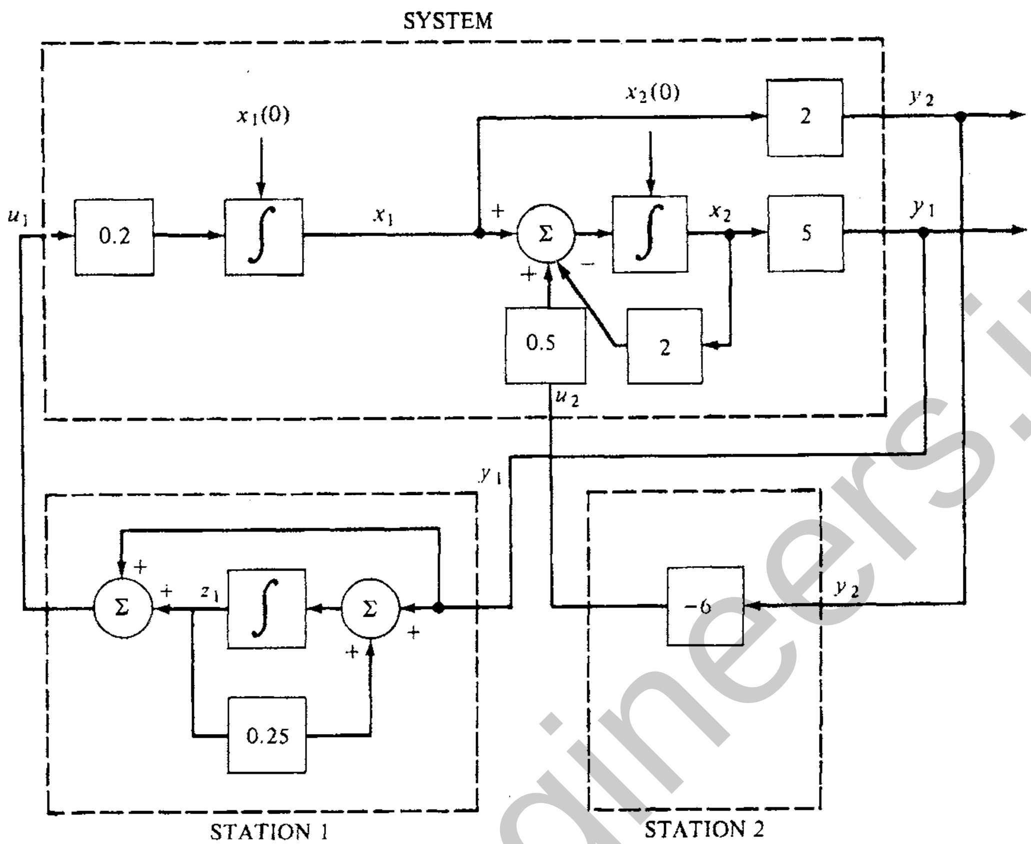


Figure 5.1. A block diagram for the two-controller decentralized system of Example 5.2.5.

ized system the closed-loop poles turned out to be $\lambda(\cdot) = -0.375 \pm j1.9$, -1.0 which are not too far from the desired locations.

Although the pole placement problem has been one of the most fruitful areas of research, until recently (Jamshidi *et al.*, 1992), there has not been an effective computational algorithm for the problem. In its frequency-domain framework, the problem has been very successful (Nyquist, 1932; Bode, 1940; Evans, 1950) within the context of Nyquist, Bode, and root locus plots. In the state-space case, by virtue of canonical transformation, SISO systems have been compensated (Chen, 1970). The multivariable case, considered by many authors (Simon and Mitter, 1968; Retallack and MacFarlane, 1970; Godbout, 1974; Jamshidi *et al.*, 1992), reduces the problem to one of solving a set of nonlinear equations which may or may not be computationally effective.

The pole placement problem has also been taken up by Porter and Bradshaw (1978), and Porter *et al.* (1979) have developed a software package for it. One of the more recent efforts in placing poles in dynamic

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

compensation, due to Lee *et al.* (1979), takes advantage of unity rank gain matrices properties and proposes an algorithm for it. This algorithm, as pointed out by Porter (1980), can potentially lead to erroneous results where an infinite number of gain matrices may satisfy the algorithm. This was subsequently pointed out by Lee *et al.* (1980).

The pole placement schemes for centralized systems are not easily applicable to decentralized systems in general. Such methods can be used separately for each decentralized station in such a way that the n eigenvalues of the system are divided into N groups corresponding to the N controllers. Then utilizing such stabilizing schemes, K_i, $i = 1, 2, \ldots, N$, gains are obtained which results in an overall closed-loop matrix $A - \sum_{i=1}^{N} BK_iC_i$.

5.2.3 Stabilization via Multilevel Control

In this section a decentralized stabilization scheme based on multilevel control (Chapter 4) using local and global controllers is described. This method is based on the works of Šiljak (1978b, 1991), Ikeda and Šiljak (1980b), and Sezer and Šiljak (1981b).

Consider a linear time-invariant system represented in "input-decentralized" form

$$\dot{x}_i(t) = A_i x_i(t) + b_i u_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{N} G_{ij} x_j(t), \quad i = 1, \ldots, N \tag{5.2.33}$$

where all vectors and matrices are of appropriate dimensions and the pairs (A_i, b_i) are controllable. It is desired to apply a decentralized multilevel control

$$u_i(t) = u_i^l(t) + u_i^g(t) \tag{5.2.34}$$

with the ith local control $u_i^l(t)$ and global control $u_i^g(t)$ to be chosen in the following feedback forms:

$$u_i^l(t) = -k_i^T x_i(t) \tag{5.2.35}$$

$$u_i^g(t) = -\sum_{\substack{j=1 \\ j\neq i}}^{N} k_{ij}^T x_j(t) \tag{5.2.36}$$

controlengineers.ir
@controlengineers

where k_i and k_{ij} are n_i- and n_j-dimensional constant vectors. Applying (5.2.35) and (5.2.36) to (5.2.33) results in the closed-loop system

$$\dot{x}(t)=\left(A_i-b_ik_i^T\right)x_i(t)+\sum_{\substack{j=1\\ j\neq i}}^{N}\left(G_{ij}-b_ik_{ij}^T\right)x_j(t),\quad i=1,\ldots,N \tag{5.2.37}$$

It is noted that since each pair (A_i, b_i) is controllable, a feedback gain vector k_i can be always obtained to assign a particular set of eigenvalues to $\left(A_i-b_ik_i^T\right)$ at desired locations $-\sigma_i^1 \pm j\omega_i^1,\ldots,-\sigma_i^r \pm j\omega_i^r, -\sigma_i^{r+1},\ldots,-\sigma_i^{n_i-r}$ where $r = 0, 1, \ldots, n_i/2$ and $\sigma_i^p > 0$ for $p = 1, 2, \ldots, n_i - r$. Then each decoupled subsystem

$$\dot{x}_i(t)=\left(A_i-b_ik_i^T\right)x_i=\hat{A}_ix_i,\quad i=1,\ldots,N \tag{5.2.38}$$

is stabilized with a degree of exponential stability

$$\alpha_i=\min_p \sigma_i^p,\quad p=1,2,\ldots,n_i-r \tag{5.2.39}$$

The essential point of this stabilization scheme, as discussed by Šiljak (1978b, 1991), is that through relatively simple classical techniques each subsystem is stabilized and then through aggregating their stability properties into a single scalar Lyapunov function, the stability of the original large-scale system is deduced. This point will be elaborated further in this and the next section. The stability of interconnected systems was considered in Chapter 3.

The construction of this aggregate model for the interconnected system (5.2.37) is achieved by the proper choice for a Lyapunov function. Let us apply a modal decomposition (transformation) to the decoupled closed-loop subsystem (5.2.38) to obtain

$$\dot{\hat{x}}_i(t)=\Lambda_i\hat{x}_i(t)$$

where

$$x_i(t)=M_i\hat{x}_i(t)$$
$$\Lambda_i=M_i^{-1}\hat{A}_iM_i$$

and the transformation (modal) matrix M_i *is* given by (See Problem 2.6)

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$M_i = \left[\mathrm{Re}\{\xi_i^1\} \mid \mathrm{Im}\{\xi_i^1\} \mid \cdots \mid \mathrm{Re}\{\xi_i^r\} \mid \mathrm{Im}\{\xi_i^r\} \mid \xi_i^{2r+1} \mid \cdots \mid \xi_i^{n_i}\right] \tag{5.2.40}$$

and $\xi_i^m = \mathrm{Re}\{\xi_i^m\} + j\,\mathrm{Im}\{\xi_i^m\}$ and ξ_i^{r+1}, … are complex, real distinct (or generalized) eigenvectors of the decoupled subsystem's closed-loop matrix $\hat{A}_i$. The quasidiagonal matrix Λ_i for the ith subsystem can take on the form

$$\Lambda_i = Block-diag\left\{\begin{bmatrix} -\sigma_i^1 & \omega_i^1 \\ -\omega_i^1 & -\sigma_i^1 \end{bmatrix}, \ldots, \begin{bmatrix} -\sigma_i^r & \omega_i^r \\ -\omega_i^r & -\sigma_i^r \end{bmatrix}, -\sigma_i^{r+1}, \ldots, -\sigma_i^{n_i-r}\right\} \tag{5.2.41}$$

A choice for the aggregated Lyapunov function for the ith subsystem is $\upsilon_i\colon R^{n_i} \to R^+$, i.e.,

$$\upsilon_i(\hat{x}_i) = \left(\hat{x}_i^T \hat{P}_i \hat{x}_i\right)^{1/2} \tag{5.2.42}$$

where

$$\hat{P}_i\Lambda_i + \Lambda_i^T\hat{P}_i + \hat{H}_i = 0 \tag{5.2.43}$$

and

$$\hat{P}_i = \beta_i I_i, \quad \hat{H}_i = 2\beta_i\,\mathrm{diag}\left\{\sigma_i^1, \sigma_i^1, \ldots, \sigma_i^r, \sigma_i^r, \sigma_i^{r+1}, \ldots, \sigma_i^{n_i-r}\right\} \tag{5.2.44}$$

In (5.2.44), β_i is an arbitrary positive constant and I_i is an $n_i \times n_i$ identity matrix. Šiljak (1978b, 1991) has shown that the Lyapunov function (5.2.42) provides an exact estimate of α_i given in (5.2.39). The aggregate Lyapunov function $\upsilon\colon R^n \to R_+^N$ given by

$$\upsilon = (\upsilon_1, \upsilon_2, \ldots, \upsilon_N)^T$$

is obtained for the transformed system (5.2.37)

$$\dot{\hat{x}}_i(t) = \Lambda_i\hat{x}_i(t) + \sum_{\substack{j=1\\ j\neq i}}^{N}\left(\hat{G}_{ij} - \hat{b}_i\hat{k}_{ij}^T\right)\hat{x}_j(t), \quad i = 1, \ldots, N$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and using the Lyapunov functions $\upsilon_i(\hat{x}_i)$ defined in (5.2.42). In (5.2.44) we have $\hat{G}_{ij} = M_i^{-1}G_{ij}M_j$, $\hat{b}_i = M_i^{-1}b_i$, $\hat{k}_{ij}^T = k_{ij}^T M_j$. Using the vector version of the Comparison Principle (Šiljak, 1978b), the aggregate model is represented by

$$\dot{\upsilon} \le \hat{S}\upsilon \tag{5.2.45}$$

where the $N \times N$ constant aggregation matrix $\hat{S}$ has the elements $\hat{s}_{ij}$

$$\hat{s}_{ij} = -\delta_{ij}\alpha_i + \hat{\gamma}_{ij} \tag{5.2.46}$$

and δ_{ij} *is* the Kronecker delta, α_i is given by (5.2.39), and

$$\hat{\gamma}_{ij} = \lambda_M^{1/2}\left\{\left(\hat{G}_{ij} - \hat{b}_i\hat{k}_{ij}^T\right)^T\left(\hat{G}_{ij} - \hat{b}_i\hat{k}_{ij}^T\right)\right\} \tag{5.2.47}$$

The term $\lambda_M\{\cdot\}$ represents the maximum eigenvalue of its matrix argument. Then by applying Sevastyanov-Kotelyanskii condition

$$(-1)^m \begin{vmatrix} -\alpha_1 + \hat{\gamma}_{11} & \hat{\gamma}_{12} & \cdots & \hat{\gamma}_{1m} \\ \hat{\gamma}_{21} & -\alpha_2 + \hat{\gamma}_{22} & & \hat{\gamma}_{2m} \\ \vdots & & \ddots & \\ \hat{\gamma}_{m1} & \hat{\gamma}_{m2} & & -\alpha_m + \hat{\gamma}_{mm} \end{vmatrix} > 0, \quad m = 1, 2, \ldots, N \tag{5.2.48}$$

it can be proved (Šiljak, 1978b) that the origin is an "exponentially connectively stable"* equilibrium point in the large for the system (5.2.44). One way to satisfy conditions (5.2.48) is to choose $\hat{k}_{ij}$ in (5.2.44) such that the subsystem's interconnection strengths represented by the nonnegative numbers $\hat{\gamma}_{ij}$ are minimized. Using the Moore-Penrose generalized inverse of $\hat{b}_i$ (Jamshidi, 1983), the interconnecting gains are given by

$$\hat{k}_{ij}^0 = \left[\left(\hat{b}_i^T\hat{b}_i\right)^{-1}\hat{b}_i^T\hat{G}_{ij}\right]^T \tag{5.2.49}$$

* For a definition of connective stability, see Section 3.2.3.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $\left(\hat{b}_i^T \hat{b}_i\right)^{-1} \hat{b}_i^T$ is the generalized inverse of $\hat{b}_i$. Using these choices for gains $\hat{k}_{ij}$, the overall system (5.2.44) becomes

$$\dot{\hat{x}}_i(t) = \Lambda_i \hat{x}_i(t) + \left[I_i - \hat{b}_i \left(\hat{b}_i^T \hat{b}_i\right)^{-1} \hat{b}_i^T \right] \sum_{\substack{j=1 \\ j \neq i}}^{N} \hat{G}_{ij} \hat{x}_j, \quad i = 1, 2, \ldots, N \tag{5.2.50}$$

where I_i is an $n_i \times n_i$ identity matrix. In order to check the stability of this system, it suffices to check the conditions (5.2.48) using the optimal aggregate matrix $S^o = \left(s_{ij}^o\right)$ defined by (5.2.46) and parameters γ_{ij}, which are now defined by

$$\hat{\gamma}_{ij} = \hat{\gamma}_{ij}^o = \lambda_M^{1/2} \left\{ \hat{G}_{ij}^T \left[I_i - \hat{b}_i \left(\hat{b}_i^T \hat{b}_i\right)^{-1} \hat{b}_i^T \right]^T \left[I_i - \hat{b}_i \left(\hat{b}_i^T \hat{b}_i\right)^{-1} \hat{b}_i^T \right] \hat{G}_{ij} \right\} \tag{5.2.51}$$

Šiljak (1978b) has provided two theorems which essentially, through (5.2.48), guarantee the stability of a large-scale system $\dot{x} = Ax + Bu$ using the feedback gains k_i^T and k_{ij}^T thus obtained. The following algorithm illustrates this stabilization scheme.

Algorithm 5.2. Stabilization via Multilevel Control

Step 1: Use a canonical transformation (Jamshidi *et al.*, 1992) to represent the system in its companion (canonical) input-decentralized form.

Step 2: Use any simple pole placement scheme to find local state feedback gains k_i^T for each subsystem controllable pair (A_i, b_i), $i = 1, \ldots, N$.

Step 3: Using the new decoupled subsystem's closed-loop matrices $\hat{A}_i = \left(A_i - b_i k_i^T\right)$, determine the transformation (modal) matrices M_i in (5.2.40) and evaluate the transformed vectors and matrices $\hat{b}_i$ and $\hat{G}_{ij}$ for $i, j = l, \ldots, N;\ i \neq j$.

Step 4: Evaluate the interconnection gains $\hat{k}_{ij}^o$ using (5.2.49) and obtain the overall system state equations (5.2.50).

Step 5: To check stability of the overall system, use aggregate matrix

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$\hat{S}$ defined by (5.2.46) and (5.2.51) to check for conditions (5.2.48).

Step 6: If conditions (5.2.48) are not satisfied, Steps 2 through 4 can be repeated using local subsystem feedback gains based on larger values of α_i in (5.2.39).

Step 7: Stop.

The following fifth-order system is used to illustrate this algorithm.

Example 5.2.6. Consider an interconnected system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0.1 & 1 \\ 4 & -1 & 2 & 0 & 0.5 \\ 0.4 & 0.2 & 0 & 0 & 1 \\ 0.5 & 0.2 & 1 & -1 & 2 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u \tag{5.2.52}$$

The system matrix A has its eigenvalues at $(3.5, 0.47 \pm j1.56, -0.21 \pm j0.6)$, which indicates that the system is unstable. Use Algorithm 5.2 to stabilize (5.2.52) using multilevel control.

SOLUTION: For a 3×2 decomposition of the system, it is clear that the system (5.2.52) would already be in input-decentralized and companion form. The eigenvalues of the two 3×3 and 2×2 companion submatrices A_1 and A_2 are $\lambda\{A_1\} = \{-0.157 \pm j1.3, 2.31\}$ and $\lambda\{A_2\} = \{1,1\}$. Let the desired poles for the decoupled subsystems closed-loop submatrices $\hat{A}_i$, $i = 1, 2$, be at

$$\lambda\{\hat{A}_1\} = \{-5 \pm j2, -10\}, \quad \lambda\{\hat{A}_2\} = \{-2 \pm j\}$$

which would provide local feedback gains

$$k_1^T = (294 \quad 128 \quad 22), \quad k_2^T = (4 \quad 6)$$

The transformation matrices M_1 and M_2 of Step 3 of Algorithm 5.2 turn out to be

$$M_1 = \begin{bmatrix} 0.02497 & 0.02378 & -0.00995 \\ -0.1724 & -0.06896 & 0.09950 \\ 1 & 0 & -0.99500 \end{bmatrix}, \quad M_2 = \begin{bmatrix} -0.4 & 0.2 \\ 1 & 0 \end{bmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which lead to the following interconnected subsystems:

$$\dot{\hat{x}}_1 = \begin{bmatrix} -5 & 2 & 0 \\ -2 & -5 & 0 \\ 0 & 0 & -10 \end{bmatrix} \hat{x}_1 + \begin{bmatrix} -94.34 & -20.70 \\ 84.40 & 21.43 \\ -95.32 & -20.80 \end{bmatrix} \hat{x}_2 + \begin{bmatrix} -2.45 \\ 1.12 \\ -3.46 \end{bmatrix} u_1^g$$

$$\dot{\hat{x}}_2 = \begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix} \hat{x}_2 + \begin{bmatrix} 0.98 & -0.002 & -0.98 \\ 1.83 & -0.025 & -1.90 \end{bmatrix} \hat{x}_1 + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u_2^g \tag{5.2.53}$$

Note that the system is still not identical to the system (5.2.44). Thus, what remains to be found are the global controls u_i^g, $i = 1, 2$. However, before the gains k_{ij}^{oT} are obtained, an investigation is made on the effects of the global controls by setting the gain vectors $\hat{k}_{12}$ and $\hat{k}_{21}$ to zero and checking the stability conditions (5.2.48). From (5.2.39) and (5.2.53), it is clear that $\alpha_1 = 5$ and $\alpha_2 = 2$, while by using (5.2.47) and (5.2.53) one computes $\hat{\gamma}_{21} = 2.97$ and $\hat{\gamma}_{12}=162.5$. The aggregate matrix $\hat{S}$ in (5.2.45) becomes

$$\hat{S} = \begin{bmatrix} -5 & 2.97 \\ 162.5 & -2 \end{bmatrix}$$

which violates the stability conditions (5.2.48). Thus, the local feedback controls would not stabilize the overall system.

Now by utilizing (5.2.49) and Step 4 of Algorithm 5.2, the interconnection gains turn out to be

$$\hat{k}_{12}^{oT} = (34.05 \quad 7.6), \quad \hat{k}_{21}^{oT} = (0.93 \quad -0.01 \quad -0.95)$$

which, when used to replace the global controls u_i^g, $i = 1, 2$, using (5.2.37), yields the subsystems (5.2.53) as

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \hline \dot{\hat{x}}_2 \end{bmatrix} = \left[\begin{array}{ccc|cc} -5 & 2 & 0 & -10.96 & -2.04 \\ -2 & -5 & 0 & 46.24 & 12.9 \\ 0 & 0 & -10 & 22.70 & 5.61 \\ \hline 0.05 & 0.008 & -0.03 & -2 & -1 \\ -0.024 & -0.004 & 0.016 & 1 & -2 \end{array}\right] \begin{bmatrix} \hat{x}_1 \\ \hline \hat{x}_2 \end{bmatrix} \tag{5.2.54}$$

controlengineers.ir @controlengineers

The system provides parameters $\hat{\gamma}_{21} = 0.066$ and $\hat{\gamma}_{12} = 54.5$ and hence an aggregate matrix $\hat{S}$,

$$\hat{S}^o = \begin{bmatrix} -5 & 54.5 \\ 0.066 & -2 \end{bmatrix}$$

which satisfies the stability conditions (5.2.48). In fact, the eigenvalues of the closed-loop matrix in (5.2.54) turn out to be $(-5.06 \pm j2.2, -1.97 \pm j0.86, -9.93)$, indicating that system (5.2.52) has been stabilized by a two-level control scheme.

The above formulation for a decentralized stabilization scheme by multi-level control has been restricted to single inputs. The extension to multivariable systems follows the above development rather closely. This is considered in part in the next section under exponential stabilization.

5.2.4 Exponential Stabilization

In this section a decentralized stabilization procedure with a prescribed degree for the closed-loop composite system is considered. Most of this section is based on the works of Sundareshan (1977a) and Šiljak (1978b).

Consider an input-decentralized large-scale system consisting of N subsystems:

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + g_i(t,x), \quad i = 1,\dots,N \tag{5.2.55}$$

where each decoupled subsystem (A_i, B_i) is assumed to be completely controllable, $g_i(\cdot,\cdot): R \times R^n \to R^{n_i}$ describes the interaction function between the ith subsystem, and the $N-1$ and all other matrices have appropriate dimensions. The problem is to find a state feedback decentralized control $u_i(t) = \psi_i(t,x)$ such that all the solutions of the compensated system

$$\dot{x}_i(t) = A_i x_i(t) + B_i \psi_i(t,x) + g_i(t,x), \quad i = 1,\dots,N$$

satisfy $x_i(t)\exp(-\alpha t) \to 0$ as $t \to \infty$, where $\alpha > 0$ is the prescribed degree of exponential stability. An exponential stabilized system is formally defined as follows.

Definition 5.4. The system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is called "exponentially stabilizable" by a control law $u = Px + Fy$ if for any prespecified positive number α, a feedback gain matrix P exists such that the closed-loop system state $x(t)$ satisfies

$$\|x(t_k)\| \le \gamma \|x(t_{k-1})\| \exp\{-\alpha(t_k - t_{k-1})\}$$

for a positive number γ, all t, and $t_k \ge t_{k-1}$.

As a sequel, a completely decentralized scheme for the solution of the above problem is given.

Let the interactions $g_i(t, x)$ be zero for (5.2.55) to provide a set of completely decoupled linear subsystems

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t), \quad i = 1, \ldots, N \tag{5.2.56}$$

It is well known (Kwakernaak and Sivan, 1972) that each of the subsystems in (5.2.56) can be exponentially stabilized with a prescribed degree α by control functions

$$u_i(t) = -B_i^T K_i x_i(t) \tag{5.2.57}$$

where the K_i is an $n_i \times n_i$ symmetric positive-definite solution of the algebraic matrix Riccati equation (AMRE),

$$(A_i + \alpha I_i)^T K_i + K_i(A_i + \alpha I_i) - K_i S_i K_i + Q_i = 0 \tag{5.2.58}$$

where $S_i = B_i B_i^T$, $Q_i = C_i^T C_i$ is a nonnegative definite matrix such that the pair (A_i, C_i) is completely observable, and I_i is an $n_i \times n_i$ identity matrix. Under these conditions each closed-loop subsystem

$$\dot{x}_i(t) = (A_i - S_i K_i) x_i(t), \quad i = 1, \ldots, N$$

has the property that $x_i(t)\exp(-\alpha t) \to 0$ as $t \to \infty$. What is left is to find conditions on the interactions $g_i(t, x)$ such that with the completely decentralized controls (5.2.57) the overall system (5.2.55) is exponentially stabilized within the above context, i.e., the system

$$\dot{x}_i(t) = (A_i - S_i K_i) x_i(t) + g_i(t, x), \quad i = 1, \ldots, N$$

has the same property with respect to $\exp(-at)$. The above equations can

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

be combined to obtain a description for the composite system

$$\dot{x}(t) = (A - SK)x(t) + g(t,x) \tag{5.2.59}$$

where A = Block-diag($A_1, \dots, A_N$), S = Block-diag($S_1, \dots, S_N$), and K = Block-diag($K_1, \dots, K_N$). The following two theorems, due to Sundareshan (1977b), set up conditions for decentralized exponential stabilization with respect to the structural arrangement of interconnections or the strength (the magnitude of information flow) of interconnections among various subsystems.

Theorem 5.3. *If the interconnection vector $g(t, x)$ can be represented by*

$$g(t,x) = \left[H(t,x) - W(t,x)\right]Kx(t) \tag{5.2.60a}$$

where $H(\cdot,\cdot)$ and $W(\cdot,\cdot)$ are two $n \times n$ arbitrary skew-symmetric and symmetric matrix functions, respectively, and if the $n \times n$ matrix

$$U(t,x) = P + 2KW(t,x)K \tag{5.2.60b}$$

is nonnegative-definite, then the system (5.2.59) *is exponentially stable with degree α. In* (5.2.59), *P = Block-diag($P_1, \dots, P_N$), where $P_i = Q_i + K_i S_i K_i$.*

PROOF: The proof proceeds with the aid of the Lyapunov stability theory. Let $\upsilon(x) = x^T(t)Kx(t)$ be a potential Lyapunov function for (5.2.59). Since each K_i, $i = 1, \dots, N$, is positive-definite, $\upsilon(x)$ is also positive-definite. Taking the time derivative of $\upsilon(x)$ along the trajectories of (5.2.59), one obtains

$$\dot{\upsilon}(x) = x^T(t)\left[(A - SK)^T K + K(A - SK)\right]x(t) + x^T(t)Kg(t,x) \\ + g^T(t,x)Kx(t) \tag{5.2.61}$$

Now by using the AMRE (5.2.59) in its augmented form, (5.2.61) can be simplified to

$$\dot{\upsilon}(x) = x^T(t)[-2\alpha K - P]x(t) + x^T(t)Kg(t,x) + g^T(t,x)Kx(t) \tag{5.2.62}$$

Using the expression for $g(t, x)$ from (5.2.60a) and the properties of matrix functions $H(\cdot,\cdot)$ and W$(\cdot,\cdot)$, (5.2.62) is simplified even further to

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{\upsilon}(x) = x^T(t)\left[-2\alpha K - P - 2KW(t,x)K\right]x(t) \tag{5.2.63}$$

Now in view of the nonnegative definiteness assumption of $U(t, x)$ in (5.2.60b), (5.2.63) can be changed to an inequality

$$\dot{\upsilon}(x) \leq -2\alpha\upsilon(x) \tag{5.2.64}$$

for all $n \times 1$ x vectors. By integrating (5.2.64) and making use of definitions of $\upsilon(x)$ and $\upsilon(x(t_o))$, it will lead to

$$\|x(t)\| \leq \gamma \|x(t_0)\| \exp\left(-\alpha(t-t_0)\right)$$

for all $t > t_0$ and $\gamma = \lambda_M^{1/2}(K)\lambda_m^{-1/2}(K)$, where $\lambda_M(\cdot)$ and $\lambda_m(\cdot)$ represent, respectively, the maximum and minimum eigenvalues of their matrix arguments. Thus, the system (5.2.59) is exponentially stable with a prescribed degree α. Q.E.D. ■

The case when the interaction term $g_i(t, x)$ is linear would result in simpler conditions. Theorem 5.4 provides conditions for exponential stability in terms of the strength of the interconnections.

Theorem 5.4. *If the interconnections $g_i(t, x)$ satisfy the inequality*

$$\|g_i(t,x)\| \leq \sum_{j=1}^{N} \gamma_{ij} \|x_j\|, \quad i = 1, \ldots, N \tag{5.2.65}$$

for all $(t, x) \in R^{n+1}$, where γ_{ij} are N^2 nonnegative numbers, and if $\gamma = \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_{ij}$ satisfies

$$\min_i\left(\lambda_m(P_i)\right) \geq 2\gamma \max_i\left(\lambda_M(K_i)\right) \tag{5.2.66}$$

the composite system (5.2.59) is *exponentially stable with degree α.*

PROOF: The proof of this theorem closely follows that of Theorem 5.3. Making a choice of Lyapunov function $\upsilon(x) = x^T(t)Kx(t)$ and evaluating $\dot{\upsilon}(x)$ as before, one obtains

$$\dot{\upsilon}(x) = x^T(t)(-2\alpha K - P)x(t) + 2g^T(t,x)Kx(t) \tag{5.2.67a}$$

The time derivative of $\upsilon(x)$ satisfies the exponential stability condition

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{v}(x) \le -2\alpha v(x) \tag{5.2.67b}$$

for all $x \in R^n$ if

$$x^T(t)Px(t) \ge 2g^T(t,x)Kx(t) \tag{5.2.68}$$

for $x \in R^n$. Moreover, utilizing triangular-type inequality and (5.2.65), it is easy to observe that

$$\|g(t,x)\| \le \sum_{i=1}^{N} \|g_i(t,x)\| \le \sum_{i=1}^{N}\sum_{j=1}^{N} \gamma_{ij} \|x_j\| \le \gamma \|x\|$$

which implies that condition (5.2.66) is sufficient to ensure (5.2.68). Therefore, (5.2.67b) holds and the theorem has been proved. Q.E.D. ■

It is noted again that if the interconnection vector is linear, i.e., $g(t, x) = G(t)x(t)$, then the parameter γ in (5.2.66) can be obtained simply as $\gamma = \mathrm{Sup}_{t\in R} \lambda_M^{1/2}(G^T(t)G(t))$. Moreover, it is possible to generate simpler stabilizing controls through the solution of matrix Lyapunov equations instead of more difficult matrix Riccati equations. The following example illustrates the complete decentralized exponential stabilization method.

Example 5.2.7. Consider a fourth-order system

$$\dot{x} = \left[\begin{array}{cc:cc} 0 & 0 & 0 & -0.1 \\ 0.5 & -1 & 0.15 & 0 \\ \hdashline -0.1 & 0 & 1 & -0.1 \\ 0 & 0.15 & 2 & -0.8 \end{array}\right] x + \left[\begin{array}{c:c} 1 & 0 \\ 0 & 0 \\ \hdashline 0 & 1 \\ 0 & 0.2 \end{array}\right] u \tag{5.2.69}$$

with eigenvalues $\lambda\{A\} = \{-0.26 \pm j0.18, 1.07, -1.36\}$. It is desired to exponentially stabilize this system through complete decentralized control.

SOLUTION: The two decoupled subsystems (A_1, B_1) and (A_2, B_2) are completely controllable. Using an $\alpha = 2$ and $Q_i = 2I_2$, $i = 1, 2$, the AMREs (5.2.58) are solved by Newton's iterative method using the generalized inverse initialization scheme (Jamshidi, 1980):

$$K_1 = \begin{bmatrix} 1.48 & 0.198 \\ 0.198 & 0.98 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 3 & 0.37 \\ 0.37 & 1 \end{bmatrix} \tag{5.2.70}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

with eigenvalues $\lambda\{K_1\} = \{0.912, 1.55\}$ and $\lambda\{K_2\} = \{0.93, 3.07\}$. The two decentralized controls turn to be $u_1 = -1.48x_1 - 0.198x_2$ and $u_2 = -3.08x_3 - 0.6x_4$. Since the subsystems' interconnections are linear, the value of $\gamma = \lambda_M^{1/2}(G^T G) = 0.15$. The magnitudes of the eigenvalues of P_i, $i = 1, 2$, are

$$\lambda\{P_1\} = \lambda\{Q_1 + K_1 S_1 K_1\} = \lambda\left\{\begin{bmatrix} 4.2 & 0.29 \\ 0.29 & 2.04 \end{bmatrix}\right\} = \{2, 4.24\} \tag{5.2.71}$$

and

$$\lambda\{P_2\} = \lambda\{Q_2 + K_2 S_2 K_2\} = \lambda\left\{\begin{bmatrix} 11.45 & 1.76 \\ 1.76 & 2.33 \end{bmatrix}\right\} = \{2, 11.78\} \tag{5.2.72}$$

Therefore, the terms of condition (5.2.66) would become

$$\min_i(\lambda_m(P_i)) = 2, \quad \max_i(\lambda_M(K_i)) = 3.07, \quad \gamma = \lambda_M^{1/2}(G^T G) = 0.15 \tag{5.2.73}$$

which leads to

$$\min_i \lambda_m(P_i) = 2 > 2(0.15)\max_i \lambda_M(K_i) = (0.3)(3.07) = 0.921 \tag{5.2.74}$$

indicating that the system (5.2.69) is exponentially stable in accordance to Theorem 5.4. Next, a structural perturbation was made on the interconnections, i.e., the G matrix is assumed to take on a form given by

$$G(\varepsilon) = \left[\begin{array}{cc|cc} 0 & 0 & 0.5\varepsilon & -0.1 \\ 0 & 0 & 0.15 & -\varepsilon \\ \hline -0.1 & 0.5\varepsilon & 0 & 0 \\ -\varepsilon & 0.15 & 0 & 0 \end{array}\right] \tag{5.2.75}$$

The values of the perturbation parameter ε were varied between $-1 \le \varepsilon \le 1$. The value of the right-hand side of (5.2.74) less the left-hand side value, i.e., $f(\varepsilon) = 3.07\,\gamma(\varepsilon) - 1$, was calculated. Clearly, for all values of ε for which $f(\varepsilon) \le 0$, the system can be stabilized exponentially. The plot of $f(\varepsilon)$ versus ε is shown in Figure 5.2. For the set of all values of ε such that $-0.246 \le \varepsilon \le 0.246$, the system is exponentially stabilized.

In a fashion similar to the one in the last section, one can use an additional global control $u_i^g(t)$ similar to (5.2.34) for each subsystem. Using a

controlengineers.ir

controlengineers.ir
@controlengineers

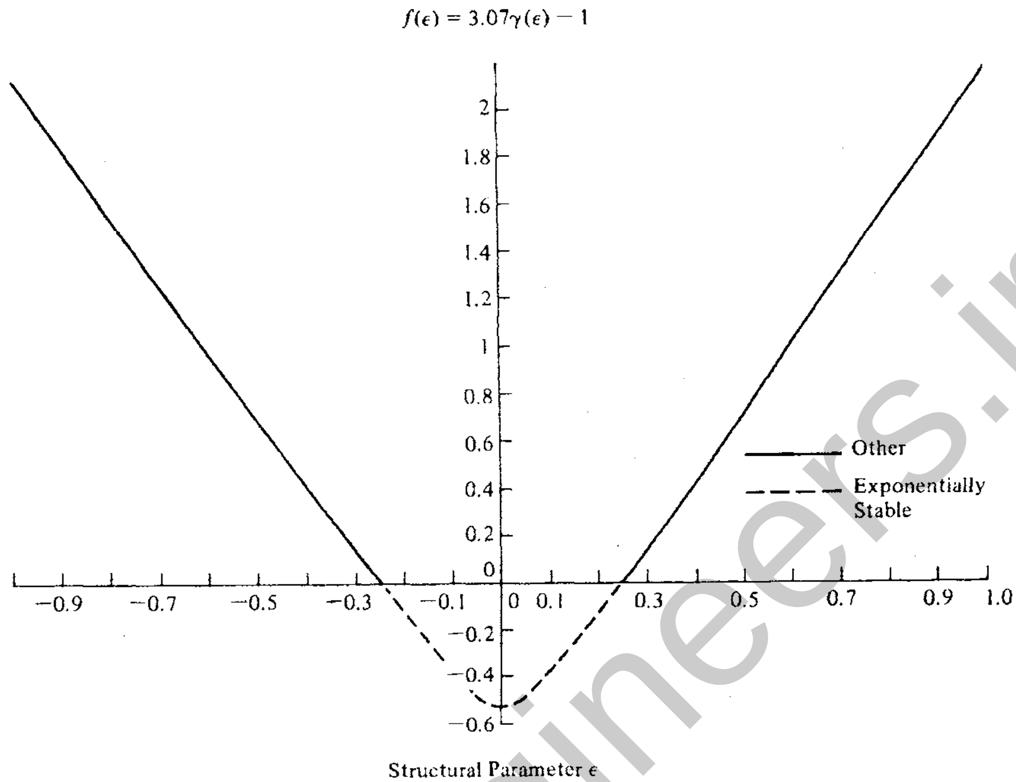


Figure 5.2 A graphical representation for structural perturbation of exponential stability in Example 5.2.7.

suitable negative state feedback form for $u_i^g(t) = -\phi_i(t,x)$, the composite system can be represented by

$$\dot{x}(t) = (A - SK)x(t) + [g(t,x) - B\phi(t,x)] \tag{5.2.76}$$

where $\phi^T(\cdot) = [\phi_1^T(\cdot), \ldots, \phi_N^T(\cdot)]$. A comparison of (5.2.76) and (5.2.59) indicates that, as in Chapter 4 under structural perturbations and discussions of the previous section, the global control would offset any perturbation of the interconnections. If B is nonsingular, an obvious choice for $\phi(\cdot,\cdot)$ would be

$$\phi(t,x) = B^{-1}g(t,x) \tag{5.2.77}$$

For further discussions on the implications of the above relation and the case of linear interconnections, see Problem 5.5

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

5.3 Decentralized Adaptive Control

In many decentralized control systems, just as in nondecentralized cases, the system's performance is heavily affected by the plant's unmodeled dynamics, parameters, uncertain values, disturbances, etc. Experience on many actual, real-time large-scale systems shows a considerable improvement in the gain allocation as well as in the performance of the closed-loop system can be achieved using adaptive decentralized feedback. The local gains are adaptively adjusted to the levels necessary to neutralize the interconnections and, at the same time, drive the subsystems with unknown parameters to the relaxed operating point or, in the case of regulation, toward the performance of the locally chosen reference models.

In recent years, decentralized control approaches combined with adaptive schemes have appeared in the literature (Ortega and Tang, 1989; Peterson and Narendra, 1982), especially on control of robot manipulators. One adaptive decentralized state feedback design deals with the case of single input subsystems and the knowledge of the local control gain vectors. On the other hand, the design approach proposed in Ortega and Tang (1989) treats the multiinput subsystem case but requires knowledge of the local control gain matrices as well as the subsystem interconnection matrices.

In this section we use the results of the model reference adaptive control (MRAC) of linear systems via state feedback (Ioannou and Kokotović, 1983) to obtain an adaptive decentralized control for the case of subsystems with multiple inputs and unknown parameters and interconnections. Sufficient conditions for decentralized adaptive regulation in the form of algebraic criteria are established which guarantee the asymptotic stability under certain structural perturbations. In the case of tracking, the decentralized adaptive control scheme guarantees boundedness of all the closed-loop system signals and the convergence of the state error to a residual set.

Finally, this control structure is applied to the model of a liquid-metal cooled nuclear reactor (LMR) (Monopoli, 1974) operating as a baseline power plant. The subsystems considered are the reactor core and the primary heat transport loop. Simulation results are also included (Benitez-Read and Jamshidi, 1992).

5.3.1 Decentralized Adaptation

Consider a multivariable linear, time-invariant system which is described as an interconnection of N subsystems and is represented by

$$\dot{x}_i = A_i x_i + B_i u_i + g_i(x) \qquad i = 1, \ldots, N \tag{5.3.1}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$g_i(x) = \sum_{\substack{j=1 \\ j \neq i}}^{N} A_{ij} x_j \tag{5.3.2}$$

which for the ith subsystem $x_i \in R^{n_i}$ is the state vector, $u_i \in R^{l_i}$ is the control vector, and $g_i(x) \in R^{n_i}$ is the interaction vector from the other subsystems. The parameters A_i, B_i, and A_{ij} are unknown constant matrices and all the pairs (A_i , B_i) are completely controllable. The composite system is described as

$$\dot{x} = Ax + Bu + Hx \tag{5.3.3}$$

where $x = \left[x_1^T \cdots x_N^T\right]^T$ is the overall system state vector, $u = \left[u_1^T \cdots u_N^T\right]^T$ is the corresponding control vector, the matrix

$$H = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1N} \\ A_{21} & 0 & & \vdots \\ \vdots & & & \vdots \\ A_{N1} & \cdots & \cdots & 0 \end{bmatrix} \tag{5.3.4}$$

is the interconnection matrix, and the overall decoupled state and control matrices are given by A = diag (A_i) and B = diag (B_i), respectively.

The decentralized adaptive control problem is to design a set of N local adaptive controllers u_i such that the states of the composite system (5.3.3) are regulated to zero or track the state trajectories of a given reference model formed by the locally defined reference system models. Each subsystem is controlled independently on the basis of its own performance criterian and locally provided information, that is, there is no sharing of information among the local controllers.

The ith stable local reference model is given by

$$\dot{x}_{mi} = A_{mi} x_{mi} + B_{mi} r_i \qquad i = 1, \ldots, N \tag{5.3.5}$$

where $x_{mi} \in R^{n_i}$ is the reference state vector, $r_i \in R^{l_i}$ is a bounded piecewise continuous reference input vector, and A_{mi} is a stable matrix. It is further assumed that an ($l_i \times n_i$) matrix K_i^* and an ($l_i \times l_i$) matrix L_i^* exist,

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

such that

$$A_i + B_{mi} K_i^* = A_{mi} \tag{5.3.6}$$

$$B_i L_i^* = B_{mi} \tag{5.3.7}$$

which are known as the matching conditions between the plant and the reference model (Ioannou and Kokotović, 1983). The composite reference system is described by

$$\dot{x}_m = A_m x_m + B_m r \tag{5.3.8}$$

where $x_m = \left[x_{m1}^T \cdots x_{mN}^T\right]^T$, $r = \left[r_1^T \cdots r_N^T\right]^T$, $A_m = \text{diag}\left(A_{mi}\right)$, and $B_m = \text{diag}$ (B_{mi}).

In the case of regulation, that is, $r = 0$ and $x_m = 0$, the objective is to find local adaptive control inputs u_i to drive the states of the plant to the origin. For the tracking problem, the local controllers are determined such that the error $e = x - x_m$ between the plant and the reference model, as well as all the signals in the closed-loop system remain bounded. Due to the interconnections $g_i(x)$, $i = 1, \ldots, N$, it is not possible to ensure $\lim_{n\to\infty} e(t) = 0$ for all bounded reference input vectors r. However, we can achieve convergence of the state error e to some bounded residual set.

In summary, the decentralized control problem is one of determining the local control inputs u_i, $i = 1, \ldots, N$, such that the state of the plant is driven to zero or tracks some desired trajectory, while at the same time stability of the closed-loop system is maintained. The solution to the regulation and tracking problems is covered in the next two sections.

5.3.2 *Decentralized Regulation Systems*

In the regulation problem, the reference input r, as well as the state of the reference model x_m, are set to zero, and the local state feedback controllers are chosen as

$$u_i(t) = L_i(t) K_i(t) x_i(t) \tag{5.3.9}$$

where the feedback gain matrix $K_i(t)$ and the feedforward matrix $L_i(t)$ are adjusted according to the adaptive laws

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{K}_i = -\left(B_{mi}^T P_i x_i\right) x_i^T \Gamma_i \tag{5.3.10}$$

$$\dot{L}_i = -L_i\left(B_{mi}^T P_i x_i\right)\left(K_i x_i\right)^T L_i^T \Lambda_i L_i \tag{5.3.11}$$

where $\Gamma_i = \Gamma_i^T > 0$, $\Lambda_i = \Lambda_i^T > 0$, $P_i = P_i^T > 0$, and P_i satisfies the Lyapunov equation

$$A_{mi}^T P_i + P_i A_{mi} = -Q_i, \quad Q_i = Q_i^T > 0 \tag{5.3.12}$$

The closed-loop subsystems are described by

$$\dot{x}_i = \left[A_{mi} + B_i\left(L_i K_i - L_i^* K_i^*\right)\right] x_i + \sum_{\substack{j=1 \\ j \neq i}}^{N} A_{ij} x_j \tag{5.3.13}$$

The closed-loop decoupled subsystems have the property that if $K_i(t) \equiv K_i^*$ and $L_i(t) \equiv L_i^*$, then the plant together with the controller are identical to the reference model, hence, the closed loop system stability is guaranteed. As shown below, the adaptive laws (5.3.10) and (5.3.11) assure boundedness of the parameters and convergence of the state errors to zero for the decoupled subsystems only when the initial parameter values $K_i(t_o)$ and $L_i(t_o)$ lie in the vicinity of the desired values K_i^* and L_i^*.

The presence of interconnections among subsystems can change the stability properties of the decoupled subsystems, and thus, it is necessary to obtain sufficient structural conditions to guarantee the stability of the overall system. These conditions are given by the following theorem.

Theorem 5.5 Let

$$q_i = \min \lambda\left(Q_i\right) \tag{5.3.14}$$

$$a_{ij} = \left\| P_i A_{ij} \right\| \tag{5.3.15}$$

where $\lambda(Q_i)$ is the set of eigenvalues of Q_i. These eigenvalues are all positive-real from the symmetric positiveness condition on Q_i. If there exist constants $\delta_i > 0$, $i = 1, \ldots, N$, such that the $N \times N$ matrix S with elements

$$S_{ij} = \begin{cases} \delta_i q_i & i = j \\ -\left(\delta_i a_{ij} + \delta_j a_{ji}\right) & i \neq j \end{cases} \tag{5.3.16}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is positive-definite, then the controller parameters $K_i(t)$, $L_i(t)$ are bounded and $\lim \left\| x(t) \right\| = 0$ as $t \to \infty$ when the initial conditions of $K_i(t)$ and $L_i(t)$ lie in some bounded neighborhood of their matching condition values K_i^* and L_i^*.

PROOF: Consider the positive definite function

$$V = \sum_{i=1}^{N} \delta_i \left[V_{i1} + V_{i2} + V_{i3} \right] \tag{5.3.17}$$

where

$$V_{i1} = x_i^T P_i x_i \tag{5.3.18}$$

$$V_{i2} = tr\left[\left(K_i - K_i^* \right) \Gamma_i^{-1} \left(K_i - K_i^* \right)^T \right] \tag{5.3.19}$$

$$V_{i3} = tr\left[\Psi_i \Lambda_i^{-1} \Psi_i^T \right] \tag{5.3.20}$$

with

$$\Psi_i = L_i^{*^{-1}} - L_i^{-1} \tag{5.3.21}$$

$$\dot{\Psi}_i = -\left(B_{mi}^T P_i x_i \right) \left(K_i x_i \right)^T L_i^T \Lambda_i \tag{5.3.22}$$

Using the properties of the *trace* operation

$$\begin{aligned} &tr(A) = tr\left(A^T\right) \\ &tr(A+B) = tr(A) + tr(B) \\ &tr\left(Ayx^T\right) = x^T Ay \end{aligned} \tag{5.3.23}$$

the matching conditions (5.3.6)–(5.3.7) and the Lyapunov equation (5.3.12), the time derivatives of V_{i1}, V_{i2}, and V_{i3} along with the solutions of (5.3.13), (5.3.10), and (5.3.22) respectively, are given by

$$\dot{V}_{i1} = -x_i^T Q_i x_i + 2x_i^T P_i B_i \left(L_i K_i - L_i^* K_i^* \right) x_i + 2x_i^T P_i \sum_{\substack{j=1 \\ j \neq i}}^{N} A_{ij} x_j$$

$$\dot{V}_{i2} = -2x_i^T P_i B_i \left(L_i K_i - L_i^* K_i^* \right) x_i$$

$$\dot{V}_{i3} = -2x_i^T P_i B_i \left(L_i - L_i^* \right) \left(K_i x_i \right)$$

Thus, the time derivative of V can be expressed as

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{V} = \sum_{\substack{j=1 \\ j\neq i}}^{N} \delta_i \left[-x_i^T Q_i x_i + 2x_i^T P_i \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij} x_j \right] \tag{5.3.24}$$

For the symmetric positive-definite matrix Q_i, we have

$$q_i \|x_i\|^2 \leq x_i^T Q_i x_i \leq \max \lambda\left(Q_i\right) \|x_i\|^2 \tag{5.3.25}$$

with q_i as defined in (5.3.14). Clearly, in the absence of interconnections, the existence of the Lyapunov function V in (5.3.17) assures global stability in the $\{ x, K, \Psi \}$ space, with $K = \text{diag}\,(K_i)$, $\Psi = \text{diag}\,(\Psi_i)$. However, since our interest is in the parameter errors $\tilde{L}_i \equiv L_i - L_i^*$ and not $\Psi_i = L_i^{*^{-1}} - L_i^{-1}$, then only uniform stability is implied in the $\{x, K, \tilde{L}\}$ space. The function V in (5.3.17) is not radially unbounded in this latter space.

Considering now the interconnections among subsystems, let us use (5.3.14), (5.3.15), and (5.3.25)to obtain the following inequality for V:

$$\dot{V} \leq \sum_{i=1}^{N} \delta_i \left[q_i \|x_i\|^2 - \|x_i\| \sum_{\substack{j=1 \\ j\neq i}}^{N} 2a_{ij} \|x_j\| \right] = -\bar{x}^T S_a \bar{x} \tag{5.3.26}$$

where

$$\bar{x} = \left[\|x_1\| \cdots \|x_N\|\right]^T \tag{5.3.27}$$

$$S_a = \begin{bmatrix} \delta_1 q_1 & -2\delta_1 a_{12} & \cdots & -2\delta_1 a_{1N} \\ -2\delta_2 a_{21} & \delta_2 q_2 & & \vdots \\ \vdots & & & \\ -2\delta_N a_{N1} & \cdots & & \delta_N q_N \end{bmatrix} \tag{5.3.28}$$

Since

$$\bar{x}^T S_a \bar{x} = \bar{x}^T \frac{\left(S_a^T + S_a\right)}{2} \bar{x} = \bar{x}^T S \bar{x}$$

then, (5.3.26) can be expressed as

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{V} \leq -\bar{x}^T S \bar{x} \tag{5.3.29}$$

where matrix S is

$$S = \begin{cases} \delta_i q_i & i = j \\ -\left(\delta_i a_{ij} + \delta_j a_{ji}\right) & i \neq j \end{cases} \tag{5.3.30}$$

If S is positive-definite, then $\dot{V}$ is negative-semidefinite. Again, the solutions $x_i(t)$, $K_i(t)$, and $L_i(t)$ to (5.3.13), (5.3.10), and (5.3.11), respectively, are bounded when $K_i(t_o)$ and $L_i(t_o)$ are sufficiently close to their desired values K_i^* and L_i^*

Using Barbalat's lemma (Benitez-Read, 1993) we know from (5.3.29) and the boundedness of $x(t)$ that $\ddot{V}(t)$ is bounded. Hence $\dot{V}(t)$ is uniformly continuous. Since $V(t)$ is a nonincreasing function of time and is bounded from below, it converges to a finite value $V\infty$. Thus, $\lim_{t\to\infty}\int_0^t \dot{V}\,dt = V_\infty - V_o <$ ∞ and hence, $\lim \dot{V}(t) = 0$ as $t \to \infty$, that is, $\lim \|x(t)\| = 0$ as time progresses. Q.E.D. ■

5.3.3 Decentralized Tracking Systems

For the tracking problem, the local control inputs are determined by

$$u_i(t) = L_i(t)K_i(t)x_i(t) + L_i(t)r_i(t) \tag{5.3.31}$$

with the adaptive laws for the feedback and feedforward gain matrices K_i and L_i given by

$$\dot{K}_i = -\left(B_{mi}^T P_i e_i\right)x_i^T \Gamma_i - \sigma_i K_i \Gamma_i \tag{5.3.32}$$

$$\dot{L}_i = -L_i\left(B_{mi}^T P_i e_i\right)\left(K_i x_i + r_i\right)^T L_i^T \Lambda_i L_i \tag{5.3.33}$$

where

$$e_i = x_i - x_{mi} \tag{5.3.34}$$

and σ_i is a design positive scalar parameter. Using the matching conditions (5.3.6) and (5.3.7), the state error dynamics is obtained as

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{e}_i = A_{mi}e_i + B_i\left(L_iK_i - L_i^*K_i^*\right)x_i + B_i\left(L_i - L_i^*\right)r_i + \sum_{\substack{j=1\\ j\neq i}}^{N} A_{ij}x_i \tag{5.3.35}$$

A persistent input due to the interconnections acts as a disturbance in the overall error equation and therefore the solution $e(t)$ may not converge to or may not even possess an equilibrium. The following theorem establishes sufficient conditions for boundedness and convergence of the state error to a residual set (Benitez-Read, 1993).

Theorem 5.6 If the matrix S defined by (5.3.16) is positive-definite and the initial parameter values $K_i(t_o)$ and $L_i(t_o)$ are sufficiently close to their desired values K_i^* and L_i^*, respectively, then the solution to (5.3.32), (5.3.33), and (5.3.35) is ultimately bounded. Furthermore, there exist finite nonnegative constants T and c_o such that for all $t \geq T$ the solution $e_i(t)$ is inside the set

$$D_o = \left\{ e,\, K_i : \left[\frac{\lambda_S}{2}\|e\|^2 + \sum_{i=1}^{N} \delta_i\sigma_i \left\|K_i - K_i^*\right\|^2 \right] \leq \left(1 + c_o\right)d_o \right\} \tag{5.3.36}$$

where

$$d_o = \frac{b_o^2}{2\lambda_S}\left\|x_m\right\|^2 + \sum_{i=1}^{N} \delta_i\sigma_i\left\|K_i^*\right\|^2 \tag{5.3.37}$$

$\lambda_s = \min \lambda(S)$, and b_o is a positive constant.

PROOF: Consider the positive-definite function

$$V = \sum_{i=1}^{N} \delta_i\left[V_{i1} + V_{i2} + V_{i3}\right] \tag{5.3.38}$$

where

$$V_{i1} = e_i^T P_i e_i \tag{5.3.39}$$

$$V_{i2} = tr\left[\left(K_i - K_i^*\right)\Gamma_i^{-1}\left(K_i - K_i^*\right)^T\right] \tag{5.3.40}$$

$$V_{i3} = tr\left[\Psi_i \Lambda_i^{-1} \Psi_i^T\right] \tag{5.3.41}$$

with matrix Ψ_i and its dynamic equation given by

$$\Psi_i = L_i^{*^{-1}} - L_i^{-1} \tag{5.3.42}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\dot{\Psi}_i = -\left(B_{mi}^T P_i e_i\right)\left(K_i x_i + r_i\right)^T L_i^T \Lambda_i \tag{5.3.43}$$

with P_i satisfying (5.3.12). Using the trace operation properties (5.3.23), the matching conditions (5.3.6)–(5.3.7), and the Lyapunov equation (5.3.12), derivatives of V_{i1}, V_{i2}, and V_{i3}, along with the solutions of (5.3.35), (5.3.32), and (5.3.43), respectively, are given by

$$\dot{V}_{i1} = -e_i^T Q_i e_i + 2e_i^T P_i B_i\left(L_i K_i - L_i^* K_i^*\right)x_i + \\ 2e_i^T P_i B_i\left(L_i - L_i^*\right)r_i + 2e_i^T P_i \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij} x_j$$

$$\dot{V}_{i2} = -2e_i^T P_i B_i\left(L_i K_i - L_i^* K_i^*\right)x_i - 2tr\left[\sigma_i\left(K_i - K_i^*\right)K_i^T\right]$$

$$\dot{V}_{i3} = -2e_i^T P_i B_i\left(L_i - L_i^*\right)\left(K_i x_i + r_i\right)$$

The time derivative of V can then be expressed as

$$\dot{V} = \sum_{i=1}^{N} \delta_i \left[-e_i^T Q_i e_i + 2e_i^T P_i \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij}\left(e_j + x_{mj}\right) - 2tr\left[\sigma_i\left(K_i - K_i^*\right)K_i^T\right] \right] \tag{5.3.44}$$

Let us now define the following vectors:

$$\bar{e} = \left[\|e_1\| \cdots \|e_N\|\right]^T \tag{5.3.45}$$

$$\bar{x}_m = \left[\|x_{m1}\| \cdots \|x_{mN}\|\right]^T \tag{5.3.46}$$

Now, proceeding as in the regulation problem to obtain the structural condition on the composite system, and using the inequality

$$-2tr\left[\left(K_i - K_i^*\right)K_i^T\right] \le -\left\|K_i - K_i^*\right\|^2 + \left\|K_i^*\right\|^2 \tag{5.3.47}$$

the following inequality is obtained for $\dot{V}$:

$$\dot{V} \le -\bar{e}^T S \bar{e} + b_o \|e\| \cdot \|x_m\| - \sum_{i=1}^{N} \delta_i \sigma_i \left\|K_i - K_i^*\right\|^2 + \sum_{i=1}^{N} \delta_i \sigma_i \left\|K_i^*\right\|^2 \tag{5.3.48}$$

where b_o is a finite constant which depends on the norms of P_i, A_{ij}, and δ_i,

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$i = 1, \ldots, N$. Noting that

$$
\begin{aligned}
& -e^T S e + b_o \|e\| \|x_m\| \le \\
& -\lambda_S \|e\|^2 + b_o \|e\| \|x_m\| + \left[\sqrt{\frac{\lambda_S}{2}} \|e\| - \frac{b_o \|x_m\|}{\sqrt{2\lambda_S}} \right]^2 = \\
& -\frac{\lambda_S}{2} \|e\|^2 + \frac{b_o^2}{2\lambda_S} \|x_m\|^2
\end{aligned}
\tag{5.3.49}
$$

where $\lambda_S = \min \ \lambda(S)$, we can now express (5.3.48) as

$$
\begin{aligned}
\dot{V} \le & \frac{\lambda_S}{2} \|e\|^2 - \sum_{i=1}^{N} \delta_i \sigma_i \|K_i - K_i^*\|^2 + \\
& \frac{b_o^2}{2\lambda_S} \|x_m\|^2 + \sum_{i=1}^{N} \delta_i \sigma_i \|K_i^*\|^2
\end{aligned}
\tag{5.3.50}
$$

In view of (5.3.38) and (5.3.50), the solutions $e(t)$, $K_i(t)$, $L_i(t)$ are uniformly ultimately bounded (Praly, 1983) for all initial conditions $K_i(t_o)$, $L_i(t_o)$ lying in some bounded neighborhood about their matching condition values K_i^*, L_i^*. Further, since $\dot{V} < \varepsilon(c_o) < 0$ for e, K_i outside D_o with $c_o \ge 0$, then for some finite constants T and c_o, the solution $e(t)$, $K_i(t)$ remains inside D_o for all $t > T$. Q.E.D. ■

The use of σ_i has been found to be useful in obtaining sufficient conditions for boundedness in the presence of unmodeled interactions. In the absence of such interactions, that is, when each subsystem is decoupled, the design parameters $\sigma_i > 0$ cause nonzero state errors. This is a tradeoff between boundedness of all signals in the presence of the subsystem's interactions and the loss of exact convergence of the errors to the origin in their absence.

5.3.4 Liquid-Metal Cooled Reactor

Although applications of large-scale systems are treated in much more detail in Chapter 8, this section introduces the model and some initial simulation results of a liquid-metal cooled reactor (LMR) of a nuclear power system. The control approaches considered here are the linear optimal regulator and model-reference adaptive control, while its decentralized adaptive control will be treated in Chapter 8. Much of the material in this section is from (Benitez-Read *et al.*, 1992).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Consider the LMR module of a nuclear reactor shown in Figure 5.3. Each reactor module consists of an LMR reactor with the corresponding primary and intermediate heat transport loops, a recirculating steam generator, and steam drum. All the modules are connected to a common steam heater that feeds the turbine.

The equations describing the dynamic behavior of a reduced model (Rovere, 1989) of the plant under study, namely the reactor core and primary heat transport loop, are given by (Benitez-Read and Jamshidi, 1992).

$$
\begin{aligned}
\dot{x}_1 &= \left(rho0 + afuel \cdot x_3 + acool \cdot x_4\right) \cdot bele \cdot x_1 + bele \cdot x_2 + \left(rhoc \cdot bele \cdot x_1\right) \cdot u_1 \\
\dot{x}_2 &= \lambda \cdot \left(x_1 - x_2\right) \\
\dot{x}_3 &= thn \cdot x_1 + thf \cdot \left(x_4 - x_3\right) \\
\dot{x}_4 &= thc \cdot \left(x_3 - x_4\right) + ttc \cdot \left(x_7 - x_4\right) \cdot u_2 \\
\dot{x}_5 &= ttphL \cdot \left(x_4 - x_5\right) \cdot u_2 \\
\dot{x}_6 &= \frac{haps}{mpx \cdot cpna}\left[\frac{a_o}{tr} + \frac{a_1}{tr \cdot \left[\log\left(pr \cdot K_8\right) + a_2\right]} - x_6\right] + ttpx \cdot \left(x_5 - x_6\right) \cdot u_2 \\
\dot{x}_7 &= ttcL \cdot \left(x_6 - x_7\right) \cdot u_2
\end{aligned}
\tag{5.3.51}
$$

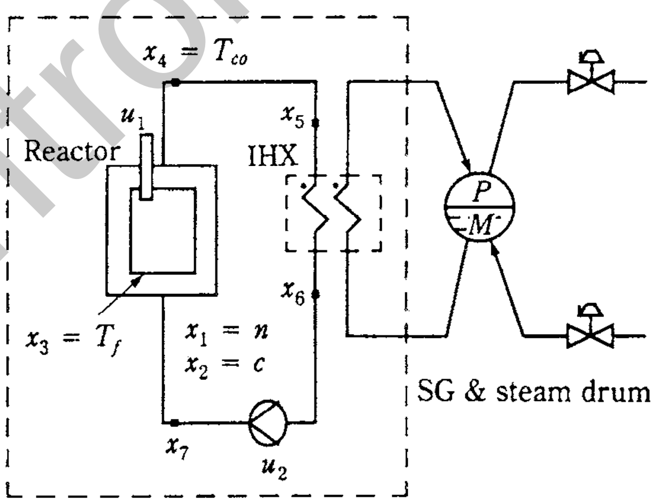


Figure 5.3. Liquid-metal cooled reactor (LMR) module.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where the state variables of the system are

$x_1 = n$ = Neutron power (normalized to 1 for full power),

$x_2 = c$ = Delayed neutron precursor concentration (normalized to 1 for full power),

$x_3 = \dfrac{T_f}{T_r}$ = Fuel temperature to reference temperature ratio,

$x_4 = \dfrac{T_{co-\mathrm{out}}}{T_r}$ = Core coolant outlet temperature to reference temperature ratio,

$x_5 = \dfrac{T_{p-\mathrm{in}}}{T_r}$ = IHX primary inlet temperature to reference temperature ratio,

$x_6 = \dfrac{T_{p-\mathrm{out}}}{T_r}$ = IHX primary outlet temperature to reference temperature ratio,

$x_y = \dfrac{T_{co-\mathrm{in}}}{T_r}$ = Core inlet temperature to reference temperature ratio.

The control inputs are

$u_1 = u_{rod}$ = Control rod position,

$u_2 = u_{pfp}$ = Primary pump fractional flow.

The control input signals are scaled so that $0 \le u_i \le 1$ with 0 meaning inserted, minimum, or closed, and 1 withdrawn, maximum or open.

Notation: The SI unit system was adopted in this work. Units for variables used herein are as follows (otherwise as noted):

Mass:	kg	Pressure:	MPa
Density:	kg/m^3	Temperature:	°C
Power:	Joule/s	Flowrate:	kg/s
Volume:	m^3	Enthalpy:	Joule/kg

Model constants:

alpf: Fuel reactivity coefficient [$°C^{-1}$]

alpc: Coolant reactivity coefficient [$°C^{-1}$]

betat: Total effective delayed neutron yield per fission

*rho*0*:* Initial core reactivity (in units of *betat,* no control rod, no temp. feedback)

rhoc: Total control reactivity worth (in units of *betat)*

ele: Prompt neutron generation time [s^{-1}]

λ: Precursor decay constant [s^{-1}]

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

hafc:	Heat transfer coefficient between fuel and coolant times contact area
*pow*0:	Initial reactor thermal power
*pres*0:	Initial steam drum pressure
*msd*0:	Initial steam drum water mass
powr:	Reference power
pr:	Reference pressure
msdr:	Reference mass
tr:	Reference temperature
hr:	Reference enthalpy
ftp:	Neutron flux to thermal power factor
mphL:	Primary hot leg coolant mass
mpcL:	Primary cold leg coolant mass
mcc:	Core coolant mass
mpx:	IHX primary coolant mass (per node)
mf:	Fuelmass
vpri:	Primary pump proportionality constant
ctna:	Sodium specific heat
ctfuel:	Fuel specific heat

The constant coefficients of the nonlinear differential equations are given by

$$\left.\begin{array}{lll}
rhoc = 7.0, & cpna = 1283.2, & a_0 = -230.45 \\
rho0 = -3.2565, & msdr = 60000, & a_1 = -3892.7 \\
\lambda = 0.01, & haps = 5943329.1 & a_2 = -9.48654 \\
tr = 282.323, & pr = 6.65298, & K_8 = \dfrac{pres0}{pr} \\
mpx = 43000. & & \\
afuel = \dfrac{alpf - tr}{betat}, & thc = \dfrac{hafc}{mccc} & \\
acool = \dfrac{alpc - tr}{betat}. & ttc = \dfrac{vpri}{mphL} & \\
bele = \dfrac{betat}{ele}, & ttphL = \dfrac{vpri}{mpx} & \\
thn = \dfrac{ftp}{mcf \cdot tr}, & ttpx = \dfrac{vpri}{mpx} & \\
thf = \dfrac{hafc}{mcf}, & ttpcL = \dfrac{vpri}{mpcL} &
\end{array}\right\} \quad (5.3.52)$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where

$$\begin{array}{lll} alpf = -15.0e-06, & hafc = 5.195122e+06, & mcf = mf \cdot cpfuel, \\ alpc = 7.0e-06, & vpri = 3235.7, & mccc = mcc \cdot cpna, \\ betat = 4.0e-03, & mcc = 10442, & mf = 30000, \\ ele = 4.2e-06, & mphL = 130000, & cpfuel = 350, \\ ftp = 4.26e+08, & mpcL = 130000, & pres0 = 6.65398 \end{array}$$

The equilibrium operating values of the plant state and control input variables are

$$\left.\begin{array}{ll} x_{1eq} = \dfrac{pow0}{ftp}, & x_{6eq} = \dfrac{354}{tr} \\ x_{2eq} = x_{1eq}, & x_{7eq} = x_{6eq} \\ x_{3eq} = \dfrac{550}{tr}, & u_{1eq} = 0.5 \\ x_{4eq} = \dfrac{468}{tr}, & u_{2eq} = 0.9 \\ x_{5eq} = x_{4eq,} & pow0 = 4.26e+08 \end{array}\right\}$$

Linear Model A linear model of the reactor core and primary heat transport loop is obtained from the nonlinear differential equations by taking the first-order terms of the Taylor series expansion about the equilibrium operating values (5.3.52). Thus, the system can be represented by

$$\dot{x} = Ax + Bu \tag{5.3.53}$$

where $A = [a_{ij}]$ and $B = [b_{ij}]$ with the nonzero elements of A and B being computed as

$$\left.\begin{array}{lll} a_{11} = \left(afuel \cdot x_{3eq} + acool \cdot x_{4eq} + rhoc \cdot u_{1eq} + rho0\right) \cdot bele & & \\ a_{12} = bele, & a_{13} = afuel \cdot bele \cdot x_{1eq}, & a_{14} = acool \cdot bele \cdot x_{1eq} \\ a_{21} = \lambda, & a_{22} = -\lambda, & a_{31} = thn \\ a_{33} = -thf, & a_{34} = thf, & a_{43} = thc \\ a_{44} = -ttc \cdot u_{2eq} - thc, & a_{47} = ttc \cdot u_{2eq}, & a_{54} = ttphL \cdot u_{2eq} \\ a_{55} = -ttphL \cdot u_{2eq}, & a_{65} = ttpx \cdot u_{2eq} & \\ a_{66} = -ttpx \cdot u_{2eq} - \dfrac{haps}{mpx \cdot cpna}, & & a_{76} = ttpcL \cdot u_{2eq} \\ a_{77} = -ttpcL \cdot u_{2eq} & b_{11} = rhoc \cdot bele \cdot u_{1eq} & \\ b_{42} = ttc \cdot \left(x_{7eq} - x_{4eq}\right), & b_{62} = ttpx \cdot \left(x_{5eq} - x_{6eq}\right) & \end{array}\right\} \tag{5.3.54}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Once the model of this reactor subsystem is available, we can further proceed with the application of different control algorithms which result in a closed-loop system with either a specified performance function being optimized or a desired output or state trajectory being tracked. The design of a linear quadratic optimal control and some simulation results on the linear and nonlinear models are presented in the next section. Likewise, a model reference adaptive control scheme applied to the linear model of the system is investigated later in this and the next section.

Linear Quadratic Regulator Consider the linear time-invariant (LTI) model of the system described in (5.3.53)–(5.3.54) with some assumed initial state $\rightarrow x(t_o)$. Let the constant matrices Q and R be nonnegative and positive-definite, respectively. Define the performance index as

$$J(x,u) = \int_{t_0}^{\infty} \left(x^T Q x + u^T R u\right) dt \tag{5.3.55}$$

and the minimization problem as the task of finding an optimal control $u^*(\cdot)$ which minimizes J. To ensure solvability of the problem, the pair (A, B) is required to be stabilizable.

The optimal control at time t, when the initial time is arbitrary, is uniquely given by the control law

$$u^*(t) = -R^{-1} B^T K x(t) = P x(t) \tag{5.3.56}$$

where matrix K is the solution to the algebraic matrix Riccati equation (AMRE)

$$A^T K + KA - KBR^{-1} B^T K + Q = 0 \tag{5.3.57}$$

Simulation Results The state and control weighting matrices were both chosen to be the identity matrix with proper dimensions. The dynamic behavior of the closed-loop system with the plant taken to be the linear model of the reactor core and primary heat trasnport loop of the LMR module is shown in Figures 5.4 and 5.5. The control law is given by

$$u = Px + r \tag{5.3.58}$$

where $r(t)$ represents the reference input signal to the closed-loop system and it is used here to introduce standard perturbations into the system. The simulation conditions were set as follows: 1) The initial condition of the system was chosen to be $x_1 = 0.1,\ 1 \le i \le 7$; 2) the state matrix A stays at its nominal value during the first 10 seconds of simulation; then it is

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

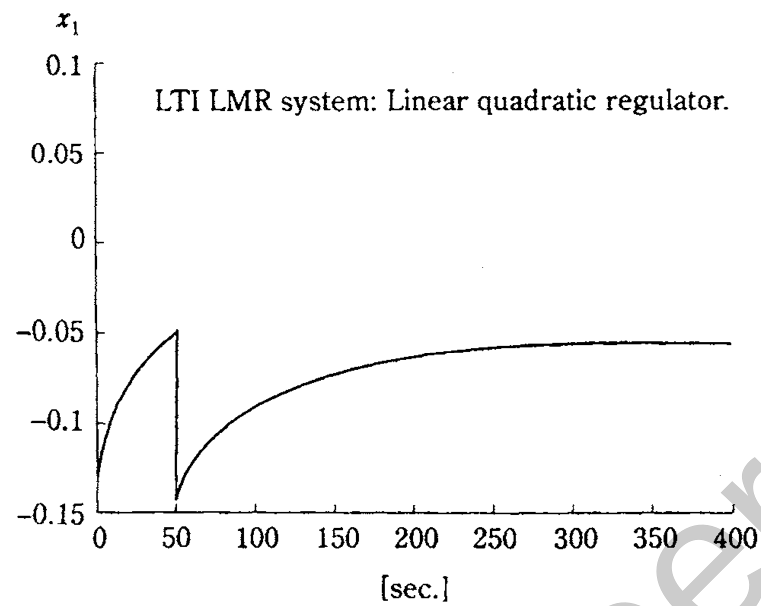


Figure 5.4. Neutron power.

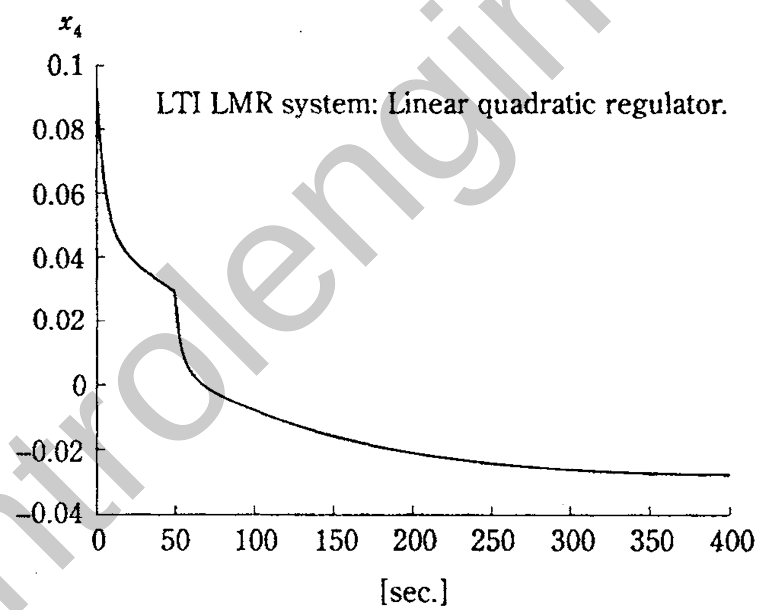


Figure 5.5. Core coolant outlet temperature.

controlengineers.ir

changed to 5% above the nominal, and remains there until the end of the simulation; 3) the reference input has a zero value during the first 50 seconds of the simulation; then it is step changed to $[-0.1 \quad 0]^T$ and stays there for the rest of the simulation.

The state variable x_1, which is related to the neutron power and is shown

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

in Figure 5.4, appears to be very sensitive to the input perturbation $r_1(t)$. This is expected since $r_1(t)$ is one of the additive terms of $u_1(t)$; hence, it represents an insertion of reactivity, negative in this case, into the plant, causing the neutron power to likewise suffer a sudden change. After this transient, a lower steady-state value is quickly and smoothly approached. The state variable x_4, related to the core coolant outlet temperature, is shown in Figure 5.5. The effect of the input perturbation on x_4 takes place indirectly, through the state feedback, and therefore, this variable approaches its steady-state value without experiencing abrupt changes. The variation on the value of state matrix A does not have a significant effect on the system response, thus showing by simulation, closed-loop stability against parametric perturbation.

The closed-loop system response when the optimal control input is applied to the nonlinear model of the plant is shown in Figures 5.6 and 5.7. The initial condition is $x = x_{eq}$ as given by Equation (2.3), i.e., the equilibrium values of the plant state. The control law is given by

$$u = K(x - x_{eq}) + r \tag{5.3.59}$$

with

$$r = \begin{pmatrix} -0.1 & 0 \end{pmatrix}^T + u_{eq} \tag{5.3.60}$$

The step change on $r(t)$ at $t = 0$ below the equilibrium u_{eq} represents an

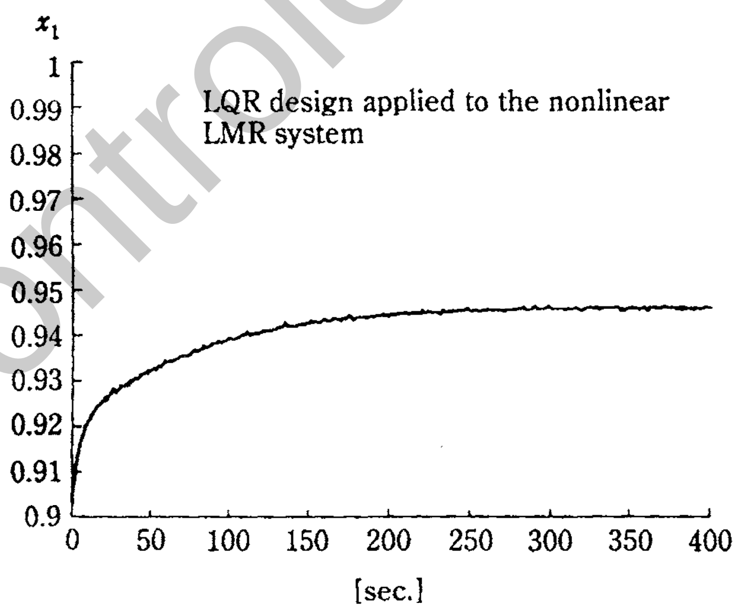


Figure 5.6. Neutron power.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

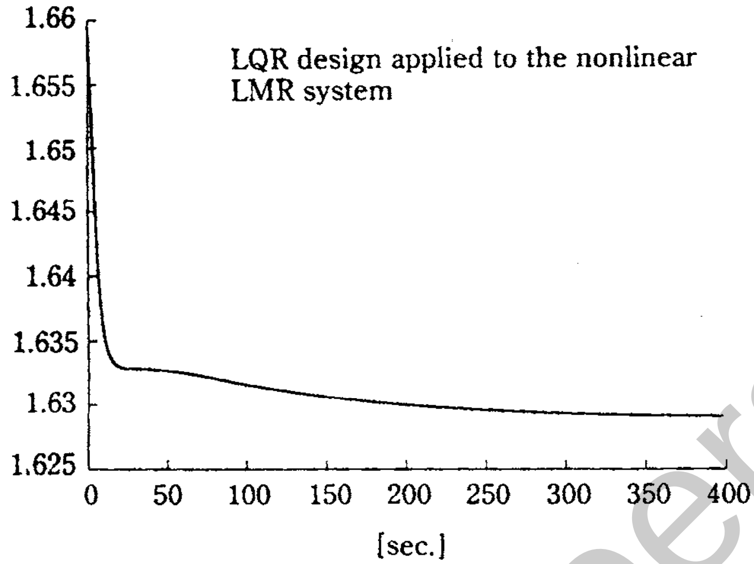


Figure 5.7. Core coolant outlet temperature.

insertion of the control rod (control input u_1), which in turn introduces a negative reactivity step into the system. This causes the neutron power to have a sudden reduction, shown in Figure 5.6, followed by a smooth approach to a lower steady-state value. The core coolant outlet temperature is less sensitive to this reactivity perturbation and shows a gradual response at all times, reaching also a lower steady-state value. These results show the controlling action of the linear quadratic optimal regulator applied to the nonlinear LMR module subsystem when the operating conditions are close to the equilibrium point.

5.3.5 Application of Model Reference Adaptive Control

We will now consider the case where the linear model of the system is regarded as a good approximation to the actual plant about the given baseline operating point, but with some uncertainty on the parameters, that is, on the matrix entry values. The theory of adaptive control treats these cases in different ways; thus, the approach used in this work is the direct method of the model reference adaptive control (MRAC) scheme (Narendra and Annaswamy, 1989) with the state variables of the plant accessible to the designer.

Direct method of the MRAC The plant to be adaptively controlled is described by the differential equation

$$\dot{x} = Ax + Bu \tag{5.3.61}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where *(A, B)* is controllable, and A and B are unknown constant matrices. The n-dimensional state x is assumed to be accessible. A reference model is specified by the stable LTI differential equation

$$\dot{x}_M = A_M x_M + B_M r \tag{5.3.62}$$

where A_M is an asymptotically stable matrix and $r(\cdot)$ is a bounded reference input. It is assumed that $x_M(t)$ represents a desired trajectory for $x(t)$ to follow. The aim is to determine a method for controlling the plant so that

$$\lim_{t\to\infty} e(t) \equiv \lim_{t\to\infty}\left[x(t) - x_M(t)\right] = 0 \tag{5.3.63}$$

The case considered here is the one where the elements of matrix A are constant but unknown while those of matrix B are constant and known. The matrix B_M of the reference model, which is at the discretion of the designer, can be chosen to be identical to B, or alternately, to be equal to BQ^*, where Q^* is a known constant matrix (see Figure 5.8).

Let the control u be given by

$$u(t) = \Theta(t)x(t) + Q^* r(t) \tag{5.3.64}$$

where $\Theta(t)$ is an $m \times n$ matrix of adjustable parameters (m is the number of control inputs) and Q^* is the $m \times m$ constant matrix defined before. The error dynamics is expressed as

$$\dot{e} = A_M e + \left(A + B\Theta(t) - A_M\right)x \tag{5.3.65}$$

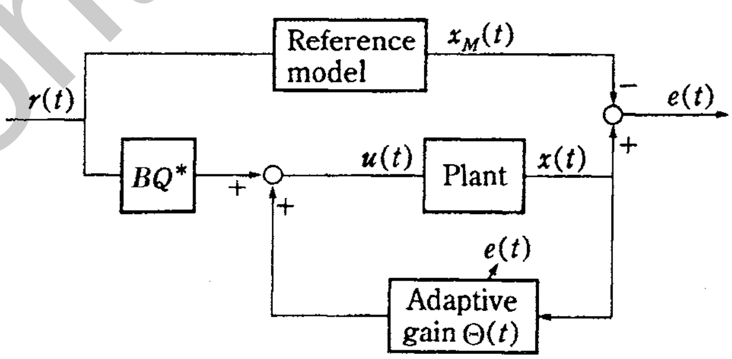


Figure 5.8. Model reference adaptive control (MRAC).

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Let us assume that a constant matrix Θ^* exists such that

$$A + B\,Q^* = A_M \tag{5.3.66}$$

then

$$\dot{e} = A_M e + B\left[\Theta(t) - \Theta^*\right]x \tag{5.3.67}$$

and the aim is to determine an adaptive law for adjusting $\Theta(t)$ so that $\lim_{t\to\infty} e(t) = 0$. One way of finding the adaptive law is by the Lyapunov stability analysis (Jamshidi, 1983).

Lyapunov Stability Analysis Let a candidate Lyapunov function be

$$V(e,\Phi) \equiv e^T P e + tr\left[\Phi^T \Phi\right] \tag{5.3.68}$$

where

$$\Phi \equiv \Theta - \Theta^* \tag{5.3.69}$$

and P is the unique symmetric positive definite solution to the matrix equation

$$A_M^T P + P A_M + Q_o = 0 \tag{5.3.70}$$

where $Q_o > 0$. The error dynamic equation can be now expressed as

$$\dot{e} = A_M e + B\Phi x \tag{5.3.71}$$

The next step is to examine the time derivative of the candidate Lyapunov function. Such derivative is given by

$$\dot{V} = e^T\left(A_M^T P + P A_M\right)e + 2x^T\Phi^T B^T P e + 2tr\left[\Phi^T\dot{\Phi}\right] \tag{5.3.72}$$

Using the identity $tr(ab^T) = b^T a$ where a and b are column vectors, we get

$$\dot{V} = -e^T Q_o e + 2tr\left(\Phi^T\left[B^T P e x^T + \dot{\Phi}\right]\right) \tag{5.3.73}$$

Thus, by letting the feedback matrix adaptive law being

$$\dot{\Phi} = \dot{\Theta} = -B^T P e x^T \qquad 1\times 2 \tag{5.3.74}$$

we have

$$\dot{V} = -e^T Q_o e \le 0 \tag{5.3.75}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which ensures that the function V is indeed a Lyapunov function for the system of equations given by Equations (5.3.71) and (5.3.74), and that $\lim_{t\to\infty} e(t) = 0$.

Simulation Results The design of the adaptive controller has been done for the linear model of the reactor core and primary heat transport loop of the LMR system, as follows:

a) The reference model is defined as

$$\dot{x}_M = A_M x_M + B_M r$$

where

$$A_M = A + B\Theta, \quad B_M = BQ^*, \quad Q^* = I$$

A and B are the nominal values of the state and control input matrices of the linear model of the plant, and Θ is the state feedback matrix obtained from the solution of the LQR problem (see Equations (5.3.51)–(5.3.54)).

b) Matrix Q_o in Equation (5.3.70) was chosen to be the identity matrix.

The results of the computer simulation are shown in Figures 5.9 and

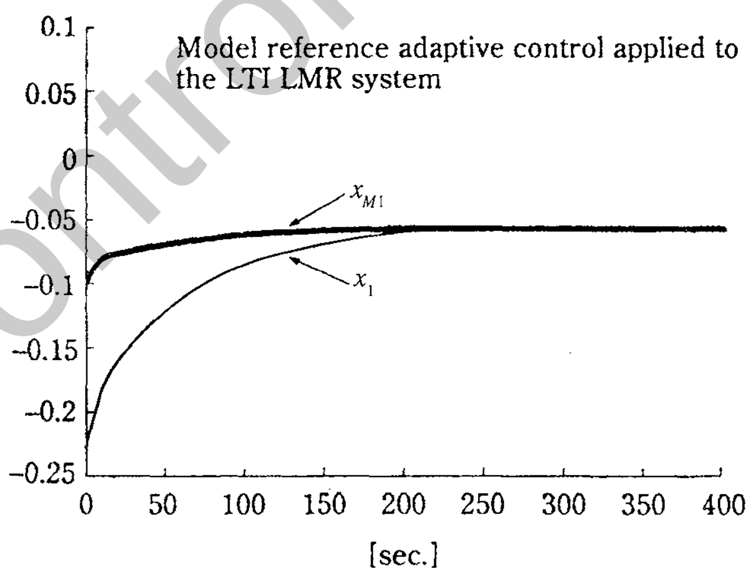


Figure 5.9. Neutron power.

controlengineers.ir

controlengineers.ir
@controlengineers

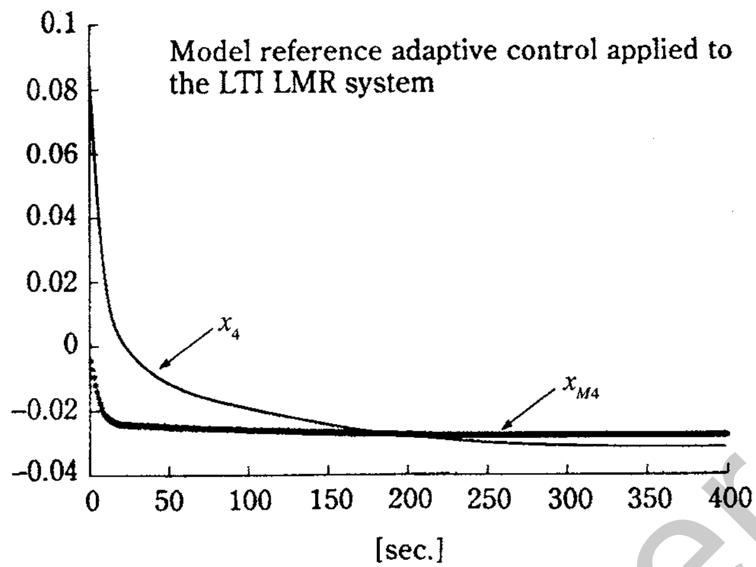


Figure 5.10. Core coolant outlet temperature.

5.10. The initial condition of the plant is chosen to be $x_i = 0.1$, $1 \le i \le 7$, whereas that of the reference model system is set to zero. A step from 0 to –0.1 at $t = 0$ [sec.] is applied to the reference input $r_1(t)$, while $r_2(t)$ remains at zero during the entire simulation. The purpose of applying this input is to analyze the dynamic behavior of the plant under this external reactivity perturbation. Regarding the parametric uncertainty of the LMR module subsystem, the plant state matrix A has its nominal value during the first 10 seconds of simulation, changing afterwards to a value 5% above the nominal.

The neutron power of both the plant and the reference model is shown in Figure 5.9. The end of a step change of the neutron power is due to the negative step insertion of reactivity into both systems. After this transient, the plant neutron power follows closely the reference system neutron power to a steady-state operating point. The state variables representing the core coolant outlet temperature of the plant and reference model systems are shown in Figure 5.10. The negative insertion of reactivity due to the input $r_1(t)$ does not affect directly the dynamics of these variables, but rather indirectly through the state feedback. Thus, the plant state follows the reference system state to a steady-state operating point, without showing any severe transient. The parametric variation imposed on the plant state matrix A does not appreciably affect the dynamic response of the plant. In general, good signal tracking performance obtained with this control scheme can be deduced from the simulation results.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

5.4 Discussion and Conclusions

Two important problems within the context of large-scale systems' decentralization property were considered in this chapter. These problems were decentralized stabilization and decentralized adaptive control. They are only two topics in large-scale systems design and synthesis. The primary motivations for these new developments are twofold. One is that straight application of centralized techniques, such as pole placement, identification, estimation, control, optimization, etc., are inappropriate due to the physical and natural characteristics of large-scale systems. The other is the vast number of areas into which the decentralized or hierarchical structures of large-scale systems fall naturally. These areas of applications, which include economics, education, urban systems, transportation, power, energy, environment (including water resources systems), pose challenging problems for several years to come.

A usual condition for centralized stabilization of a system is the controllability and observability of the modes. The decentralized version of this, as discussed in Section 5.3.1, requires that the "fixed modes" of the systems be contained on the left-half s-plane. Following this important development, decentralized stabilization can be effectively and systematically achieved through a number of computational algorithms. In Section 5.2, two decentralized stabilization schemes were discussed and two algorithms were presented.

A typical decentralized stabilization approach based on multilevel control was given in Section 5.2.2. When a large-scale system is described in input-decentralized form, either by physical decomposition or input decentralization, the proposed control stabilizes the system through connective Lyapunov stability (see Chapter 3). In this scheme, each subsystem is stabilized through local controllers, while the effects of interconnections among subsystems are reduced through global controllers.

Another multilevel control scheme based on exponential stabilization of the linear state regulator was presented in Section 5.2.3. It was shown that a large-scale system in its input-decentralized form can be stabilized with a prescribed degree of stability if the system's interconnection pattern satisfies certain symmetry conditions (Theorem 5.3) or admits some norm bounds (Theorem 5.4). If the prescribed conditions are not satisfied by the existing interconnections, a higher-level controller can be used. Another decentralized stabilization scheme along the lines of reducing the influence of interconnections in an attempt to give more reliable decentralized controls is due to Huang and Sundareshan (1980). The scheme makes a modification of the objective function based on the maximum interconnection matrix leading to a modified linear state regulator for each

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

subsystem similar to the exponential stabilization scheme.

In Section 5.3, a decentralized adaptive control scheme was presented for a class of large-scale systems. The system was composed of interconnected linear subsystems with multiple inputs, in which their state, control input matrices, and the strength of the interconnections are unknown.

The control law for each subsystem, based on local adaptive state feedback, and the adaptive laws of the feedback and feedforward gain matrices, were developed for the stabilization and tracking problems. Sufficient conditions in the form of algebraic constraints were obtained which guarantee asymptotic regulation of the plant states. Moreover, in the case of reference trajectory tracking, the proposed control structure achieves boundedness of all closed loop system signals, as well as convergence of the plant states to a residual set, under certain structural perturbations.

The decentralized adaptive control scheme reduces the number of parameters to be dynamically adjusted through the adaptive laws, compared to the centralized case. In the latter, this number is given by the product of the total number of the system inputs and the sum of the number of the system's states and inputs. For the decentralized case this figure is computed as the sum over the number of subsystems of the above product applied separately to each subsystem. Furthermore, the gains of the adaptive controller parameters are generally of lower magnitude than those required in the nonadaptive decentralized control design. The gains are adjusted to the levels necessary to bring the state errors to the origin.

Within the framework of MRAC a liquid-metal reactor model was used in two ways. One of the control approaches is based on the linear quadratic regulator (LQR) problem. The controller design has been applied to the linear and nonlinear models of the system under consideration. The results show closed loop stability and relatively fast response to perturbations as well as low system sensitivity to parametric variation of the state matrix in the linear case. In the latter case, if signal tracking is desired, one possible approach is to solve the linear quadratic servomechanism problem, assuming certain system requirements are met. Another control scheme reported was the control of the system using the model reference adaptive control (MRAC) method. The simulation is performed on a linear model of the reactor which experiences parameter variation on the plant state matrix. Good tracking performance and stability of the overall system, to which input perturbation and parameter variations were applied, were observed from the results. Decentralized adaptive control is applied to this reactor in Chapter 8. Unfortunately, constraints on the length of this book did not allow other important decentralized control problems, such as robust servomechanism, uncertain systems, etc., to be addressed here.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Problems

5.1. Determine the fixed modes of the following system:

$$\dot{x} = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} x$$

Is the system stabilizable under decentralized control?

5.2. Find a decentralized stabilizing output controller for the system

$$\dot{x} = \begin{bmatrix} 1 & 0 \\ 0.5 & -1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x$$

5.3. Consider the following two subsystems of an interconnected system:

$$(A_1, B_1, C_1) = \left(\begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 1 \\ -1 & -2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \right)$$

$$(A_2, B_2, C_2) = \left(\begin{bmatrix} 2 & 1 \\ -2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix} \right)$$

Use Algorithm 5.2 to stabilize the system.

5.4. CADLAB Consider a three-subsystem problem

$$\dot{x}_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0.5 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1, \quad y_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} x_1$$

$$\dot{x}_2 = -0.5 x_2 + u_2, \quad y_2 = x_2$$

$$\dot{x}_3 = \begin{bmatrix} 1 & -0.2 \\ 1 & -1 \end{bmatrix} x_3 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_2, \quad y_3 = \begin{bmatrix} 1 & 1 \end{bmatrix} x_3$$

Use Algorithm 5.2 to find a decentralized stabilizing control for the system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

5.5. Consider the stabilization of a large-scale multivariable linear system $\dot{x} = Ax + Bu$ where the control vector consists of a local and a global control component, i.e., $u = u^l + u^g$. Use the matrix Riccati formulation and discussions in Section 5.2.3 to find an exponentially stabilizing control.

5.6. CADLAB For the system

$$\dot{x} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0.5 \\ \hline 0.1 & 0.1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 2 & -3 \end{array}\right] x + \left[\begin{array}{c|c} 0 & 0 \\ 1 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{array}\right] u$$

use Algorithm 5.2 to find a two-level stabilizing controller. Use $\lambda\left\{\hat{A}_1\right\} = \{-1, -2\}$ and $\lambda\left\{\hat{A}_2\right\} = \{-5 \pm j, -10\}$ as the desired poles for the two subsystems.

5.7. Write a piece of code to implement MRAC in MATLAB or your favorite language.

5.8. CADLAB Design a MRAC controller similar to Section 5.3.5 for the system

$$\left(A_M, B_M\right) = \left(\begin{bmatrix} 0 & 1 \\ -4 & -4 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

and a plant model

$$\left(A, B\right) = \left(\begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 6

Near-Optimum Design of Large-Scale Systems

6.1 Introduction

One of the most significant motivations for the development of new design techniques suitable for large-scale systems has been the computational impracticality of direct application of optimal control or optimization theory. This impracticality is due to many system complexities, such as large dimensions, nonlinearities, coupling, time delays, and physical separation of components. One possibility, as seen in Section 2.2, is to simplify the model so that the application of optimal control theory is possible. Another possibility is to develop near-optimum design techniques via perturbation of plant parameters while keeping the system model as realistic as possible. The perturbation of plant parameters leads to a decomposition in system structure which, as seen in Chapter 4, provides a number of subsystems interacting with each other in hierarchical fashion. Still another class of techniques is based on the decentralization of the information structure which leads to some interesting stabilizing control algorithms, as discussed in Chapter 5. Due to these characteristics of large-scale systems, the optimal control design of such systems is, in the most part, necessarily near-optimum in nature. The reduction of order, perturbation of parameters, decomposition of structure, omission of time delays, linearization of nonlinear terms, hierarchical interaction, and decentralization of control all would normally lead to near-optimality of system performance.

In this chapter the maximum principle of optimal control theory and optimization techniques are applied to various types of large-scale systems. Different applications of the maximum principle, such as the linear state regulator (matrix Riccati formulation) and nonlinear optimal control systems (solution of two-point boundary value problems) will be utilized.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Unconstrained optimization methods will be used to find optimal decentralized state and output feedback control laws.

The near-optimum design of large linear time-invariant systems based on aggregation, perturbation, hierarchical, and decentralized approaches is discussed in Section 6.2.

In Section 6.3 the near-optimum control of nonlinear systems is discussed. The hierarchical control through interaction prediction is first introduced for nonlinear systems.

Section 6.4 deals with the important problem of near-optimality bounds or degrees of large-scale optimal control systems. It is important for the designer to have an estimate on the loss of the system performance index by adapting a particular near-optimum technique. Near optimality due to aggregation, perturbation, hierarchical control, nonlinearities are all considered. Some CAD examples will be discussed in Section 6.5.

6.2 Near-Optimum Control of Linear Time-Invariant Systems

In this section a number of near-optimum control techniques based on model reduction methods (Chapter 2), hierarchical control (Chapter 4), and decentralized control (Chapter 5) for linear time-invariant systems are discussed, and numerous examples are given.

6.2.1 Aggregation Methods

A number of aggregated methods were discussed in Section 2.2. Here the aggregation procedure is addressed from an optimal control point of view. Consider a large-scale linear TIV system

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(0) = x_0 \tag{6.2.1}$$

with a quadratic cost functional

$$J = \frac{1}{2}\int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt \tag{6.2.2}$$

where A, B, x, and u are $(n \times n)$-, $(n \times m)$-, n-, and m-dimensional system matrix, control matrix, state vector, and control vector, respectively, and Q and R are $n \times n$ nonnegative and $m \times m$ positive-definite matrices. The optimal control problem, treated extensively in literature, is to find a control vector $u^*(t)$ such that (6.2.1) is satisfied while the cost functional (6.2.2) is minimized. The solution (Kalman, 1960) to this so-called "state regula-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

tor" is well known:

$$u^*(t) = -R^{-1}B^T Kx(t) \tag{6.2.3}$$

where K is an $n \times n$ symmetric positive-definite matrix solution of the following algebraic matrix Riccati equation (AMRE):

$$KA + A^T K - KSK + Q = 0 \tag{6.2.4}$$

where $S = BR^{-1}B^T$. The solution of AMRE (6.2.4) is well documented, and a survey by Jamshidi (1980) reports at least eight different classes of solution techniques for it. In particular, some of the best methods are the Newton-based iterative scheme initialized by a parameter-imbedding method used to solve a 90th-order river pollution system by Jamshidi (1980) and the Schur's vector transformation of the Hamiltonian matrix

$$H = \begin{bmatrix} A & -S \\ Q & -A^T \end{bmatrix} \tag{6.2.5}$$

which has been successfully used for systems of order 64 to 104 (Laub, 1979). The latter method is constrained by computer memory and is most effective for dense A matrices. It is not our objective to give a detailed treatment of various solution methods of AMRE (6.2.4); the reader can consult the overview (Jamshidi, 1980), which treats not only AMRE but also provides an exhaustive literature survey on differential, discrete-time, stochastic, and singularly perturbed matrix Riccati as well as the Lyapunov-type equations.

For any value of n, the optimal control problem (6.2.1)–(6.2.2) and its solution (6.2.3)–(6.2.4) requires a set of at least $n(n + 1)/2$ elemental values of the $n \times n$ symmetric Riccati matrix K. Clearly for a large n, the task of solving the AMRE (6.2.4) calls for considerable computational effort. Although there are some approximate solutions of the AMRE (6.2.4) through regular (Kokotović *et al.*, 1969) or singular perturbations (Yackel and Kokotović 1973), here the approximation is assumed through the representation of the state model (6.2.1) by a "coarser" set of states called "aggregated" system,

$$\dot{z}(t) = Fx(t) + Gu(t), \quad z(0) = z_0 \tag{6.2.6}$$

where $z = Cx$ is the l-dimensional ($l < n$) aggregated state, C is the $l \times n$ aggregation matrix, and F and G are $l \times l$ and $l \times m$ aggregated state and

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

control matrices (see Section 2.2) obtained from

$$F = CAC^T\left(CC^T\right)^{-1}, \qquad G = CB \tag{6.2.7}$$

One of the more popular aggregation procedures discussed in detail in Section 2.2.2 is the modal method originated by Davison (1966), which was considered as a special case of the aggregation technique due to Aoki (1968, 1978), which, in turn, is generalized by a chained aggregation scheme. Next, the aggregation models described by (2.2.22)–(2.2.26) and (2.2.39)–(2.2.47) due to Davison (1966) and Chidambara (1969), respectively, are used within the context of near-optimum design of the original system.

For the full model, the control is optimal and is given by

$$u^*(t) = -R^{-1}B^TK_fx(t) = -F^*x(t) \tag{6.2.8}$$

where K_f is the solution of the full model's AMRE

$$A^TK_f + K_fA - K_fSK_f + Q = 0 \tag{6.2.9}$$

and $S = BR^{-1}B^T$. The control for the aggregated model is

$$u^a(t) = -R^{-1}G^TK_az(t) \tag{6.2.10}$$

where K_a is the solution of the aggregated model's AMRE

$$F^TK_a + K_aF - K_aGR^{-1}G^TK_a + Q_a = 0 \tag{6.2.11}$$

Using the aggregation condition (2.2.5)–(2.2.6a), i.e., $CA = FC$, $G = CB$, pre- and postmultiplying (6.2.11) by C^T and C, respectively, the aggregated model's AMRE becomes

$$A^T(C^TK_aC) + (C^TK_aC)A - (C^TK_aC)S(C^TK_aC) + C^TQ_aC = 0 \tag{6.2.12}$$

which is identical to the full model's AMRE (6.2.9) if the following relations hold:

$$K_f = C^TK_aC, \quad Q = C^TQ_aC \tag{6.2.13}$$

Using the pseudo-inverse of C, Q_a can be written as

$$Q_a = (CC^T)^{-1}\,CQC^T(CC^T)^{-1} \tag{6.2.14}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Thus, using (6.2.10), the aggregated model's near-optimum control is given by

$$u^a(t) = -R^{-1}G^T K_a C x(t) = -F_a x(t) \tag{6.2.15}$$

The following example illustrates this near-optimum control based on aggregation.

Example 6.2.1. Consider a fifth-order system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} -0.2 & 0.5 & 0 & 0 & 0 \\ 0 & -0.5 & 1.6 & 0 & 0 \\ 0 & 0 & -14.28 & 85.71 & 0 \\ 0 & 0 & 0 & -25 & 75 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 30 \end{bmatrix} u \tag{6.2.16}$$

which represents a voltage regulator system (Jamshidi, 1983). It is desired to find an aggregated model for (6.2.16) and a near-optimum control with a quadratic cost

$$J = \frac{1}{2}\int_0^\infty \left(0.1x_1^2 + 0.01x_3^2 + 0.01x_5^2 + u^2\right) dt \tag{6.2.17}$$

SOLUTION: A careful look at (6.2.16) indicates that since the system matrix is upper triangular, the eigenvalues are −0.2, −0.5, −14.28, −25, and −10. Thus a two-dimensional reduced-order model may be sought. For a two-dimensional reduced model, the aggregation and aggregated matrices for (6.2.16) turn out to be

$$C = \begin{bmatrix} 1 & 0 & -0.004 & -0.0224 & -0.336 \\ 0 & 1 & 0.115 & 0.40400 & 3.22 \end{bmatrix}$$

$$F = \begin{bmatrix} -0.2000 & 0.50 \\ 0.0087 & -0.58 \end{bmatrix}, \quad G = \begin{bmatrix} -10.08 \\ 96.60 \end{bmatrix} \tag{6.2.18}$$

Using the full model's $\{A, B, Q, R\}$ matrices defined in (6.2.16)–(6.2.17), a fifth-order AMRE (6.2.9) is solved and the optimal feedback

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

law is given by

$$u*(t) = -R^{-1}B^T K_f x(t) = -0.26x_1 - 0.11x_2 - 0.04x_3 - 0.15x_4 - 0.59x_5 \tag{6.2.19}$$

where K_f corresponds to the full model's Riccati matrix. Next, by virtue of matrices $\{F, G, Q_a, R\}$ defined in (6.2.18), (6.2.17), and using (6.2.14) to find $Q_a = \text{diag}\{0.1, 0\}$, a second-order AMRE is solved. A feedback law (6.2.10) is obtained for the aggregated system which results in an approximate feedback law

$$u^a(t) = -R^{-1}G^T K_a Cx = -0.60x_1 - 0.17x_2 - 0.0288x_3 - 0.096x_4 - 0.67x_5 \tag{6.2.20}$$

The two control laws (6.2.19) and (6.2.20) provide the optimum and near-optimum (aggregated) control whose control and output responses are shown in Figures 6.1 and 6.2, respectively. The initial state was chosen to be $x(0) = (\,0.5\ \ 0\ \ 0\ \ 0\ \ 0\,)^T$, and the output was $y = x_1$ in both cases.

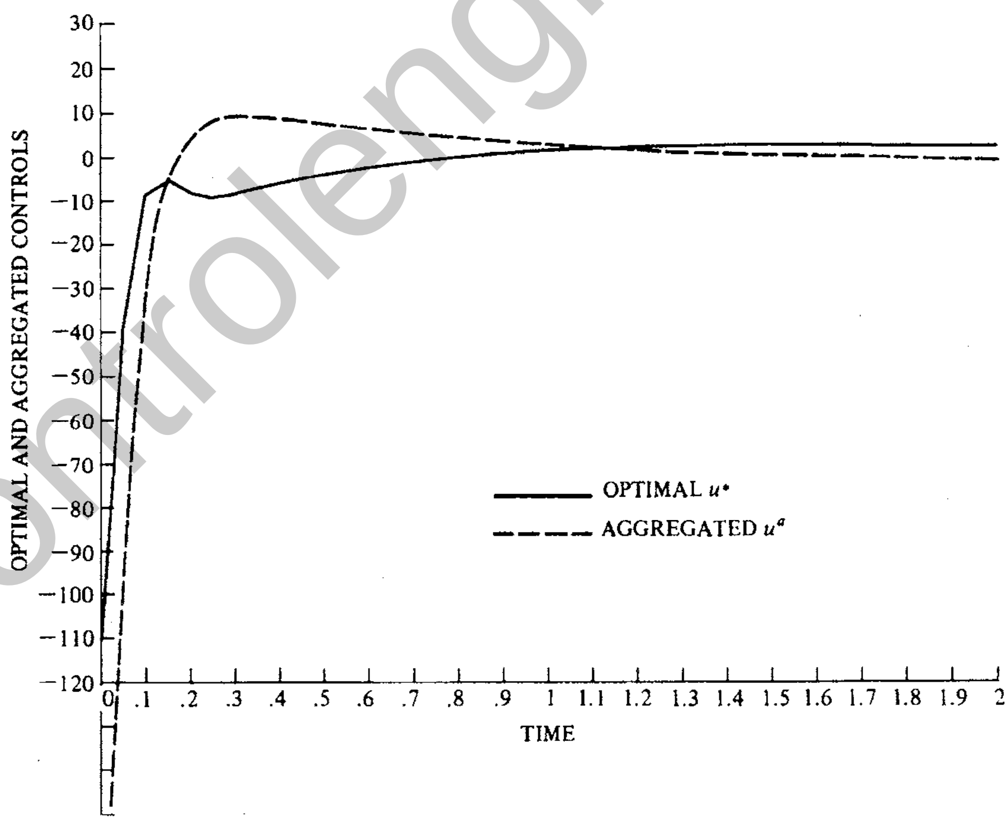


Figure 6.1. Optimal and aggregated control responses for Example 6.2.1.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

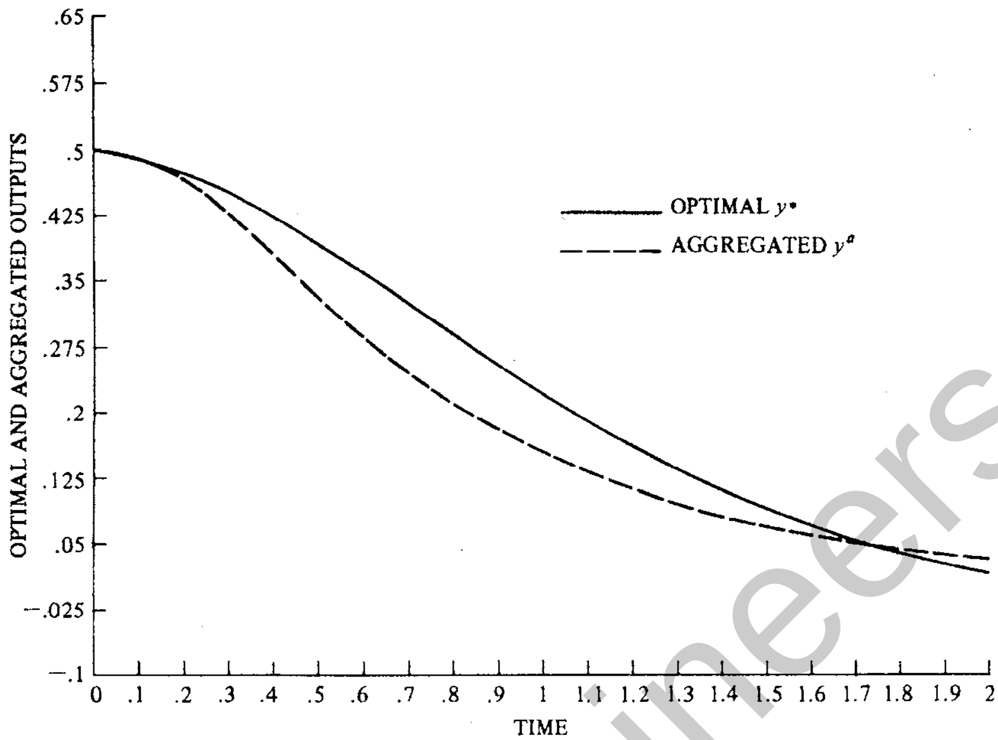


Figure 6.2. Optimal and aggregated ouput responses for Example 6.2.1.

In a similar fashion, a near-optimum control can be developed for the second aggregated model (2.2.39)–(2.2.47). Let us rewrite the quadratic cost (6.2.2) in the following form:

$$J = \frac{1}{2}\int_0^\infty \left\{\left[z^T x_2^T\right]\begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}\begin{bmatrix} z \\ x_2 \end{bmatrix} + u^T R u\right\} dt$$
$$= \frac{1}{2}\int_0^\infty \left(z^T Q_1 z + 2z^T Q_{12} x_2 + x_2^T Q_2 x_2 + u^T R u\right) dt \tag{6.2.21}$$

which can be rewritten in terms of z and u by eliminating x_2 from it using (2.2.46):

$$J_l = \frac{1}{2}\int_0^\infty \left(z^T Q_l z + 2z^T S u + u^T R_l u\right) dt \tag{6.2.22}$$

where J_1 represents the equivalent cost function and

$$Q_l = Q_1 + 2Q_{12}N + N^T Q_2 N, \;\; R_l = R + E^T Q_2 E, \;\; S = Q_{12}E + N^T Q_2 E \tag{6.2.23}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

In order to formulate a linear, quadratic problem from (2.2.47) and (6.2.22), let

$$u_l \triangleq R_l^{-1} S^T z + u \tag{6.2.24}$$

in (2.2.47),

$$\dot{z} = \left(F - GR_l^{-1}S^T\right)z + Gu_l = F_l z + Gu_l \tag{6.2.25}$$

and (6.2.22) becomes

$$J_l = \frac{1}{2}\int_0^\infty \left(z^T Q_l z + u_l^T R_l u_l\right) dt \tag{6.2.26}$$

The solution of the linear regulator problem for the aggregated system (6.2.25)–(6.2.26) for a $Q_l \geq 0$ and $R_l > 0$ is

$$u_l^* = -R_l^{-1} G^T K_l z \tag{6.2.27}$$

where K_l is the positive-definite solution of the AMRE,

$$F_l^T K_l + K_l F_l - K_l G R_l^{-1} G^T K_l + Q_l = 0 \tag{6.2.28}$$

The near-optimum control can thus be obtained as

$$u^a = u_l^* - R_l^{-1} S^T z = -R_l^{-1}\left(G^T K_l + S^T\right)z \tag{6.2.29}$$

The following example illustrates this near-optimum control law.

Example 6.2.2. Consider the fourth-order system of Example 2.2.4:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.6 & -9.22 & -33.32 & -11.3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} u \tag{6.2.30}$$

with $x(0) = (\,1 \;\; 1 \;\; 1 \;\; 1\,)^T$, cost function

$$J = \frac{1}{2}\int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{6.2.31}$$

$Q = \text{diag}(5 \;\; 4 \;\; 2 \;\; 1)$, and $R = 1$. It is desired to find a reduced-order model and a near-optimum control for the system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

SOLUTION: A second-order reduced model (F, G) was obtained in Example 2.2.4 as given by (2.2.50). The matrices F_l, R_l, and Q_l are given by

$$F_l = \begin{bmatrix} 0 & 1 \\ -0.015 & -0.234 \end{bmatrix}, \quad R_l = 1.14, \quad Q_l = \begin{bmatrix} 5.0 & 0.006 \\ 0.006 & 4.10 \end{bmatrix} \tag{6.2.32}$$

The near-optimum control law (6.2.29) resulting from the solution of a second-order AMRE becomes

$$u^a = -R_l^{-1}\left(G^T K_l + S^T\right)z = -2.087x_1 - 2.553x_2 \tag{6.2.33}$$

The optimal control was obtained by solving a fourth-order AMRE:

$$u^* = -2.233x_1 - 2.853x_2 - 0.814x_3 - 0.063x_4 \tag{6.2.34}$$

The performance indices for the particular initial condition turn out to be $J^* = 9.0062$ and $J^a = 9.1200$.

The results of this example motivate a number of comments. The reduced model has remained stable, and the corresponding near-optimum (aggregated) control law requires only two of the four states. This would mean that if the number of inaccessible states does not exceed the number of nondominant (fast) modes, this method would be appropriate. This is a contrasting point to the near-optimum law (6.2.15) for a fully modal aggregation. The degradation of the performance index is, of course, expected and its detail discussion is given in Section 6.5.

In a similar fashion a near-optimum control can be found for other aggregation methods considered in Section 2.2. Some of these applications are considered in the problem section of this chapter as well as in Section 6.5.

6.2.2 Perturbation Methods

The second principal class of model reduction methods is perturbation, which was discussed in Section 2.3. Here the concepts of regularly perturbed and singularly perturbed systems are used to find near-optimum control of a large-scale system. In this section a near-optimum control for a weakly coupled (regularly perturbed) large-scale system is developed, and a 17th-order example is used to illustrate the method. Consider the two subsystem "ε-coupled" system (2.3.2) with the (A, B) pair

$$A(\varepsilon) = \left[\begin{array}{c|c} A_1 & \varepsilon A_{12} \\ \hline \varepsilon A_{21} & A_2 \end{array}\right], \quad B(\varepsilon) = \left[\begin{array}{c|c} B_1 & 0 \\ \hline 0 & B_2 \end{array}\right] \tag{6.2.35}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where ε is a small positive parameter, A_1, A_{12}, A_{21}, A_2, B_1, and B_2 are $n_1 \times n_1$, $n_1 \times n_2$, $n_2 \times n_1$, $n_2 \times n_2$, $n_1 \times m_1$, and $n_2 \times m_2$ matrices, respectively, with $n = n_1 + n_2$ and $m = m_1 + m_2$. The corresponding AMRE (6.2.4) must be solved using (6.2.35) and matrices Q and R in (6.2.2). Here an additional approximation is obtained with regard to the accompanying Riccati matrix. An approximate solution to the AMRE (6.2.4) has been proposed by Kokotović *et al.* (1969b), Kokotović (1972), and Jamshidi (1983), which is based on a truncated MacLaurin series expansion in ε,

$$P \cong K(0) + \varepsilon K^1(0) + \ldots + \frac{\varepsilon^m}{m!} K^m(0) \tag{6.2.36}$$

where $K^i\ (0) \triangleq \partial^i K(\varepsilon) / \partial \varepsilon^i \big|_{\varepsilon=0}$.

If $K(\varepsilon)$ is partitioned,

$$K = \left[\begin{array}{c|c} K_1 & \varepsilon K_{12} \\ \hline \varepsilon K_{12}^T & K_2 \end{array}\right] \tag{6.2.37}$$

and substituted into (6.2.36), three truncated series in ε are obtained for K_1, K_{12}, K_2. It has been shown that the even-order partials of K are diagonal while the odd-order partials are antidiagonal, i.e.,

$$K^{2j} = \left[\begin{array}{c|c} K_1^{2j} & 0 \\ \hline 0 & K_2^{2j} \end{array}\right], \quad K^{2j+1} = \left[\begin{array}{c|c} 0 & K_{12}^{2j+1} \\ \hline \left(K_{12}^{2j+1}\right)^T & 0 \end{array}\right], \quad j = 0,1,2\ldots \tag{6.2.38}$$

Now if P in (6.2.36) is substituted in AMRE (6.2.4) for K, noting (6.2.37)–(6.2.38) and equating the coefficients of ε^m, $m = 0, 1, 2, \ldots$, it is deduced that K_i^0, $i = 1, 2$, are solutions of

$$K_i^0 A_i + A_i^T K_i^0 - K_i^0 S_i K_i^0 + Q_i = 0 \tag{6.2.39}$$

which are diagonal and correspond to $m = 0$. For $m = 1$, K_{12}^1 is the solution of

$$K_{12}^1 G_2 + G_1^T K_{12}^1 + F_{12}^0 = 0 \tag{6.2.40}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where

$$G_i = A_i - S_i K_i^0, \quad i = 1, 2$$
$$F_{12}^0 = A_{21}^T K_2^0 + K_1^0 A_{12} \tag{6.2.41}$$

For $m = 2$, the block-diagonal matrices K_i^2 are the solutions of

$$K_i^2 G_i + G_i^T K_i^2 + F_i^1 = 0, \quad i = 1, 2 \tag{6.2.42}$$

where

$$F_1^1 = 2\left(A_{21} - S_2 K_{21}^1\right)^T K_{21}^1 + 2K_{12}^1 A_{21}$$
$$F_2^1 = 2\left(A_{12} - S_1 K_{12}^1\right)^T K_{12}^1 + 2K_{21}^1 A_{12} \tag{6.2.43}$$

and $K_{21}^1 = \left(K_{12}^1\right)^T$. For $m = 3$ and 4, two sets of matrix equations similar to (6.2.40) and (6.2.42) are obtained for K_{12}^3 and K_i^4 except for the forcing terms, which are now F_{12}^1 and F_i^2, $i = 1, 2$, respectively. The following algorithm summarizes the near-optimum control of an "ε-coupled" system.

Algorithm 6.1. Near-Optimum Control of "ε-Coupled" Systems

Step 0: Solve two low-order AMREs (6.2.39) for K_i^0, $i = 1, 2$, and store.

Step 1: Solve the Lyapunov-type* equations (6.2.40) for K_{12}^1 using K_i^0 and store.

Step 2: Solve the Lyapunov equation (6.2.42) for K_i^2, $i = 1, 2$, and store.

⋮

Step m − 1: Solve for antidiagonal block matrix K_{12}^{m-1} and store.

Step m: Solve for diagonal block matrices K_i^m, $i = 1, 2$.

Step m + 1: The mth-order truncated Riccati matrix is obtained from (6.2.36) noting the diagonal properties (6.2.38).

Step m + 2: The near-optimum control is given by

$$u^a(t) = -R^{-1} B^T P x(t) \tag{6.2.44}$$

where $R = \text{Block-diag}\{R_1, R_2\}$, $B = \text{Block-diag}\{B_1, B_2\}$.

---

* For the numerical solution of a Lyapunov-type equation, see Algorithm 6.2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The following example illustrates this technique.

Example 6.2.3. Consider the 17th-order system of Example 2.3.1, which represents a simplified model of a three-stand cold rolling mill (Figure 2.12) with the following partitions: $n_1 = 8$, $n_2 = 9$, $m_1 = 5$, and $m_2 = 7$. For computational purposes, let the payoff or winding reel radius have a value $r = 0.84$ ft., as defined in (2.3.5). Find a near-optimum feedback law for this system. The other matrices are $Q_1 = \text{diag}\{0.5, 0.5, 0.5, 1, 1, 5, 1, 5\}$, $Q_2 = \text{diag}\{1, 5, 1, 0.5, 0.5, 0.5, 1, 1, 5\}$, $R_1 = I_5$, and $R_2 = I_7$ (Jamshidi, 1980).

SOLUTION: The results of Step 0 summarize into

$$P = \left[\begin{array}{c:c} P_1 & P_{12} \\ \hdashline P_{12}^T & P \end{array}\right] \cong \left[\begin{array}{c:c} K_1^0 + \varepsilon^2/2!K_1^2 & \varepsilon K_{12}^1 \\ \hdashline \varepsilon\left(K_{12}^1\right)^T & K_2^0 + \varepsilon^2/2!K_2^2 \end{array}\right] \tag{6.2.45}$$

where symmetric matrices P_i, K_i^0, $i = 1, 2$, and matrix P_{12} are given by (6.2.46).

$$P_1 = \begin{bmatrix} -0.08 & -1.4 & -0.08 & -1.31 & -0.17 & -0.43 & -0.08 & -13.7 \\ & 16.95 & -0.15 & 9.07 & -0.33 & -1.98 & -0.44 & -19.0 \\ & & 0.7 & -0.14 & -0.02 & -0.04 & -0.08 & -1.46 \\ & & & 26.0 & 3.52 & 1.44 & 3.3 & -21.0 \\ & & & & 0.94 & 0.45 & 0.83 & -43.8 \\ & & & & & 0.94 & -2.76 & -0.95 \\ & & & & & & -7.8 & 0.03 \\ & & & & & & & -233.0 \end{bmatrix}$$

$$K_1^0 = \begin{bmatrix} 1.11 & -0.03 & 0.05 & 0.03 & 0.12 & -0.376 & -0.09 & 0.526 \\ & 18.87 & -0.003 & 11.04 & 0.07 & -1.91 & -0.51 & 1.066 \\ & & 0.688 & 16.78 & 0.01 & -0.04 & -0.01 & 0.054 \\ & & & 28.01 & 3.92 & 1.66 & 3.63 & 1.21 \\ & & & & 10.22 & 0.50 & 0.93 & 0.24 \\ & & & & & 0.85 & -3.06 & 0.07 \\ & & & & & & -8.63 & -0.02 \\ & & & & & & & 12.71 \end{bmatrix} \tag{6.2.46}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$K_2^0 = \begin{bmatrix} -8.78 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 23.32 & -2.39 & -0.16 & -0.01 & -0.02 & 0.90 & -0.35 & 0.75 \\ & & -10.2 & 0.14 & -0.50 & 0.01 & 3.48 & -1.31 & 3.3 \\ & & & 1.14 & -1.11 & 0.05 & -0.72 & 0.15 & -0.32 \\ & & & & 20.70 & -0.11 & 12.9 & -1.0 & 0.92 \\ & & & & & 0.73 & -0.07 & 0.02 & -0.032 \\ & & & & & & 20.01 & -0.347 & -1.64 \\ & & & & & & & 1.05 & 0.58 \\ & & & & & & & & 0.12 \end{bmatrix}$$

The matrices in (6.2.45) are used at Step 1 to find K_{12}^1, which in turn, is used to find K_i^2, $i = 1, 2$.

$$P_{12} = \begin{bmatrix} 0.02 & 0.57 & 0.0 & -0.25 & -0.03 & -0.03 & -0.05 & -0.07 & 0.0 \\ 0.15 & 0.58 & -0.01 & -0.2 & -0.02 & -0.02 & -0.09 & -0.06 & 0.02 \\ 0.0 & 0.75 & 0.0 & -0.02 & 0.0 & 0.0 & 0.0 & -0.01 & 0.0 \\ -0.36 & 0.76 & -0.01 & -0.19 & 0.05 & -0.02 & 0.02 & -0.07 & 0.0 \\ -0.1 & 0.16 & 0.0 & -0.04 & 0.02 & 0.0 & 0.03 & -0.01 & 0.0 \\ 0.13 & 0.04 & 0.0 & -0.01 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -10.44 & 8.4 & -0.03 & -2.1 & 1.05 & -0.25 & 1.32 & -0.73 & -0.08 \end{bmatrix} \tag{6.2.47}$$

$$P_2 = \begin{bmatrix} -19.4 & 0.03 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ & 32.93 & -11.5 & -0.2 & -0.25 & -0.02 & 4.12 & -1.55 & 3.74 \\ & & -50.9 & 0.65 & -2.33 & 0.06 & 17.0 & -6.44 & 16.6 \\ & & & 1.13 & -1.1 & 0.04 & -0.9 & 0.22 & -0.5 \\ & & & & 20.6 & -0.11 & 13.5 & -1.23 & 1.52 \\ & & & & & 0.73 & -0.09 & 0.02 & -0.05 \\ & & & & & & 15.5 & -1.76 & -6.0 \\ & & & & & & & 0.4 & 2.25 \\ & & & & & & & & -4.22 \end{bmatrix}$$

Equation (6.2.47) is then used to obtain the near-optimum feedback control (6.2.48).

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$
\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -\left[\begin{array}{c|c} R_1^{-1}B_1^T P_1 & R_1^{-1}B_1^T P_{12} \\ \hline R_2^{-1}B_2^T P_{21} & R_2^{-1}B_2^T P_2 \end{array}\right]\left[\begin{array}{c} x_1 \\ \hline x_2 \end{array}\right]
= -\left[\begin{array}{c|c} F_1 & F_{12} \\ \hline F_{21} & F_2 \end{array}\right]\left[\begin{array}{c} x_1 \\ \hline x_2 \end{array}\right] \tag{6.2.48}
$$

The resulting (12 × 17)-dimensional gain matrix is given by

$$
F = \left[\begin{array}{cccccccc|ccccccccc}
-0.02 & -0.45 & & -0.42 & & -0.14 & & -4.38 & 0.0 & 0.22 & & -0.08 & & & & & \\
-0.45 & 5.4 & & 2.9 & -0.1 & -0.63 & -0.14 & -6.05 & 0.0 & 0.25 & & -0.06 & & & & & \\
 & & 0.14 & & & & & -0.29 & 0.0 & 0.0 & & & 0 & & & & \\
0.09 & 0.54 & & -4.0 & -1.01 & 3.38 & 9.48 & -0.04 & 0.0 & 0.0 & & & & & & & \\
0.09 & 0.54 & & -4.0 & -1.01 & 3.38 & 9.48 & -0.04 & 0.0 & 0.0 & & & & & & & \\
\hline
 & -0.12 & 0.29 & 0.1 & -0.1 & & & 8.62 & 16.0 & & & & & & & & \\
 & -0.12 & 0.29 & 0.1 & -0.1 & & & 8.62 & 16.0 & & & & & & & & \\
 & & & & & & & 0.0 & & 4.9 & 21.6 & -0.27 & 0.99 & 0.0 & -7.22 & 2.73 & -7.03 \\
 & & & 0 & & & & 0.0 & & 4.9 & 21.6 & -0.27 & 0.99 & 0.0 & -7.22 & 2.73 & -7.03 \\
-0.08 & -0.06 & & & & & & -0.66 & & & 0.2 & 0.36 & -0.35 & 0.0 & -0.29 & 0.07 & -0.15 \\
0.0 & & & & & & & 0.33 & & -0.08 & -0.74 & -0.35 & 6.55 & 0.0 & 4.28 & -0.39 & 0.48 \\
0.0 & & & & & & & 0.0 & & 0.0 & & & & & & & 0.0
\end{array}\right] \tag{6.2.49}
$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

6.2.3 Decentralized Control via Unconstrained Minimization

In Chapter 5, the decentralized control was presented; several methods were discussed and supported by numerical examples. In this section, one such method for application to large-scale systems is presented. The method is a modification of the algorithm due to Geromel and Bernussou (1979).

Consider a large-scale system described by a set of N subsystems,

$$\dot{x}_i = A_{ii}x_i + B_iu_i + \sum_{j=1}^{N} A_{ij}x_j, \quad x_i(0) = x_{io}, \quad i = 1,\ldots,N \tag{6.2.50}$$

where x_i and u_i are n_i- and m_i-dimensional local state and control vectors, respectively. Geromel and Bernussou (1979) have proposed a set of N totally decentralized control laws while minimizing a quadratic cost

$$u_i = -k_ix_i, \quad i = 1, 2, \ldots, N \tag{6.2.51}$$

while minimizing a quadratic cost

$$J = \frac{1}{2}\sum_{i=1}^{N}\int_0^\infty \left(x_i^T Q_i x_i + u_i^T R_i u_i\right) dt \tag{6.2.52}$$

where each Q_i and R_i matrix pairs are positive-semidefinite and positive-definite, respectively. Defining the block and block-diagonal matrices

$$\begin{aligned} &A = \{A_{ij}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, N\}, B = \text{Block-diag}(B_1, B_2, \ldots, B_N)\\ &K = \text{Block-diag}(K_1, K_2, \ldots, K_N), Q = \text{Block-diag}(Q_1, Q_2, \ldots, Q_N)\\ &R = \text{Block-diag}(R_1, R_2, \ldots, R_N) \end{aligned} \tag{6.2.53}$$

the above problem is reformulated as

$$\min_u: J = \frac{1}{2}\int_0^\infty \left(x^TQx + u^TRu\right) dt \tag{6.2.54}$$

$$\dot{x} = Ax + Bu, \quad x(0) = x_0 \tag{6.2.55}$$

$$u = -Kx \tag{6.2.56}$$

Unlike the Hamiltonian-based methods of optimizing with respect to the control function, the problem (6.2.54)–(6.2.56) can be solved through pa-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

rameter optimization once the cost (6.2.54) as a function of K, i.e., $J(K)$, is obtained. This function can be obtained by

$$J(K) = \frac{1}{2} x_0^T \left(\int_0^\infty \Phi^T(t)\left(Q + K^T RK\right)\Phi(t)dt \right) x_0$$
$$= tr\left\{ \frac{1}{2} \int_0^\infty \Phi^T(t)\left(Q + K^T RK\right)\Phi(t)dt\left(x_0 x_0^T\right) \right\} \tag{6.2.57}$$

where $\Phi(t)$ is the transition matrix of closed-loop system, $\dot{x} = (A - BK)x$ and tr{ · } represents trace of { · }. With this expression for $J(K)$, the optimal decentralized control problem (6.2.54)–(6.2.56) is reduced to

$$\min_K J(K) = tr\{F(K)X_0\} \tag{6.2.58}$$

where $F(K)$ is the solution of the following matrix Lyapunov equation:

$$(A - BK)^T F + F(A - BK) + Q + K^T RK = 0 \tag{6.2.59}$$

and $X_0 = x_o x_0^T$. Thus, the functional optimization problem with respect to $u(t)$ is reduced to a parametric one. This problem, however, depends on the initial state x_0 as formulated. In order to avoid this dependency, one can assume that x_0 is a random variable, and then an average value of $J(K)$ is chosen as the optimum cost. One possible way is to assume that x_0 is a uniformly distributed random variable, i.e., $X_0 = E\{x_0 x_0^T\} = (I / n)$, where E{ · } is the expected value and I is the identity matrix. Geromel and Bernussou (1979) have used the standard gradient method and update K by the so-called feasible direction matrix, which depends on the gradient of $J(K)$ with respect to K:

$$\nabla_K J(K) = G(K) = (RK - B^T F)P \tag{6.2.60}$$

$$(A - BK)P + P(A - BK)^T + X_0 = 0 \tag{6.2.61}$$

and F is the solution of (6.2.59). For a positive-definite solution of (6.2.82) and (6.2.61), the necessary and sufficient condition is that (A–BK) be asymptotically stable (Barnett and Storey, 1970). Instead of using the feasible direction method, an algorithm is formed by using the well-known Davidon-Fletcher-Powell variable metric method (Fletcher and Powell, 1963). However, before the decentralized control algorithm is formally

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

given, another algorithm for the iterative solution of the Lyapunov equation due to Davison and Man (1968) is considered. The following algorithm is presented without any further discussion.

Algorithm 6.2. Solution of the Lyapunov Equation

Step 1: For the solution of a Lyapunov equation

$$A^T L + LA + S = 0 \tag{6.2.62}$$

where A is a stable matrix, choose a step size $h = 10^{-4}/(2\|A\|)$ and set $L_0 = hS$.

Step 2: Calculate the matrix

$$E = \left(I - hA/2 + h^2A^2/12\right)^{-1}\left(I + hA/2 + h^2A^2/12\right) \tag{6.2.63}$$

where I is an identity matrix.

Step 3: Find the next value of L, i.e.,

$$L_{i+1} = \left(E^T\right)^{2^i} L_i E^{2^i} + L_i \tag{6.2.64}$$

Step 4: Check if $\|\Delta L\| = \left\|L_{i+1} - L_i\right\| < \varepsilon$, a prespecified tolerance. If not, set $i = i + 1$ and go to Step 3.

Step 5: Stop.

It is emphasized that the above algorithm converges only if all the eigenvalues of A have negative real parts. This algorithm was coded in BASIC, and it was found that in only ten iterations the algorithm converged to within six digits of accuracy. The following algorithm provides the solution for the desired optimal decentralized control problem.

Algorithm 6.3. Optimal Decentralized Control via Functional Minimization

Step 1: Select initial values $K = K_0$ and set $k = 0$.

Step 2: Solve Lyapunov equations (6.2.59) and (6.2.61) using A, B, Q, R, X_0, K and Algorithm 6.2 for P and F. Evaluate $J(K)$ and $G(K)$ using (6.2.58) and (6.2.60), respectively.

Step 3: Use the Fletcher-Powell method to update gain K^k

$$K^{k-\text{new}} = K^{k-\text{old}} - \varepsilon D^k \tag{6.2.65}$$

where D^k is the search direction during the kth iteration.

Step 4: Check whether convergence is achieved, e.g., $\left\|G\left(K_r^k\right)\right\| \le \delta$, a

controlengineers.ir
@controlengineers

prespecified tolerance value. If convergence is reached, stop; otherwise go to Step 2.

The above algorithm was also coded in LSSPAK and was used for optimal decentralized control system design. The following example illustrates this algorithm.

Example 6.2.4. Consider a fourth-order system

$$\dot{x} = \begin{bmatrix} 0 & 0.2 & 0.25 & 1 \\ -1 & -2 & 1 & 0 \\ -1 & 0.1 & 0.85 & 1 \\ 0.25 & -0.5 & 0 & -0.25 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0.1 \\ 0 & 1 \end{bmatrix} u$$

with a cost functional,

$$J = \frac{1}{2}\sum_{i=1}^{2} \int_0^\infty \left(x_i^T Q_i x_i + u_i^T R_i u_i\right) dt$$

where $Q_i = 2I_{ni}$, $R_i = 2I_{mi}$ with $n_1 = n_2 = 2$ and $m_1 = m_2 = 1$. Find two decentralized gains $u_i = -K_i x_i$ such that J is minimized.

SOLUTION: The two decentralized controllers are $u_1 = -K_1 x_1$ and $u_2 = -K_2 x_2$ with the following compact form

$$K = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ 0 & 0 & k_{21} & k_{22} \end{bmatrix}$$

Thus, there are four unknown gain parameters to be found. For a value of $X_0 = I_4$ and two initial values of K_0,

$$K_0 = \begin{bmatrix} 0.1 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 \end{bmatrix}, \quad K_0 = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

Algorithm 6.3 converged in seven and 13 iterations, respectively. The re-

controlengineers.ir
@controlengineers

sulting averaged optimum gain K^* and the gradient matrix $\nabla_K J(K^*)= \partial J(K^*)/\partial K$ were found to be

$$K^* = \begin{bmatrix} 0.494 & 0.168 & 0 & 0 \\ 0 & 0 & 0.35 & 0.58 \end{bmatrix},$$

$$\nabla_K J(K^*) = \begin{bmatrix} -2.5\times10^{-3} & 1.4\times10^{-3} & 0 & 0 \\ 0 & 0 & 0.017 & 9.4\times10^{-3} \end{bmatrix}$$

which gives $J^* = 1.0985$. The optimum decentralized control laws are then,

$$u_1^* = -0.494x_1 - 0.168x_2$$
$$u_2^* = -0.35x_3 - 0.58x_4$$

Further discussion on this algorithm and other examples are considered in Section 6.6 and Problem 6.4.

6.3 Near-Optimum Control of Large-Scale Nonlinear Systems

Throughout our discussions in Chapters 4 through 6, where the subject of large-scale systems control has been discussed, the focus has been primarily on large-scale linear TIV systems. In this section the case of nonlinear systems is considered. A general approximate linearization scheme of optimally controlling nonlinear systems is first discussed. Then hierarchical and perturbation control schemes will be discussed.

6.3.1 Near-Optimum Control via Sensitivity Methods

Consider a nonlinear system

$$\dot{x} = g(x,u), \quad x(0) = x_0 \tag{6.3.1}$$

where x, u, and x_0 are defined as before. The problem of finding an optimum control which satisfies (6.3.1) and minimizing a cost functional

$$J = L\big(x(t_f)\big) + \int_0^{t_f} V(x, u, t)\,dt \tag{6.3.2}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

has been of interest for several years. Although the optimization of this problem by the classical methods—gradient, Newton's second variation, etc.—is possible, near-optimum control laws have received special attention (Kelly, 1964; White and Cook, 1973; Garrard *et al.*, 1967; Garrard, 1969, 1972; Garrard and Jordan, 1977; Werner and Cruz, 1968; Nishikawa *et al.*, 1971; Kokotović *et al.*, 1969a,b; Jamshidi, 1969, 1976). Some of the methods suggested in literature have been "equivalent linearization" (White and Cook, 1973) and "approximate solution to the Hamilton-Jacobi-Bellman's equation" (Garrard *et al.*, 1967; Garrard, 1969, 1972). Another approach is the so called "optimally adaptive" (Werner and Cruz, 1968), which minimizes the system cost functional regardless of plant parameters or initial condition variations. The control function is expanded in a MacLaurin's series of plant parameters and initial conditions. It has been asserted (Werner and Cruz, 1968) that for an rth-order truncation of the control function series, the optimum cost functional has been approximated up to $2r + 1$ terms. Another near-optimum control design developed by Nishikawa *et al.* (1971) takes on the general class of nonlinear systems given in Garrard, et al. (1967) while making use of a similar parameter expansion as in the optimally adaptive control of Werner and Cruz (1968). The "optimally sensitive controller" (Kokotović *et al.*, 1969a,b) is a first-order approximation of the optimally adaptive control which presents a convenient derivation of sensitivity equations from the maximum principle. Jamshidi (1969, 1983) has extended the optimally sensitive control to a wider class of nonlinear systems than those considered by Garrard *et al.* (1967) and Nishikawa *et al.* (1971) while preserving the same order of approximation of the cost functional as in the optimally adaptive control. This method is briefly discussed here and a numerical example illustrates it.

Expanding (6.3.1) about a nominal (equilibrium) point, taken to be the origin for convenience, i.e., $x = 0$, $u = 0$, such that $g(0,0) = 0$, then

$$\dot{x} = Ax + Bu + \varepsilon f(x, u) \tag{6.3.3a}$$

where

$$\begin{aligned} &A \triangleq \partial g/\partial x\big|_{x,\,u=0} \qquad B \triangleq \partial g/\partial u\big|_{x,\,u=0} \\ &f \triangleq g(x, u) - Ax - Bu \end{aligned} \tag{6.3.3b}$$

$\varepsilon \in [0,1]$ is a scalar parameter and A and B are constant matrices of order $n \times n$ and $n \times m$, respectively. It is noted that for $\varepsilon = 0$, the nonlinear system (6.3.1) becomes linear, while for $\varepsilon = 1$, linearized system (6.3.3a) corre-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

sponds to the original system. This process of introducing a parameter in the original system is an attempt to "imbed" it in a family of similar problems such that the solution to one of its member (for $\varepsilon = 0$ in this case) is easily obtained. The scheme, known as "parameter imbedding," or "continuation" (Ortega and Rheinboldt, 1970), is to begin from the solution of the easy member in the family of problems by varying ε from 0 to 1, until the solution of the desired problem is obtained. The uniqueness conditions for a solution to (6.3.3) are:

1. $x(\varepsilon)$, $u(\varepsilon)$ are continuously differentiable with respect to ε;
2. $f(\cdot)$ is continuously differentiable with respect to x and u; and
3. $\partial f/\partial\varepsilon$ is continuous in x, u, and ε.

Now let us rewrite (6.3.3a) as

$$\dot{x}(t,\varepsilon)=h(x,u,t,\varepsilon)=Ax(t,\varepsilon)+Bu(t,\varepsilon)+\varepsilon f\big(x(t,\varepsilon),u(t,\varepsilon),\varepsilon\big) \tag{6.3.4}$$

$$x(t_0,\varepsilon)=x_0 \tag{6.3.5}$$

Without loss of generality, the cost functional (6.3.2) is assumed to be quadratic:

$$J=\frac{1}{2}x^T(t_f,\varepsilon)Mx(t_f,\varepsilon)+\frac{1}{2}\int_0^{t_f}\left(x^TQx+u^TRu\right)dt \tag{6.3.6}$$

where M, Q, and R satisfy the usual linear state regulator conditions. The necessary optimality conditions based on the maximum principle are

$$\dot{x}=H_p=h=Ax+Bu+\varepsilon f(x,u,\varepsilon),\qquad x(t_0,\varepsilon)=x_0 \tag{6.3.7a}$$

$$\dot{p}=-H_x=Qx-A^Tp-\varepsilon(\partial f/\partial x)^T p,\quad p(t_f,\varepsilon)=-Mx(t_f,\varepsilon) \tag{6.3.7b}$$

$$0=H_u=-Ru+\big(B+\varepsilon(\partial f/\partial u)\big)^T p \tag{6.3.8}$$

where

$$H=-\tfrac{1}{2}\left(x^TQx+u^TRu\right)+p^T\big(Ax+Bu+\varepsilon f(x,u,\varepsilon)\big) \tag{6.3.9}$$

is the Hamiltonian function and the subscripts denote vector gradients. Note that for nonzero values of ε, (6.3.7)–(6.3.9) represent a nonlinear TPBV problem whose solution is usually difficult to obtain. In order to overcome this difficulty let us differentiate (6.3.7)–(6.3.9) with respect to

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ε and let $\varepsilon \to 0$, assuming that all conditions discussed above hold:

$$\dot{x}^1 = h_x x^1 + h_u u^1 + f,\ \ x^1(t_0) = 0 \tag{6.3.10}$$

$$\dot{p}^1 = -H_{xx}x^1 - h_x^T p^1 - H_{ux}^T u^1 - f_x^T p,\quad p^1(t_f) = 0 \tag{6.3.11}$$

$$0 = H_{ux}^T x^1 + h_u^T p^1 + H_{uu}u^1 + f_u^T p \tag{6.3.12}$$

where, $x \triangleq \lim_{\varepsilon \to 0} \partial x(t,\varepsilon)/\partial\varepsilon$ similarly for u^1 and p^1, are sometimes referred to as first-order sensitivity functions (Kokotović *et al*., 1969a). Assuming that H_{uu} is negative-definite, by eliminating u^1 in (6.3.10)–(6.3.11) by using (6.3.12), the following linear TPBV problem results:

$$\dot{x}^1 = Fx^1 + Ep^1 + \omega_0 \tag{6.3.13}$$

$$\dot{p}^1 = Gx^1 - F^T p^1 + \delta_0 \tag{6.3.14}$$

where

$$\begin{aligned} F &= h_x - h_u H_{uu}^{-1} H_{ux}^T\big|_{\varepsilon\to 0} = A \\ G &= H_{xu}^T H_{uu}^{-1} H_{ux}^T - H_{xx}\big|_{\varepsilon\to 0} = Q, \quad \omega_0 = f - h_u H_{uu}^{-1} f_u^T p\big|_{\varepsilon\to 0} \\ E &= -h_u H_{uu}^{-1} h_u^T\big|_{\varepsilon\to 0} = BR^{-1}B^T \triangleq S, \quad \delta_0 = H_{xu}^T H_{uu}^{-1} f_u^T p - f_x^T p\big|_{\varepsilon\to 0} \end{aligned} \tag{6.3.15}$$

It is well known (Kalman, 1960) that x^1 and p^1 are related by

$$p^1 = -Kx^1 + g_1 \tag{6.3.16}$$

where K is the symmetric positive-definite solution of the differential matrix Riccati equation (DMRE)

$$\dot{K} = -KA - A^T K + KSK - Q,\quad K(t_f) = M \tag{6.3.17}$$

and g_1 is the solution of an adjoint vector equation

$$\dot{g}_1 = -(A - SK)^T g_1 + K\omega_0 + \delta_0 \tag{6.3.18}$$

whose boundary condition in view of (6.3.11) and (6.3.16) is

$$g_1(t_f) = K(t_f)x^1(t_f) = Mx^1(t_f) \tag{6.3.19}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Note that since only the final conditions of K and g_1 are known, (6.3.17) and (6.3.18) must be solved backward in time.

Substituting p^1 of (6.3.16) in (6.3.13) results in

$$\dot{x}^1 = (A - SK)x^1 + Sg_1 + \omega_0 \tag{6.3.20}$$

which can be solved for x^1. Considering (6.3.12) and (6.3.16), the first-order control sensitivity u^1 can be obtained:

$$u^1 = -R^{-1}B^T Kx^1 + R^{-1}\left(B^T g_1 + f_u^T p^0\right) = -R^{-1}B^T Kx^1 - \theta_1 \tag{6.3.21}$$

where p^0 is the costate zeroth-order sensitivity or the costate vector itself which acts as the forcing function for the first-order terms evaluated along the "nominal trajectory." It is noted that (6.3.20)–(6.3.21) constitutes the first-order coefficients of the MacLaurin's series expansions of x and u in the parameter ε, i.e.,

$$x = x^0 + \varepsilon x^1 + \varepsilon^2 x^2/2! \ldots, \qquad u = u^0 + \varepsilon u^1 + \varepsilon^2 u^2/2! + \ldots \tag{6.3.22}$$

The second-order terms x^2 and u^2 can be similarly obtained by differentiating (6.3.10)–(6.3.12) with respect to ε and letting $\varepsilon \to 0$. This would require that the state, control, and costate vectors be continuously differentiable with respect to ε as many times as the designer wishes, i.e.,

$$\dot{x}^i = (A - SK)x^i + Sg_i + \omega_{i-1} \tag{6.3.23}$$

$$u^i = -R^{-1}B^T Kx^i + R^{-1}\left(B^T g_i + f_u^T p^{i-1}\right) = -R^{-1}B^T Kx^i - \theta_i \tag{6.3.24}$$

for $i = 1, 2\ldots$, where all coefficients are evaluated at $\varepsilon = 0$, i.e., along the "nominal trajectory." This is obtained by letting $\varepsilon \to 0$ in (6.3.7)–(6.3.8), which reduces the originally nonlinear TPBV problem to a linear regulator whose solution is

$$u^0 = -R^{-1}B^T Kx^0 \tag{6.3.25}$$

$$\dot{x}^0 = (A - SK)x^0, \quad x^0(t_0) = x_0 \tag{6.3.26}$$

where K is the solution of DMRE (6.3.17).

The proposed near-optimum control can now be shown to have an exact feedback and an approximate forward term. Substituting (6.3.25) and

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

(6.3.24) in (6.3.22) results in the following:

$$
\begin{aligned}
u &= -R^{-1}B^T K \sum_{i=0}^{\infty} \varepsilon^i x^i / i! + R^{-1} \sum_{i=1}^{\infty} \varepsilon^i \left(B^T g_i + f_u^T p^{i-1} \right) / i! \\
&= -R^{-1}B^T K x + \sum_{i=1}^{\infty} \varepsilon^i \theta_i / i!
\end{aligned}
\tag{6.3.27}
$$

Note that the second part of (6.3.27) was obtained in view of (6.3.22). The proposed rth-order near-optimum control can be obtained by truncating the second series in (6.3.27) after r terms, i.e.,

$$
u^r \simeq -R^{-1}B^T K x + \sum_{i=1}^{r} \varepsilon^i \theta_i / i! \tag{6.3.28}
$$

which clearly shows that only the forward term contributes to the suboptimality of the control. Substituting (6.3.28) in (6.3.4) results in

$$
\dot{x}(t, \varepsilon) = (A - SK)x(t, \varepsilon) - B \sum_{i=1}^{r} \varepsilon^i \theta_i / i! + \varepsilon f\left(x(t, \varepsilon), u(t, \varepsilon), \varepsilon \right) \tag{6.3.29}
$$

It must be emphasized that by virtue of (6.3.25)–(6.3.26), (6.3.20)–(6.3.21), and (6.3.23)–(6.3.24) the homogeneous portions of all orders of the sensitivity functions remain the same. Furthermore, the solution of the $(i-1)$th terms should be used as the forcing function for evaluating the ith term. Figure 6.3 shows the structure of the proposed control, while the following algorithm provides the computational procedure.

Algorithm 6.4. Near-Optimum Control of Nonlinear Systems

- *Step 1*: Input A, B, Q, R, ε, and r. Let $i = 1$.
- *Step 2*: Solve DMRE (6.3.17) and store.
- *Step 3*: Compute nominal state and control vectors using (6.3.25)–(6.3.26).
- *Step 4*: Evaluate $(i-1)$th forcing terms δ_{i-1} and ω_{i-1} using (6.3.15) and the like.
- *Step 5*: Compute the ith-order adjoint vector g_i and sensitivity functions x^i, p^i, and u^i using (6.3.20)–(6.3.21), (6.3.23), (6.3.24), and the like.
- *Step 6*: Evaluate ith-order state and control vectors using (6.3.22).
- *Step 7*: If $i = i + 1 \leq r$, go to Step 4.
- *Step 8*: Stop.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

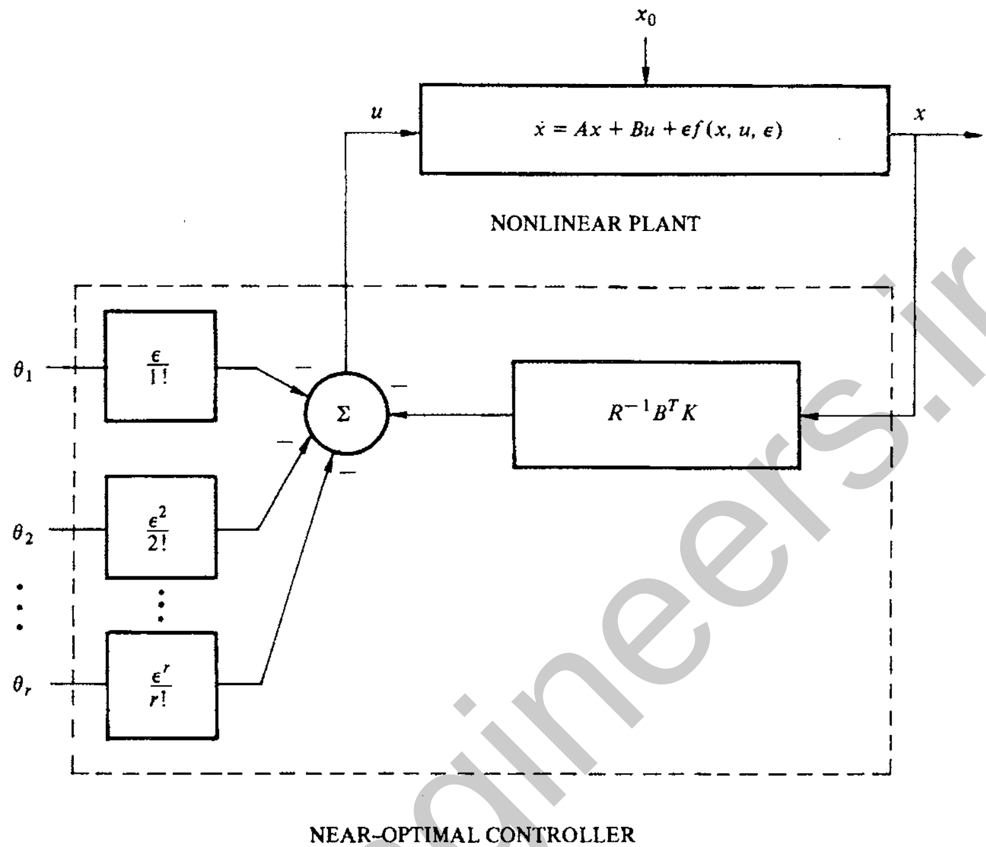


Figure 6.3. Block diagram for the rth-order near-optimum controller.

The following example illustrates Algorithm 6.5.

Example 6.3.1. Consider a synthesis problem of a "control logic for a regulation system" (Garrard *et al.*, 1967) whose torque source is a field control DC motor. The dynamic torque, field circuit equations are

$$J_L \frac{d^2(\delta\theta)}{dt^2} + B_L \frac{d(\delta\theta)}{dt} = (K_i E/R_a) i_f - (K_i K_b/R_a) \frac{d(\delta\theta)}{dt} i_f^2$$
$$L_f \frac{di_f}{dt} + R_f i_f = e_f \tag{6.3.30}$$

In (6.3.30), J_L = moment of inertia of the load, $\delta\theta$ = angular position error of the shaft, B_L= friction loss coefficient, $\delta\dot{\theta} = \delta\omega$ is the angular velocity error of the shaft, K_i= torque constant, L_f = field resistance, e_f = field

controlengineers.ir

اختصاصی مهندسی کنترل controlengineers.ir @controlengineers

voltage, E = constant armature voltage, R_a = armature resistance, and K_b = back emf constant. Armature inductance has been neglected. The nonlinear term in (6.3.30) arises from the magnetization properties of the motor's ferromagnetic material. Equation (6.3.30) can be rewritten as

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= a_{22}x_2 + a_{23}x_3 - \varepsilon x_2 x_3^2\\ \dot{x}_3 &= a_{33}x_3 + b_3 u\end{aligned} \tag{6.3.31}$$

where $x_1 = \delta\theta$, $x_2 = \delta\omega$, $x_3 = i_f$, $u = e_f$, $a_{22} = -B_L/J$, $a_{23} = K_i E / J_L R_a$, $a_{33} = -R_f / L_f$, $b_3 = 1/L_f$, and $\varepsilon = K_i K_b / J_L R_a$. Since R_a has a relatively large value, the magnitude of ε is small compared to a_{22} and a_{23}, and it is normally set equal to zero and the nonlinear term is neglected (Garrard *et al.*, 1967). The performance index is $J = \frac{1}{2}\int_0^\infty \left(x^T Qx + r_1 u^2\right) dt$. The control problem is to find a suitable near-optimum control which approximately minimizes J. For computational purposes, the same numerical values of previous work are used so that the result can be compared with those developed by Garrard *et al.* (1967) and the linearization scheme of White and Cook (1973):

$$\dot{x} = Ax + Bu + \varepsilon f(x) \tag{6.3.32}$$

$$A = \begin{bmatrix} 0 & 1 & 0\\ 0 & -0.2 & 2\\ 0 & 0 & 5\end{bmatrix},\ B = \begin{bmatrix}0\\0\\1\end{bmatrix},\ f = \begin{bmatrix}0\\ -x_2x_3^2\\ 0\end{bmatrix},\ Q_1 = \begin{bmatrix}2 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 2\end{bmatrix},\ r_1 = 1 \tag{6.3.33}$$

and $\varepsilon = 0.20$. The nominal control and state trajectories are found from

$$u^0 = -R^{-1}B^T K x^0 = -1.41x_1^0 - 2.75x_2^0 - 11.16x_3^0 \tag{6.3.34}$$

$$\begin{aligned}\dot{x}_1^0 &= x_2^0\\ \dot{x}_2^0 &= -0.2x_2^0 + 2x_3^0\\ \dot{x}_3^0 &= -1.41x_1^0 - 2.75x_2^0 - 6.16x_3^0\end{aligned} \tag{6.3.35}$$

Table 6.1 summarizes the cost functional J for different initial conditions

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

using the linearization scheme, suboptimal control using approximate solution of the Hamilton-Jacobi-Bellman's equation, and the present method (Jamshidi, 1983). As seen from the table, the first-order near-optimum control proposed here results in the best overall performance. It is also noted that in view of Nishikawa *et al.* (1971) the optimum cost functional for the case $r = 0$ is approximated up to one term while for $r = 1$ it is approximated up to $2 + 1 = 3$ terms. Thus, it is not surprising that the $r = 1$ control law is closer to the exact optimum solution. Further comments regarding this point are given in Section 6.5.

6.3.2 Hierarchical Control via Interaction Prediction

Recently the hierarchical control of nonlinear systems has received a great deal of attention. The straight application of goal coordination or interaction prediction to nonlinear systems does not, in general, provide a global sufficiency condition for optimum solution (Singh, 1980). The basic problem is that as a result of the lack of sufficiency condition in goal coordination, for example, there is a possibility for a "duality gap", i.e., a difference between the primal problem minimum and the dual problem maximum. In Chapter 4 where linear systems with linear interaction were considered it was deduced that the "duality gap" was zero. The fundamental reason behind this deduction was the Strong Lagrange Duality theorem (Geoffrion, 1971a), which requires that the interaction constraints be convex. However, the application of goal coordination to nonlinear systems,

Table 6.1 Comparison of the Cost Functional of Example 6.3.1 Using Three Methods

	Cost function J Initial State x_0		
	$\begin{bmatrix} 0.10 \\ 0.10 \\ 0.10 \end{bmatrix}$	$\begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.50 \\ 0.50 \end{bmatrix}$
Linearization {White and Cook, 1973}	0.2293	1.1297	4.1941
Approximate Hamilton-Jacobi-Bellman's equation (Garrard, *et al.* 1967)	0.2288	1.1259	4.0627
Algorithm 6.3			
$r = 0$	0.2165	1.0722	3.9520
$r = 1$	0.2162	1.0712	3.9349

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

used by some authors (Bauman, 1968; Smith and Sage, 1973) does not have a guaranteed sufficiency condition, since the nonlinear interaction constraints are not convex in general.

There are at least two approaches where for certain special cases the goal coordination can be used where no duality gap would exist. First consider the following optimization problem:

$$\begin{aligned} &\text{Maximize } J(x) \\ &\text{Subject to } f(x) \in \phi \\ &\qquad\qquad x \in X \end{aligned} \tag{6.3.36}$$

where $J(x)$ is a convex function, X is a convex set, and ϕ has either of the two forms: $\phi = \{0\}$ or $\phi = \{\upsilon\colon \upsilon \le 0\}$, $\upsilon \in R^m$. The dual problem is

$$\underset{\alpha \ge 0}{\text{Minimize}} \quad \underset{x \in X}{\text{Maximize}} \left\{ J(x) + \alpha^T f(x) \right\} \tag{6.3.37}$$

For the case when ϕ $\{\upsilon\colon\upsilon \le 0\}$, problems (6.3.36) and (6.3.37) have the same solutions if $f(x)$ is a convex vector function (Javdan, 1976a). If $\phi = \{0\}$, the two problems can have the same solution only if $f(x)$ is linear (Whittle, 1971). Therefore, more general convex constraints can be handled if the optimization problem has inequality constraints instead of equality constraints. One way to guarantee a solution for problems with equality constraints is to have all inequality constraints binding at optimum. More specifically, consider a modified version of the following problem:

$$\begin{aligned} &\text{Maximize } J(x) + u^T f(x) \\ &\text{Subject to } f(x) \le 0,\ x \in \text{X},\ \text{u} \in R^m \end{aligned} \tag{6.3.38}$$

If a vector u can be found such that inequality constraints in (6.3.38) are binding at optimum, then the dual coordination would work for such a problem (Singh, 1980). Although it may not be an easy task to find a u for a general convex problem which makes all inequalities binding at optimum, some attempts have been made to find a suitable u for certain special cases. Javdan (1976a) has readily obtained a u for certain quadratic constraints. The class of quadratic problems considered is one in which each second-order nonlinear term appears only once in the equality constraints. More insight in this approach can be obtained by referring to Javdan (1976a,b) or Singh (1980). An alternative approach for handling equality constraints is suggested by Simmons (1975) which assumes that there exist buffer stores between the interconnected subsystems. The basic idea here, due to Whittle (1971), is that one satisfy each interconnection con-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

straint on the average instead of instantaneously. Under such average satisfaction of interconnection constraint, the strong Lagrangian principle works and the corresponding goal coordination solution is the best one. There are several examples where the goal coordination has been successfully applied. Bauman (1968) has applied it to a sliding mass system with linear interaction constraint and has used the modification on the constraints discussed in Section 4.2.6.a to avoid singular solutions. Singh (1980) has considered the same system and has used the modification of Titli *et al.* (1975), discussed in Section 4.3.3.b, to curb singularities. The remainder of this section is devoted to a new application of the interaction prediction method to nonlinear systems due to Hassan and Singh (1976) and Singh (1980). The method is applied to a sixth-order power system.

Following the linearization procedure of Section 6.3.1, for the sake of computational convenience, assume that the origin is an equilibrium point and linearize a large-scale nonlinear system $\dot{x} = h(x,u)$, $x(0) = x_0$ about it to obtain

$$\dot{x} = A_d x + B_d u + f(x,u), \quad x(0) = x_o \tag{6.3.39}$$

where

$$\begin{aligned} f(x,u) &\triangleq A_o x + B_o u + h(x,u) - Ax - Bu \\ A &= \partial h(\cdot)/\partial x^o, \quad B = \partial h(\cdot)/\partial u^o \end{aligned}$$

similar to what is defined in (6.3.3b), A_d, A_0, B_d, and B_0 are, respectively, block-diagonal and block-off-diagonal parts of A and B, i.e.,

$$A_d \triangleq \text{Block-diag}(A_1, A_2, \ldots, A_N), \; B_d = \text{Block-diag}(B_1, B_2, \ldots, B_N)$$

$$A_0 = \begin{bmatrix} 0 & A_{12} & \cdots & A_{1N} \\ A_{21} & 0 & & \\ \vdots & & \ddots & \vdots \\ A_{N1} & & \cdots & 0 \end{bmatrix}, \quad B_o = \begin{bmatrix} 0 & B_{12} & \cdots & B_{1N} \\ B_{21} & 0 & & \\ \vdots & & \ddots & \vdots \\ B_{N1} & & \cdots & 0 \end{bmatrix} \tag{6.3.40}$$

The optimal control problem is to find a control u which satisfies (6.3.39) while minimizing a quadratic cost function

$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)\, dt \tag{6.3.41}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

As in Chapter 4, it is assumed that the large-scale nonlinear system (6.3.39) can be decomposed into N small-scale subsystems:

$$\dot{x}_i = A_i x_i + B_i u_i + f_i(x,u), \qquad x_i(0) = x_{io} \tag{6.3.42}$$

Furthermore, for the sake of the present discussion, it is assumed that the cost (6.3.41) is separable, i.e.,

$$J = \sum_{i=1}^{N} J_i = \sum_{i=1}^{N} \frac{1}{2} \int_0^{t_f} \left(x_i^T Q_i x_i + u_i^T R_i u_i \right) dt \tag{6.3.43}$$

where $Q = \text{Block-diag}(Q_1, Q_2, \ldots, Q_N)$ and $R = \text{Block-diag}(R_1, R_2, \ldots, R_N)$ are, respectively, assumed to be positive-semidefinite and positive-definite.

Hassan and Singh (1976) and Singh (1980) suggest a two-level hierarchical structure in which the second level (coordinator) is assumed to provide composite state and control vectors $x = x^o$ and $u = u^o$ to the first level, which in view of (6.3.42), reduces the problem to N independent problems:

$$\text{Minimize } J_i = \frac{1}{2} \int_0^{t_f} \left(x_i^T Q_i x_i + u_i^T R_i u_i \right) dt \tag{6.3.44}$$

$$\text{Subject to } \dot{x}_i = A_i x_i + B_i u_i + f_i(x^o, u^o), \quad x_i(0) = x_{io} \tag{6.3.45}$$

where x^o and u^o are initial values for the equivalent state and control vectors, respectively. These vectors, as will be seen shortly, would be periodically updated through an interaction prediction-type algorithm at the second level. Singh (1980) and Hassan and Singh (1976) further suggest adding two additional penalty terms in the cost function (6.3.44),

$$\begin{aligned} J_i^o &= \frac{1}{2} \int_0^{t_f} \left\{ x_i^T Q_i x_i + \left(x_i - x_i^o\right)^T V_i \left(x_i - x_i^o\right) + u_i^T R_i u_i + \left(u_i - u_i^o\right)^T W_i \left(u_i - u_i^o\right) \right\} dt \\ &\triangleq \int_0^{t_f} L_i\left(x_i, x_i^o, u_i, u_i^o\right) dt \end{aligned} \tag{6.3.46}$$

for regulation purposes. The solutions to the first-level problems (6.3.45), (6.3.46) follow the same Riccati formulation discussed in Section 6.3.1, except for two additional linear equality constraints:

$$x_i - x_i^o = 0, \quad u_i - u_i^o = 0 \tag{6.3.47}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Taking these two constraints into account, the ith subsystem Hamiltonian becomes

$$H_i^o = -L_i\left(x_i x_i^o, u_i, u_i^o\right) + p_i^T\left(A_i x_i + B_i u_i + f_i\left(x^o, u^o\right)\right) + \alpha_i^T\left(x_i - x_i^o\right) + \beta_i^T\left(u_i - u_i^o\right) \tag{6.3.48}$$

where p_i, α_i, and β_i are n_i-, n_i-, and m_i-dimensional co-state vectors, Lagrange multiplier vectors corresponding to the additional equality constraints (6.3.47). Writing the necessary conditions of the maximum principle, assuming $p_i = K_i x_i + g_i$, eliminating u_i; in the necessary conditions, and following a procedure similar to the derivations of sensitivity functions of (6.3.10)–(6.3.21), the optimum ith subsystem equation becomes

$$\dot{x}_i = \left(A_i - \tilde{S}_i K_i\right)x_i + B_i d_i\left(g_i, u_i^o, \beta_i\right) + f_i\left(x^o, u^o\right) \tag{6.3.49}$$

where K_i is the solution of the ith subsystem Riccati equation

$$\dot{K}_i = -K_i A_i - A_i^T K_i + K_i \tilde{S}_i K_i - \tilde{Q}_i, \quad K_i(t_f) = 0 \tag{6.3.50a}$$

$$\tilde{S}_i = B_i \tilde{R}_i^{-1} B_i^T, \quad \tilde{R}_i = R_i + W_i$$
$$\tilde{Q}_i = Q_i + V_i \tag{6.3.50b}$$

function

$$d_i(\cdot) = \tilde{R}_i^{-1}\left(B_i^T g_i + W_i u_i^o + \beta_i\right) \tag{6.3.51}$$

g_i is the solution of the ith subsystem adjoint vector equation

$$\dot{g}_i = -\left(A_1 - \tilde{S}_i K_i\right)^T g_i + K_i B_i \tilde{R}_i^{-1}\left(W_i u_i^o + \beta_i\right) + K_i f_i(x^o, u^o) - \alpha_i - V_i x_i^o,$$
$$g_i(t_f) = 0 \tag{6.3.52}$$

and the optimum local control is

$$u_i = -\tilde{R}_i^{-1} B_i^T K_i x_i + d_i\left(g_i\ u_i^o, \beta_i\right) \tag{6.3.53}$$

The relations (6.3.49), (6.3.50), (6.3.52), and (6.3.53) constitute the essen-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

tial computations at the first level, with the DMRE (6.3.50a) to be solved only once during all subsequent first-second level iterations. DMREs (6.3.50a) can be solved either by the "doubling" scheme (Davison and Maki, 1973) or by straight backward integration (Jamshidi, 1980).

The second-level problem is basically to try to find a set of Lagrange multipliers $\alpha_i(t)$, $\beta_i(t)$ for all $i = 1, 2, \dots, N$ and $0 \le t \le t_f$ such that the constraints (6.3.47) are satisfied. In other words, x_i^o, u_i^o, α_i, and β_i must be updated based on a prediction mechanism. In order to reach this mechanism, it is noted that the ith subsystem Hamiltonian (6.3.48) must also be extremized with respect to x_i^o, u_i^o, α_i, and β_i, i.e.,

$$\partial H_i^o / \partial \alpha_i = 0, \quad \partial H_i^o / \partial \beta_i = 0, \quad \partial H_i^o / \partial x_i^o = 0, \quad \partial H_i^o / \partial u_i^o = 0 \tag{6.3.54}$$

The above relations along with (6.3.47) give the desired interaction prediction iterations for the second level:

$$\alpha = \alpha(x, p) = V(x - x^o) + \left\{A_o^T + \left[\partial h(x^o, u^o)/\partial x^o\right]^T - A^T\right\}p \tag{6.3.55a}$$

$$\beta = \beta(u, p) = W(u - u^o) + \left\{B_o^T + \left[\partial h(x^o, u^o)/\partial x^o\right]^T - B^T\right\}p \tag{6.3.55b}$$

The above relations along with (6.3.47) give the desired interaction prediction iterations for the second level:

$$\begin{bmatrix} x^o \\ \alpha \\ u^o \\ \beta \end{bmatrix}^{l+1} = \begin{bmatrix} x^l \\ \alpha(x^l, p^l) \\ u^l \\ \beta(u^l, p^l) \end{bmatrix} \tag{6.3.56}$$

where the right-hand side of (6.3.56) is obtained by substituting the values of $x^l = \left(x_1^{Tl} x_2^{Tl} \cdots x_N^{Tl}\right)^T$, $u^l = \left(u_1^{Tl} u_2^{Tl} \cdots u_N^{Tl}\right)^T$ as well as x^{ol} and u^{ol} in (6.3.55). The interaction errors can be expressed by the following:

$$\text{Error}_x^l = \left\{\int_0^{t_f} \sum_{i=1}^N \left\|\Delta x_i^l\right\|^2 dt\right\}^{1/2} \tag{6.3.57}$$

$$\text{Error}_\alpha^l = \left\{\int_0^{t_f} \sum_{i=1}^N \left\|\Delta \alpha_i^l\right\|^2 dt\right\}^{1/2} \tag{6.3.58}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where

$$\Delta x_i^l = x_i^{ol+1} - x_i^{ol},\ \Delta\alpha_i^l = \alpha_i^{l+1} - \alpha_i^l \tag{6.3.59}$$

The application of the interaction prediction approach to nonlinear systems (6.3.1) is summarized by the following algorithm.

Algorithm 6.5. Hierarchical Control of Large-Scale Nonlinear Systems via Interaction Prediction

Step 1: Start with initial values x^o, u^o, α, and,β at level 2.

Step 2: Solve N matrix Riccati equations of type (6.3.50a) for K_i, $i = 1, 2, \ldots, N$, and store at level 1.

Step 3: Solve N adjoint vector equations of type (6.3.52) for g_i, $i = 1, 2, \ldots, N$, and store at level 1.

Step 4: Solve for x_i, u_i, $i = 1, 2, \ldots, N$, using (6.3.49) and (6.3.53), collate and store $x^T = \left(x_1^T x_2^T \ldots x_N^T\right)$, $u^T = \left(u_1^T u_2^T \ldots u_N^T\right)$ and transmit values to second level.

Step 5: Use the latest values of x, u, x^o, u^o to predict new values of x^o, u^o, α, and β using (6.3.56).

Step 6: Check whether the interaction errors (6.3.57)–(6.3.58) are within some tolerance ε. If not, go to Step 3.

Step 7: Stop.

The following example illustrates this algorithm.

Example 6.3.2. Consider the open-loop model of a sixth-order power system which was originally developed by Iyer and Cory (1971) and further considered by Mukhopadhyay and Malik (1973) and Jamshidi (1975):

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -c_1 x_2 - c_2 x_3 \sin x_1 - 0.5 c_3 \sin 2x_1 + x_5/M \\
\dot{x}_3 &= x_6 - c_4 x_3 + c_5 \cos x_1 \\
\dot{x}_4 &= k_1 u_1 - k_2 x_2 - k_3 x_4 \\
\dot{x}_5 &= k_4 x_4 - k_5 x_5 \\
\dot{x}_6 &= k_6 u_2 - k_7 x_6
\end{aligned} \tag{6.3.60}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where x_1 = rotor angle, x_2 = Park's transformation variable, x_3 = field flux linkage, x_4 = turbine's input steam power, x_5 = mechanical input torque, x_6 = normalized field voltage, u_1 = speeder gear setting, and u_2 = normalized exciter voltage. The cost function is

$$\begin{aligned} J = \int_0^2 \Big\{ & (x_1 - 0.7105)^2 + x_2^2 + 0.1(x_3 - 5.604)^2 \\ & + 0.5(x_4 - 0.8)^2 + 0.5(x_5 - 0.8)^2 + (x_6 - 2.645)^2 \\ & + 100(u_1 - 0.4236)^2 + 10(u_2 - 0.8817)^2 \Big\} dt \end{aligned} \tag{6.3.61}$$

which indicates that the equilibrium point of the system is

$$\begin{aligned} x^o &= (0.7105 \quad 0 \quad 5.604 \quad 0.8 \quad 0.8 \quad 2.645)^T \\ u^o &= (0.4236 \quad 0.8817)^T \end{aligned} \tag{6.3.62}$$

and the initial state was chosen as $x(0) = x^o$. It is desired to find the optimal control for this system by the hierarchical approach outlined in Algorithm 6.5.

SOLUTION: Numerical values for system (6.3.60) were calculated using the data by Mukhapadhyay and Malik (1973), $c_1 = 2.165$, $c_2 = 14$, $c_3 = -55.56$, $c_4 = 1.02$, $c_5 = 4.05$, $k_1 = 9.443$, $k_2 = 1.02$, $k_3 = 5$, $k_4 = 2.04$, $k_5 = 2.04$, $k_6 = 1.5$, $k_7 = 0.5$, and $M = 1.0$. The linearized matrices A and B are

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -51.2 & -2.16 & -9.02 & 0 & 29.6 & 0 \\ -2.61 & 0 & -1.02 & 0 & 0 & 1 \\ 0 & -1.02 & 0 & -5 & 0 & 0 \\ 0 & 0 & 0 & 2.04 & -2.04 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 9.443 & 0 \\ 0 & 0 \\ 0 & 1.5 \end{bmatrix} \tag{6.3.63}$$

controlengineers.ir
@controlengineers

The system was decomposed into two subsystems of orders $n_1 = 4$ and $n_2 = 2$ with $m_1 = m_2 = 1$. The solution, in accordance with Algorithm 6.5, was followed by solving two matrix Riccati equations using a fourth-order Runge-Kutta method and the elements of the corresponding Riccati matrices were fitted to third-order polynomials. The two subsystems Riccati formulations are summarized here.

Subsystem 1:

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -51.2 & -2.16 & -9.02 & 0 \\ -2.64 & 0 & -1.02 & 0 \\ 0 & -1.02 & 0 & -5 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 9.443 \end{bmatrix},$$

$$\tilde{R}_1 = 100, \quad Q_1 = \mathrm{diag}(1 \quad 1 \quad 0.1 \quad 0.5)$$

$$K_1(t) = \begin{bmatrix} 9.94 - 2.4t + 4.5t^2 - 2.9t^2 & -0.15 + 0.8t - 1.25t^2 + 0.43t^2 & 1.62 - 0.18t + 0.44t^2 - 0.38t^3 & 0.03 - 0.024t + 0.043t^2 - 0.02t^3 \\ ------ & 0.2 - 0.071t + 0.133t^2 - 0.074t^2 & 10^{-2}\left(-5.8 + 2t - 3t^2 + 1.7t^2\right) & 10^{-3}\left(-3.3 + 3.4t - 5.8t^2 + 2.4t^3\right) \\ ------ & ------ & 0.71 - 0.07t + 0.02t^2 - 0.08t^3 & 10^{-3}\left(5 - 4.25t + 7.5t^2 - 3.3t^3\right) \\ ------ & ------ & ------ & 10^{-2}\left(5.3 - 4t + 7.3t^2 - 3.3t^3\right) \end{bmatrix} \tag{6.3.64}$$

Subsystem 2.

$$A_2 = \begin{bmatrix} -2.04 & 0 \\ 0 & -0.5 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \quad \tilde{R}_2 = 10, \quad \tilde{Q}_2 = diag(0.5 \quad 1)$$

$$K_2(t) = \begin{bmatrix} 0.127 - 0.064t + 0.123t^2 - 0.06t^3 & 0 \\ 0 & 0.78 - 0.11t + 0.015t^2 - 0.77t^3 \end{bmatrix} \tag{6.3.65}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The algorithm was simulated on LSSPAK. The first-level problem is essentially to solve for the two subsystems, adjoint vectors $g_i(t)$, $i = 1, 2$, by integrating (6.3.50) and (6.3.52) backward in time, integrating (6.3.49) for $x_i(t)$, $i = 1, 2$, forward in time, and evaluating $u_i(t)$ using (6.3.53). The second-level problem follows (6.3.56) and is obtained for this example from

$$x^{ol+1} = x^{ol} \tag{6.3.66a}$$

$$\alpha^{l+1} = V(x - x^o) + \left\{A_o^T - A^T + \left[\partial h(x^{ol}, u^{ol})/\partial x^{ol}\right]^T\right\}p^l \tag{6.3.66b}$$

$$u^{ol+1} = u^{ol} \tag{6.3.66c}$$

$$\beta^{l+1} = W(u - u^o) + \left\{B_o^T - B^T + \left[\partial h(x^{ol}, u^{ol})/\partial u^{ol}\right]^T\right\}p^l \tag{6.3.66b}$$

where

$$\partial h(\cdot)/\partial x^o$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -c_2 x_2^o \cos x_1^o - c_3 \cos 2x_1^o & -c_1 & -c_2 \sin x_1^o & 0 & 1/M & 0 \\ -c_5 \sin x_1^o & 0 & -c_4 & 0 & 0 & 1 \\ 0 & -k_2 & 0 & -k_3 & 0 & 0 \\ 0 & 0 & 0 & k_4 & k_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -k_7 \end{bmatrix} \tag{6.3.67a}$$

$$\partial h(\cdot)/\partial u^o = B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ k_1 & 0 \\ 0 & 0 \\ 0 & k_6 \end{bmatrix} \tag{6.3.67b}$$

The interaction prediction method converged in 14 iterations. Typical responses, such as rotor angle $x_1(t)$, speeder gear setting $u_1(t)$, and exciter voltage versus time are shown in Figures 6.4 and 6.5. The results are in close agreement with the previous one-level optimization methods, such as quasilinearization (Mukhopadhyay and Malik, 1973) and parameter imbedding (Jamshidi, 1975).

controlengineers.ir

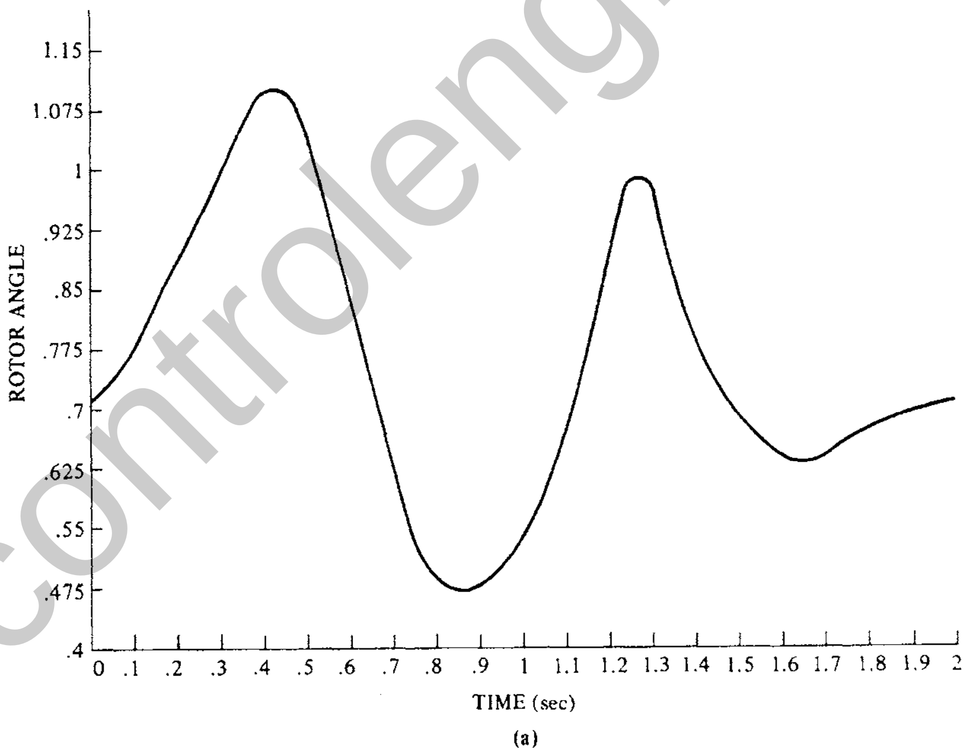
اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The interaction prediction methods seem to be very promising due to their simple computational requirements. Below a brief outline of the convergence property of the method is given. Singh (1980) and Hassan and Singh (1976) have established the following conditions:

1. $\|G\| < 1$.
2. (x, u, α, β) and $h(x, u)$ are bounded functions of time.
3. $h(x, u)$ is a continuously differentiable function of x and u on $(0, t_f)$ and its derivatives are bounded, which provides a final time t_f such that the second-level iterations (6.3.56) converge (6.3.68).

In (6.3.68). the matrix G is defined by

$$G \triangleq \begin{bmatrix} 0 & 0 & 0 & 0 \\ V & 0 & 0 & 0 \\ 0 & 0 & \tilde{R}^{-1}W & -\tilde{R}^{-1} \\ 0 & 0 & W - W\tilde{R}^{-1}W & W\tilde{R}^{-1} \end{bmatrix} \tag{6.3.69}$$

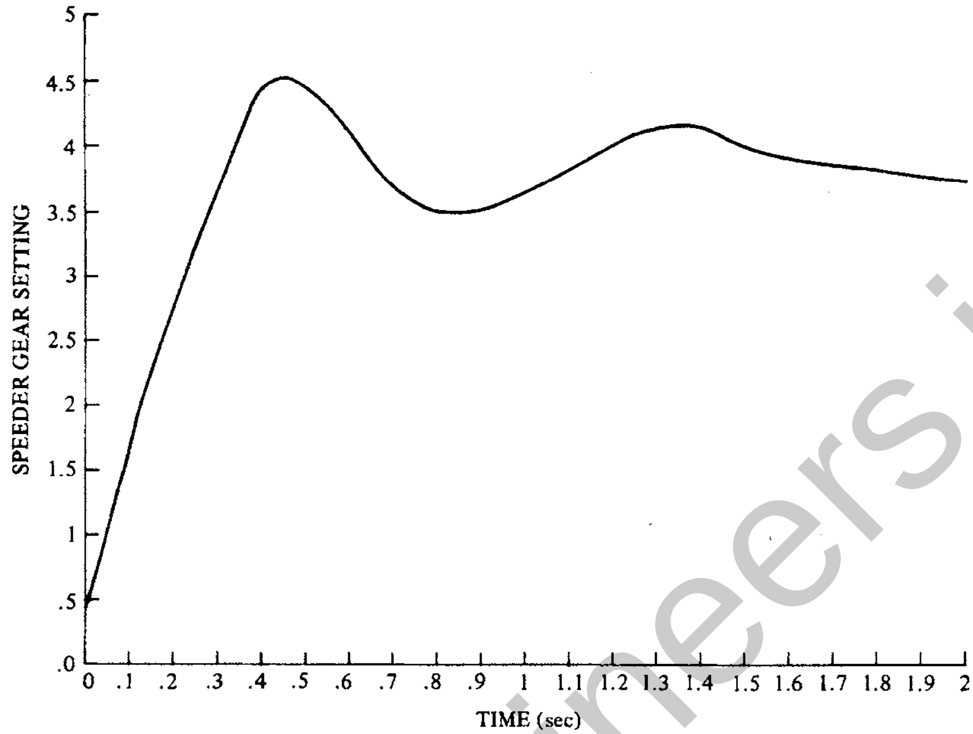


(a) rotor angle x1(t)

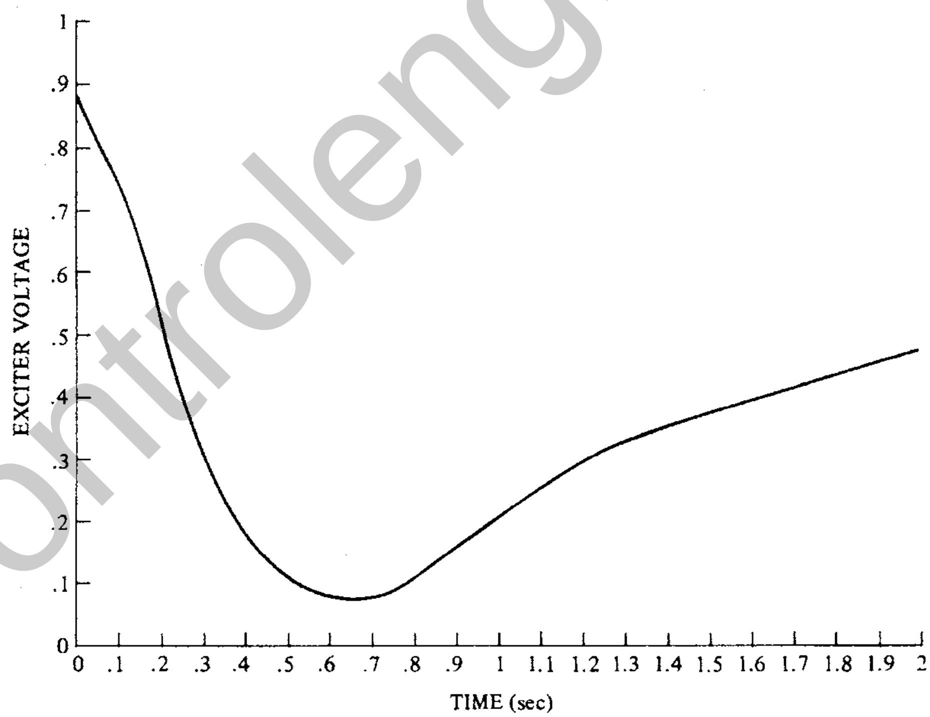
Figure 6.4. Typical responses for the power system of Example 6.3.2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(b) speeder gear setting $u_1(t)$



(c) exciter voltage $u_2(t)$

Figure 6.5. Typical responses for the power system of Example 6.3.2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It is noted by Singh and Hassan (1979) in their response to the comments of Mora-Camino (1979) that it is possible to choose W in $\tilde{R} = R + W$ large enough as compared to V in $\tilde{Q} = Q + V$ such that desirable convergence is achieved. This trend was verified in Example 6.3.2 for different (W, V) combinations.

There are two definite advantages to using the interaction prediction method for the hierarchical control of nonlinear systems. One is the fact that, unlike goal coordination, this approach converges to the optimum solution provided that a long enough final period t_f is chosen, as pointed out by Singh (1980). The second advantage is that the first-level problems are simple low-order linear ones which are centered around the solutions of accompanying matrix Riccati equations for all subsequent iterations. Moreover, the second-level problem consists of a series of simple substitutions.

On the other hand, this approach gives a local optimal solution unlike the goal coordination, which would give a global solution if there is no duality gap. The computational experiences of Singh (1980), Hassan and Singh (1976) and Hassan *et al.* (1979) indicate that the solution for a third-order synchronous machine example is no worse than the gradient and quasilinearization methods. This opinion is also shared by this author for the sixth-order example, where the results were quite comparable to the quasilinearization method of Mukhopadhyay and Malik (1973) and the imbedding method of Jamshidi (1975). The other basic shortcoming is the limitation of the quadratic form of the cost function. This difficulty can, however, be overcome by expanding a general cost function's integrand into a quadratic form and a nonlinear function. This scheme, suggested by Singh and Hassan (1978), is briefly considered.

To accompany the nonlinear system (6.3.36), let the following be the cost function which is to be minimized:

$$J = \int_0^{t_f} \gamma(x, u)\, dt \tag{6.3.70}$$

which can be modified by

$$J^o = \frac{1}{2}\int_0^{t_f} \left\{ x^T Q x + u^T R u + 2\Gamma(x, u) + \left(u - u^o\right)^T W\left(u - u^o\right) \right\} dt \tag{6.3.71}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $\Gamma(x,u)\overset{\Delta}{=}\gamma(x,u)-\frac{1}{2}\left(x^TQx+u^TRu\right)$ and the last term in (6.3.71) is added to help the convergence to the "desired" control u^o. Singh and Hassan (1978) have suggested using the desired point (x^o, u^o) to fix $\Gamma(x^o, u^o)$, very much like fixing $f(x^o, u^o)$ in the system dynamics as seen in (6.3.45). Without going through the derivations of the optimal hierarchical control, we simply give the resulting equations which need to be solved for first- and second-level problems in composite but diagonal form. The matrix Riccati and vector adjoint equations for the first-level problem become

$$\dot{K}=-KA_d-A_d^TK+K\tilde{S}K-Q, \quad K(t_f)=0 \tag{6.3.72}$$

$$\dot{g}=-\left(A_d-\tilde{S}K\right)^Tg+KB_d\tilde{R}^{-1}\left(Wu^o+\beta\right)+Kf\left(x^o,u^o\right)-\alpha, \quad g(t_f)=0 \tag{6.3.73}$$

where $\tilde{S}=B_d\tilde{R}^{-1}B_d^T$, $\tilde{R}=R+W$, A_d and B_d are defined in (6.3.40), K = Block-diag$(K_1, K_2, \dots, K_N)$, and g = $\left(g_1^Tg_2^T\cdots g_N^T\right)^T$. The control vector and the closed-loop system equation for the first level would be identical to (6.3.53) and (6.3.49), respectively. The update equations for the second level differ from (6.3.55) as indicated here:

$$\alpha=\alpha(p)=Qx^o-\partial\gamma\left(x^o,u^o\right)/\partial x^o+\left(\partial h^T/\partial x^o-A_d^T\right)p(t) \tag{6.3.74}$$

$$\beta=\beta(u,p)=W\left(u-u^o\right)-\partial\gamma\left(x^o,u^o\right)/\partial u^o+Ru^o+\left(\partial h^T/\partial u^o-B_d^T\right)p(t) \tag{6.3.75}$$

Singh (1980) has presented the heirarchical control of other problems of interest such as "tracking" or feedback forms for the local controllers. Some of these approaches are again considered as problems and are not formally treated here since their derivations are similar.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

6.4 Bounds on Near-Optimum Cost Functional

Thus far, several optimum and near-optimum control techniques have been presented. As a result of the various complexities of large-scale systems, an exact optimal control is neither computationally desirable nor even possible in many cases. Thus, whether it is through model reduction, structural decomposition, approximation of nonlinearities, time delays, or parameter sensitive series expansion, a degree of degradation of the optimal cost function is the price that one has to pay. The general problem of the performance index sensitivity with respect to the variations in plant parameters, initial conditions, uncertainties in measurements, and the model has been considered by several authors (Popov, 1960; Dorato, 1963; Rissanen, 1966; Werner and Cruz, 1968; McClamroch *et al.*, 1969; Kokotović and Cruz, 1969; Kokotović, 1972; Bailey and Ramapriyan, 1973; Weissenberger, 1974; Malek-Zavarei and Jamshidi, 1975; Šiljak and Sundareshan, 1976; Laub and Bailey, 1976; Šiljak, 1978a, b; Singh, 1980). Dorato (1963) has presented a procedure for analysis of performance index sensitivity to small plant parameter variations by introducing the "performance index sensitivity vector." Werner and Cruz (1968) have established a relation between approximations in optimal control and the optimal performance index of the system under the influence of uncertain parameters. It has been asserted that a truncation up to the rth term of the series expansion of control in plant parameters corresponds to a truncation up to the $(2r + 1)$th terms of the series expansion of the optimal performance index. McClamroch *et al.* (1969) have studied the sensitivity of the optimal performance index to large plant parameter variations in linear systems by introducing the concept of ρ-sensitivity. A system is said to be *ρ-sensitive* with respect to a certain "class" of variations if the value of the performance index does not increase by more than a factor of ρ for variations in that class. Bailey and Ramapriyan (1973) have studied the bounds of performance with respect to model uncertainties due to weak coupling. Aoki (1968, 1971, 1978) has given bounds on the performance index degradation due to model aggregation. Kokotović *et al.* (1969b), Kokotović and Cruz (1969), and Kokotović (1972) have extended the assertion of Werner and Cruz (1968) for linear weakly coupled systems. Malek-Zavarei and Jamshidi (1975) have extended the p-sensitivity approach of McClamroch *et al.* (1969) to take into account the sensitivity of the performance index with respect to time delays.

In this section some of the results on the bounds and degradation of the near-optimum cost functional due to aggregation, perturbation, expansion of sensitivity functions, and nonlinearities will be discussed.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

6.4.1 Near-Optimality Due to Aggregation

The aggregation methods were first introduced in Section 2.2 within the context of large-scale system modeling. The optimal control of large-scale systems through aggregation was considered in Section 6.2.1. In this section, bounds on the cost function of near-optimum control due to aggregation (Aoki, 1968, 1978) are presented.

The application of the near-optimum control $u^a(t)$ in (6.2.15) to the full model $\dot{x} = Ax + Bu$, a cost function (6.2.2), and a weighing matrix $Q = C^T Q_a C$ in (6.2.14) results in a value

$$J^a = \tfrac{1}{2} x_o^T P_a x_o \tag{6.4.1}$$

where x_o is the initial state and P_a is the positive-definite solution of the following Lyapunov equation:

$$S_a^T P_a + P_a S_a + G_a = 0 \tag{6.4.2}$$

where $S_a = A - BF_a$, $G_a = F_a^T R F_a + Q$, F_a is defined by (6.2.15), and C is the aggregation matrix. The optimal cost function is

$$J^* = \tfrac{1}{2} x_o^T K_f x_o \tag{6.4.3}$$

where K_f is the Riccati matrix of the full model (6.2.9). If we let $P_a = K_f + D$, it can be shown that D satisfies (Aoki, 1971)

$$S_a^T D + DS_a + \Delta S \Delta = 0 \tag{6.5.4a}$$

where $S = BR^{-1}B^T$ *and* $\Delta = K_f - K_l = K_f - C^T K_a C$, satisfying

$$0 = S_l^T \Delta + \Delta S_l - \Delta S \Delta + Q - C^T Q_a \tag{6.5.4b}$$

with Q_a defined in (6.2.14) and $S_l = A - SK_l$. If all the eigenvalues of S_a have negative real parts, i.e., if the aggregated control u_a in (6.2.15) stabilizes the full model, then a positive-definite (or positive-semidefinite) D can be uniquely determined from (6.4.4a). The right-hand side of (6.4.4b) is nonnegative-definite, and from the relation of Δ in (6.4.4b) with respect to D in (6.4.4a), it follows that

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$D = P_a - K_f \geq 0 \quad \text{or} \quad K_f \leq P_a \tag{6.4.5a}$$

and

$$\Delta = K_f - K_l \geq 0 \quad \text{or} \quad K_f \geq K_l \tag{6.4.5b}$$

The relations (6.4.5) indicate that

$$J^l \leq J^* \leq J^a \tag{6.4.5c}$$

which is the desired bound on the cost function. The following example illustrates this bound condition on J^* while J^l in (6.4.5c) is $J^l = \frac{1}{2} x_o^T K_l x_o$.

Example 6.4.1. Consider a fifth-order system:

$$\dot{x} = \begin{bmatrix} -1 & 0 & 0.01 & 0.05 & 0.25 \\ 0 & -4 & 0 & 0.45 & 0.1 \\ -0.088 & 0.2 & -10 & 0 & 0.22 \\ 1 & 0 & 0.075 & -4 & 0.05 \\ 0.11 & 0.2 & 0.999 & 0.44 & -3 \end{bmatrix} x + \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \\ 0.5 & 0.9 \\ 2 & 0.75 \\ 1 & 1 \end{bmatrix} u \tag{6.4.6}$$

The eigenvalues of matrix A are -10.03, -0.952, -0.2996, -4.073, and -3.95. A third-order aggregated model for this system consisting of mode two, an average of first and fourth modes, and an average of third and fifth modes, can be obtained by choosing an aggregation matrix

$$C = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \tag{6.4.7a}$$

The aggregated matrices F, G are obtained from (2.2.7) and (2.2.6a):

$$(F, G) = \left(\begin{bmatrix} -1.975 & 0 & 0.1925 \\ 0.45 & -4 & 0.1 \\ 0.231 & 0.2 & -5.8905 \end{bmatrix}, \begin{bmatrix} 1.5 & 0.625 \\ 0 & 1 \\ 0.75 & 0.95 \end{bmatrix} \right) \tag{6.4.7b}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The weighting matrices are chosen as $Q = I_5$ and $R = I_2$. It is desired to find the bounds on the optimal cost function.

SOLUTION: The optimal cost equation (6.4.3), was obtained by solving a fifth-order AMRE using (A, B, Q, R) and an initial state x_o, $= (1 \quad -1 \quad 0 \quad -1 \quad 1\ )^T$, i.e., $J^* = 0.4034917$. Next a third-order AMRE using $(\ F, G, Q_a, R)$ was solved for the aggregated system Riccati matrix K_a , and with the help of F_a defined by (6.2.15) and Q_a defined by (6.2.14), the Lyapunov equation (6.4.2) can be solved for P_a . The resulting upper bound, given by (6.4.1), turns out to be $J^a = 0.4145285$. In a similar fashion, J^l turns out to be $J^l = 0.0810435$. Thus, aggregation done on system (6.4.6) results in a bound $0.0810435 < J^* < 0.4145286$ or a 3% degradation of the cost function.

6.4.2 Near-Optimality Due to Perturbation

Perturbation, in both regular and singular forms, was introduced in Section 2.3 as a modeling approach. The near-optimum controls were obtained through perturbation in Section 6.2.2. Algorithm 6.1 gave a near-optimum solution to a set of two weakly coupled (ε-coupled) subsystems. It was shown that the solution to the full model Riccati equation is expressed as a MacLaurin series expansion in ε. For the suboptimality caused by ε-coupling, it has been shown (Kokotović, 1972; Kokotović and Cruz, 1969; Kokotović *et al*., 1969b) that for an rth-order expansion of the Riccati matrix, the resulting performance index J^a approximates the optimal cost J^* up to $(2r + 1)$ terms of its expansion. For instance, in Example 6.2.4, for a three-term expansion of the Riccati matrix P in (6.2.49), the approximate cost J^a matches the first seven terms of the optimal cost. The remainder of this section is devoted to the performance bounds obtained by Ramapriyan (1970) and Bailey and Ramapriyan (1973). Consider a system

$$\dot{x} = Ax + Bu, \qquad y = Cx \tag{6.4.8}$$

with a cost functional

$$J = \frac{1}{2}\int_0^\infty \left(y^T Q y + u^T R u\right) dt \tag{6.4.9}$$

The optimal control is $u^* = -R^{-1}B^T Kx \ = F^* x$ and the optimal cost is $J^* = \frac{1}{2} x_o^T K x_o$, where K is the solution of the following AMRE

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$A^T K + KA - KSK + C^T QC = 0 \tag{6.4.10}$$

with $S = BR^{-1}B^T$. Now assume that the system matrix A is decoupled into $A = A_n + A_o$, where A_n and A_o are the nominal (or block-diagonal) and uncertain (or off-diagonal) portions of A, respectively. The feedback control for the nominal case, i.e., $A = A_n$ is assumed to be $u^a = -R^{-1}B^T K_n x = -F_a x$, where K_n is the solution of nominal AMRE

$$A_n^T K_n + K_n A_n - K_n S K_n + C^T QC = 0 \tag{6.4.11}$$

It is known that the approximate cost function $J^a = \frac{1}{2} x_o^T P_a x_o$ where P_a is the solution of the following Lyapunov equation

$$P_a S_a + S_a^T P_a + G_a = 0 \tag{6.4.12}$$

where $S_a = A - BF_a$ *and* $G_a = F_a^T RF_a + C^T QC$. Bailey and Ramapriyan (1973) have proved that the optimal cost J^* is bounded by

$$p_2 J^n \le J^* \le J^a \le p_1 J^n \tag{6.4.13}$$

where $J^n = \frac{1}{2} x_o^T K_n x_o$, $p_1 = \left(1 + \lambda_M\right)^{-1}$ and $p_2 = \left(1 + \lambda_m\right)^{-1}$ and $\lambda_M = \max_i$

$\{\lambda_i(H_n)\}$, $\lambda_m = \min_i\{\lambda_i(H_n)\}$ with

$$H_n = \left(F_a^T RF_a + C^T QC\right)^{-1}\left(K_n A_o + A_o^T K_n\right) \tag{6.4.14}$$

The following example illustrates the bound on the optimal cost J^*.

Example 6.4.2. Reconsider the fifth-order system of Example 6.4.1 and find the range on the cost function due to a 2×2 and 3×3 decomposition.

SOLUTION: Consider the system A matrix given by (6.4.6) rewritten as

$$A = A_n + A_o = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & -10 & 0 & 0.22 \\ 0 & 0 & 0.075 & -4 & 0.05 \\ 0 & 0 & 0.999 & 0.44 & -3 \end{bmatrix}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$+\begin{bmatrix} 0 & 0 & 0.01 & 0.05 & 0.25 \\ 0 & 0 & 0 & 0.45 & 0.1 \\ -0.088 & 0.2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0.11 & 0.2 & 0 & 0 & 0 \end{bmatrix} \tag{6.4.15}$$

The nominal or decoupled solution of the AMRE (6.4.11) using the same values for B, Q, and R as in Example 6.4.1 results in $J^n = 0.4030237$. The optimum cost function is already known to be $J^* = 0.4034917$. The approximate cost, i.e., when $u^a = -R^{-1}B^T K_n x$ is applied to the coupled unperturbed system (6.4.8), was found by solving a fifth-order Lyapunov Equation (6.4.12) using A and K_n, i.e., $J^a = 0.4116365$. The values of p, and P_2 turned out to be 1.2975 and 1.00, respectively. It is clear that the bound on J^* as defined by (6.4.13) does check, i.e., $p_2 J_n = 0.403237 < 0.4034917 = J^* \leq J^a = 0.4116365 \leq p_1 J^n = 0.5229232$. It must be noted that the cost function bound (6.4.13) developed by Bailey and Ramapriyan (1973) differs from the assertion of Werner and Cruz (1968) that J^a approximates J^* up to $2r + 1$ terms. The latter is not a bound condition in the sense discussed in this section.

6.4.3 Near-Optimality in Hierarchical Control

The degree of near-optimality in the hierarchical control systems can be estimated. The approach stems from the relation between the suboptimal indices of the primal and dual problems (Singh, 1980).

Let us assume that the optimum performance of a primal problem, defined in Section 4.3.1, be J^* and the corresponding near-optimal value be J^s; then it is known that

$$I^s \leq I^* = J^* \leq J^s \tag{6.4.15}$$

where I^* and I^s are the optimal and near-optimal performance indices of the dual problem, respectively. The bounds in (6.4.15) can be obtained by the following development due to Singh (1980). Consider the primal problem

$$\underset{u(t)}{\text{Minimize}} \;\; J = \frac{1}{2}\int_0^{t_f} \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt \tag{6.4.16}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\text{Subject to } \dot{x} = Ax + Bu, \quad x(0) = x_o \tag{6.4.17}$$

where x and u are the collated state and control vectors of the subsystems' states and controls x_i, u_i, $i = 1,2, \ldots, N$. The dual function of the primal problem (6.4.16)–(6.4.17) is

$$I = \underset{p(t)}{Max}\left\{ \underset{x(t),\,u(t)}{Min} \int_0^{t_f} \left[x^T Qx + u^T Ru + p^T (Ax + Bu - \dot{x})\right] dt \right\} \tag{6.4.18}$$

where $p(t)$ is the adjoint vector. In order to find an explicit value for I, the minimum value in (6.4.18) is evaluated next. Let the Hamiltonian of the primal problem be

$$H = \tfrac{1}{2}\left(x^T Qx + u^T Ru\right) + p^T (Ax + Bu)$$

and consider the necessary conditions for optimality:

$$\dot{p} = -\partial H/\partial x = -Qx - A^T p, \quad p(t_f) = 0 \tag{6.4.19}$$

$$0 = \partial H/\partial u = Ru + B^T p \tag{6.4.20}$$

Using the integration by parts, the last term of the integrand in (6.4.18) is expressed by

$$\int_0^{t_f} p^T \dot{x}\, dt = -p^T(0)x_o - \int_0^{t_f} x^T\left(Qx + A^T p\right) dt \tag{6.4.21}$$

Now using the expression for u in (6.4.20), substituting it in (6.4.18), and considering (6.4.21), the dual problem cost function becomes

$$I = \max_p \left\{ p^T(0)x_o + \int_0^{t_f} \left(x^T Qx - \tfrac{1}{2} p^T Sp\right) dt \right\} \tag{6.4.22}$$

where $S = BR^{-1}B^T$ and p is defined in (6.4.19).

In order to find lower bound I^s, the value of p is replaced by Kx, where K is the solution of the accompanying Riccati equation, and the maximization procedure is ignored (Singh 1980):

$$I^s = \frac{1}{2} x_o^T K x_o = \frac{1}{2} \int_0^{t_f} x^T(t)\left(Q - 2K(t)SK(t)\right) x(t)\, dt \tag{6.4.23}$$

controlengineers.ir @controlengineers

where $K(t)$ is the solution of the desired DMRE. The following example illustrates the duality approach for hierarchical control of a serially coupled system.

Example 6.4.3. Consider two first-order serially coupled systems

$$\dot{x}_1 = x_1 + u_1, \quad x_1(0) = 1$$
$$\dot{x}_2 = 2x_2 + u_2 + x_1, \quad x_2(0) = 1$$

with

$$J = \frac{1}{2}\int_0^5 \left(x_1^2 + x_2^2 + u_1^2 + u_2^2\right) dt$$

It is desired to find a bound on the optimum performance J^*.

SOLUTION: For this problem three DMREs were solved for the two subsystems and the composite system and were fitted to second-order polynomials, i.e.,

$$k_1(t) = 2.162 + 0.643t - 0.205t^2,$$
$$k_2(t) = 3.717 + 1.254t - 0.379t^2$$

$$K(t) = \begin{bmatrix} 2.42 + 0.78t - 0.24t^2 & 0.914 + 0.362t - 0.103t^2 \\ 0.914 + 0.362t - 0.103t^2 & 3.5 + 1.14t - 0.35t^2 \end{bmatrix}$$

The system was simulated on a personal computer using BASIC and the optimum and near-optimum costs J^*, I^s, J^s and states x^* and x^s were computed. The resulting values of performance indices turned out to be I^s = 4.01416, J^s = 5.15419, and J^* = 4.45102, which clearly indicates that

$$I^s = 4.01416 < I^* = 4.45102 = J^* < J^s = 5.15419$$

The results of this simple example indicate that the hierarchical near-optimum control of serial systems is a fairly reasonable one.

6.4.4 Near-Optimality in Nonlinear Systems

The last case whose degree of near-optimality is considered is a nonlinear system. The development in this section follows the work of Laub and Bailey (1976) which extends the linear case (Bailey and Ramapriyan, 1973)

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

discussed in Section 6.4.2. The general class of nonlinear systems examined in Section 6.3 is considered again.

Consider first a linear dynamic system

$$\dot{x} = Ax + Bu, \quad x(0) = x_o \tag{6.4.24}$$

and cost function

$$J(x_o, u) = \int_0^\infty g(x, u, t)\, dt \tag{6.4.25}$$

where $g(\cdot)$ is a nonnegative function, continuously differentiable in its arguments. It is assumed that a unique feedback control law

$$u = k_n(x(t), t), \qquad t \geq 0 \tag{6.4.26}$$

exists for the problem (6.4.24)–(6.4.25). Now let the system (6.4.24) be perturbed by a nonlinear term

$$\dot{x} = Ax + Bu + f(x, u, t), \qquad x(0) = x_o \tag{6.4.27}$$

with cost function (6.4.25). Further assume that another unique feedback control law

$$u = k(x(t), t), \qquad t \geq 0 \tag{6.4.28}$$

exists for (6.4.27) and (6.4.25). If the control (6.4.26) is applied to (6.4.27), a near-optimal cost functional results which is to be studied here. Let this cost function be J^a

$$J^a = J^a(x, t) = \int_t^\infty g\left(\phi\left(\tau; x, k_n, t\right), k_n, \tau\right) dt \tag{6.4.29}$$

while J^* and J^n are defined by

$$J^* = J^*(x, t) = \int_t^\infty g\left(\phi(\tau; k, t), k, \tau\right) dt \tag{6.4.30}$$

$$J^n = J^n(x, t) = \int_t^\infty g\left(\psi\left(\tau; k_n, t\right), k_n, t\right) dt \tag{6.4.31}$$

where $\psi(\cdot)$ and $\phi(\cdot)$ are the state trajectories of (6.4.24) and (6.4.31), respectively. Laub and Bailey (1976) have presented the following definition and subsequent lemma to estimate bounds on J^* similar to linear case (6.4.13).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Definition 6.1. The system (6.4.24) is said to be weakly coupled by a perturbation f if there exists a scalar parameter, $\mu > 0$ such that

$$g_n - J_x^{nT} f_n \geq \mu > 0 \tag{6.4.31}$$

holds for all x in its feasible domain. The terms f_n, and g_n, denote, respectively, $f(x, k_n, t)$ and $g(x, k_n, t)$ and J_x correspond to the gradient of J with respect to x.

Lemma 6.1. *If the system (6.4.24) is weakly perturbed by f, then there exists a parameter* p_1, *such that* $J^a \leq p_1 J^n$.
An estimate for p_1 *is given by*

$$p_1 = \min_{(t,x)\in L} g_n \big/ \left(g_n - J_x^{nT} f_n\right) = 1/(1-r) \tag{6.4.32}$$

where L is the feasible domain of (t,x) *and* $r = \max_{(t,x)\in L,\, x\neq 0}\left(J_x^{nT} f_n / g_n\right)$.
It is noted, however, that unlike the linear case, a lower bound is not available for here. A crude lower bound p_2 *has been suggested by Laub and Bailey (1976):*

$$p_2 = \min_{\substack{(t,x)\in L \\ x\neq 0}} g_n \big/ \left(g_n - J_x^{nT} f_n\right) \tag{6.4.33}$$

If the cost function (6.4.25) is assumed to be quadratic, i.e., $g(x, u, t) = \frac{1}{2}\left(x^T Q x + u^T R u\right)$, *then u in* (6.4.26) *becomes* $u = k_n(\cdot) = -R^{-1}B^T K$, *where K is the positive-definite solution of the system* (*A, B, Q, R*) *Riccati equation. The value* of $J^n = \frac{1}{2} x_o^T K x_o$ *and the weakly perturbed condition* (6.4.31) *becomes*

$$\frac{1}{2} x^T (Q + KSK) x - x^T K f\left(x, k_n(\cdot)\right) \geq \mu > 0$$

and the parameter r in (6.4.32) *would be*

$$r = \max_{\substack{x\in R^n \\ x\neq 0}} \left\{ x^T K f(\cdot) \Big/ \left[\tfrac{1}{2} x^T (Q + KSK) x\right]\right\}$$

controlengineers.ir
@controlengineers

The following example illustrates the above development on the near optimality of nonlinear systems.

Example 6.4.4. Consider the third-order nonlinear system of Example 6.3.1,

$$\begin{aligned} \dot{x}_1 &= x_2, \quad x_1(0) = 0.1 \\ \dot{x}_2 &= -0.2x_2 + 2x_3 - 0.2x_2x_3^2, \quad x_2(0) = 0.1 \\ \dot{x}_3 &= 5x_3 + u, \quad x_3(0) = 0.1 \end{aligned} \tag{6.4.34}$$

A near-optimum control for this system via the sensitivity method of Section 6.3.1 was designed (see Table 6.1) and an approximate value of $J^a = 0.2162$ was obtained for the cost function. It is desired to find a bound on the index J^a using Lemma 6.1.

SOLUTION: The near-optimum feedback control law $u_n = k_n(x(t))$ is given by (6.3.34), i.e.,

$$u_n = -R^{-1}B^T K x_n = \begin{bmatrix} -1.41 & -2.75 & -11.16 \end{bmatrix} x_n$$

stemming from a third-order AMRE whose solution is

$$K = \begin{bmatrix} 4.7135 & 4.3260 & 1.4050 \\ 4.3260 & 7.9540 & 2.7200 \\ 1.4050 & 2.7200 & 11.1154 \end{bmatrix}$$

using the matrices A,B in (6.4.34), $Q = 2I_3$, and $R = 1$. The unperturbed system cost turned out to be $J^n = 0.2036175$. Next, an unconstrained minimization method due to Rosenbrock (1960) was used twice to find p_1 and p_2 defined by (6.4.32), (6.4.33), and 30 function evaluations. The results turned out to be $p_1 = 1.7441$ and $p_2 = 0.25$, thus bounding J^a by

$$p_2 J^n = 0.051 < J^a = 0.2162 < 0.3552 = p_1 J^n$$

The above formulation can be used for hierarchical control of nonlinear systems as well. This plus other points are considered as problems at the end of this chapter.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

6.5 Computer-Aided Design

In this section another example is given to illustrate the use of LSSTB package for the near-optimum control design of a large-scale system.

CAD Example 6.5.1 In this CAD example, the fourth-order system of Example 2.2.6 which was reduced using balanced realization is used to design a near-optimum controller. To achieve this, a MATLAB .m file called *nearo* was written to extract the reduced-model from an already balanced system, find the aggregation matrix, find Q_a [see Equation (6.2.14)], find the near-optimum feedback, find the Riccati matrix, and simulate for an output response, all in the same piece of code

```
>> a = [0 0 0  –150; 1 0 0  –245; 0 1 0  –113; 0 0 1 –19l;
>> b = [4 1 0 0]´; c = [0 0 0 1]; d = 0;
>> lab,bb,cb] = balreal (a,b,c);
>> qt = diag ([.05 .1 .01 .1]); rt = 1;
>> [fbf, ric] = Iqr (a,b,qt,rt);
>> fbf

fbf =

          0.1735     0.0035     –0.2207    –0.1343
>> ac = a – b * fbf;
>> % simulate optimum system
>> t = [0: 0.1: 10]; uin = 0 * t; xo = [1 0 0 0]´;
>> [y,x] = Isim (ac,b,c,d,uin,t,xo);
>> % find 3 near-optimum control designs
>> [y1,x1] = nearo (ab,bb,cb,d,qt,rt,xo,1);
>> [y2,x2] = nearo (ab,bb,cb,d,qt,rt,xo,2);
>> [y3,x3] = nearo (ab,bb,cb,d,qt,rt,xo,3);
>> Plot (t,y,t,y1,t,y2,t,y3)
>> title (´Optimum & Near-Optimum Responses – Full & 3 Balanced
        Models´)
>> xlabel(´Time),ylabel(´y* & yn´)
```

Following is a listing of "nearo.m" file for this CAD example.

```
function [y,x] = nearo (ab,bb,cb,d,q,v,xo,k)
% dx/dt = Ax + Bu, y = cx + Du – balanced
% minimize J = 1/2 Int (x´Qtx + u´ Rtu´)dt
% Aggregated Model: dz/dt = Fz + Gu, A = Rx
% Extract kth order balanced reduced model
f = ab [1 :k, 1: kl; 9 = bb [1: k, :];
% Find aggregation matrix using controllability method
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers


```
% R = Wf  * Wat, Wf = [b, ab, ... a ∧ n – 1 b] and
% Wa = [g,fg, ..., f ∧ k – 1 q], + is pseudo-inverse
Wa = [bb, ab * bb,ab ∧ 2 * bb, ab ∧ 3 * bb];
if k = 1
      WF = [g, f  * g];
if k = 2
      Wf = [g,f * g,f ∧ 2 * g];
if k = 3
      Wf = [g,f * g,f ∧ 2 * g,f ∧ 3 * g1;
      r = wf * pinv (wa);
% linear quadratic regulator
      qa = inv (r * r´) * r * q * r´ * inv (r * r´);
      [f ba, rica] = lqr (f, q, qa, r);
% closed-loop matrix
      fb = fba * r;
      ac = a – b * fb;
t = [0: 0.l: 101; uin = 0 * t;
[y,x] = Isim (ac,bb,cb,d,t,xo);
end
```

Figure 6.6 shows a time response of optimum and three balanced method-based near-optimum controls. As seen, there is a definite improvement of performance for higher-order models as expected.

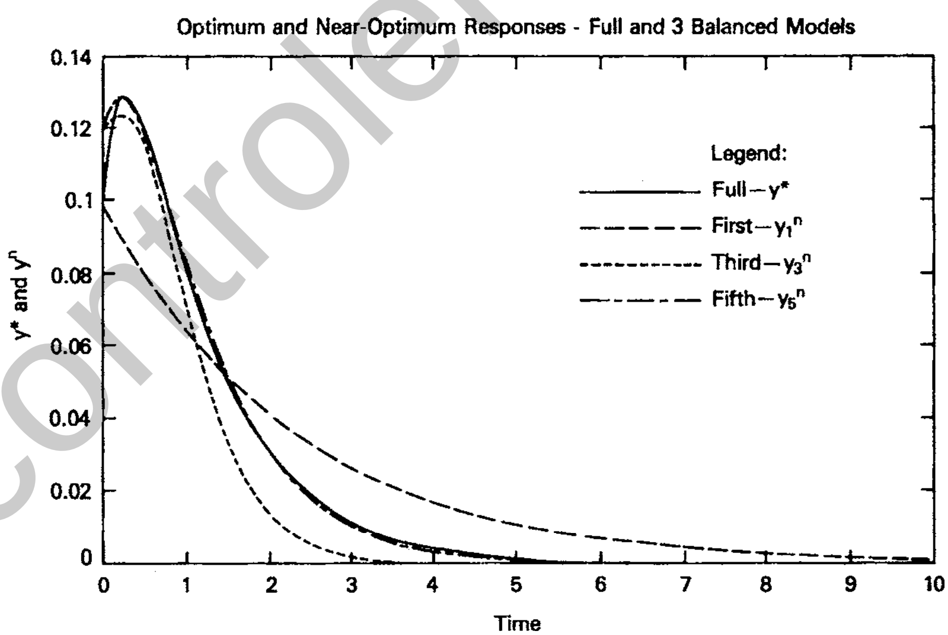


Figure 6.6. A time response comparison of an optimum and three near-optimum designs based on balanced methods for CAD Example 6.5.1.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Problems

6.1. Consider a fourth-order system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -10.1 & -9.07 & -12.41 & -7.2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

with $x^T(0) = (\,1 \quad -1 \quad 1 \quad -1)$ and cost function

$$J = \frac{1}{2}\int_0^\infty \left(x^T Q x + u^T R u\right) dt$$

with $Q = \text{diag}(1, 2, 3, 4)$ and $R = 1$. Find a second-order aggregated model using the modal aggregation method and then find the near-optimum feedback control law and the corresponding cost function.

6.2. Consider a singularly perturbed system

$$\dot{x} = -x + z + u, \qquad x(0) = 1$$
$$\varepsilon\dot{z} = -z + u, \qquad x(0) = 1, \qquad \varepsilon = 0.1$$

and a cost function

$$J = \frac{1}{2}\int_0^\infty \left(x^2 + z^2 + u^2\right) dt$$

Find (a) the left-hand boundary layer for this system and (b) the near-optimum control by methods of Section 6.2.2.b.

6.3. Consider an ε-coupled system

$$\dot{x} = \left[\begin{array}{cc|cc} -5 & 0.1 & 0 & -0.01 \\ 0 & -1 & 0 & 0 \\ \hline 0 & 0 & -0.5 & 0.1 \\ 0 & 0.05 & 0 & -0.1 \end{array}\right] x + \left[\begin{array}{c|c} 1 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 1 \end{array}\right] u$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

with a cost function

$$J = \frac{1}{2}\int_0^\infty \left(x^T \left[\begin{array}{c|c} Q_1 & 0 \\ \hline 0 & Q_2 \end{array}\right] x + u^T \left[\begin{array}{c|c} R_1 & 0 \\ \hline 0 & R_2 \end{array}\right] u \right) dt$$

with $Q_1 = 2I_2$, $Q_2 = I_2$, $R_1 = R_2 = 1$. Apply Algorithm 6.1 to find a near-optimum controller for this system.

6.4. CADLAB Find an optimal control for the system

$$\dot{x} = \begin{bmatrix} 0 & 0.1 & 1 \\ -1 & -1 & 0 \\ 0.5 & -0.5 & -0.5 \end{bmatrix} x + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} u$$

with a cost function

$$J = \frac{1}{2}\sum_{i=1}^{3}\int_0^\infty \left(x_i^T Q_i x_i + u_i^T R_i u_i \right) dt$$

where $Q_i = I_{n_i}$, $R_i = 2I_{m_i}$, with $n_i = m_i = 1$, $i = 1,2,3$, using Algorithm 6.3.

6.5. CADLAB Use Algorithm 6.5 to find a hierarchical control for the problem

$$\dot{x} = \begin{bmatrix} -1 & 0.2 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u + \begin{bmatrix} x_1^2 x_2 \\ x_2^2 x_1 \end{bmatrix},\ x(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$J = \int_0^2 \left(x_1^2 x_2^2 + x_1^3 + x_2^3 + u^2 x_1^2 + u^2 x_2^2 + x_1^2 + x_2^2 + u^2 \right) dt$$

Hint: Use the developments following Equations (6.3.70)–(6.3.75).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 7

Fuzzy Control Systems
Structures and Stability

7.1 Introduction

In this textbook, a class of control systems has been treated which encompasses many real-world problems. This class has been designated as large-scale systems. The starting point in this class of control systems, as in all other conventional control systems, is a set of differential (or difference) equations, like $\dot{x} = f(x,u)$ or $x(k+1) = f(x(k),u(k))$. Almost every aspect of a system's analysis and synthesis has been based on a mathematical model ever since the mid 1940s. The accuracy of any results in system theory depends heavily on the accuracy of the system's model and validity of the assumptions involved in obtaining the model.

Consider a large-scale system described by a set of nonlinear differential equations with time delay

$$\dot{x} = f(x(t), x(t-\tau), u(t)) \tag{7.1.1}$$

As a first-order approximation one may assume a set of nominal state and control vectors (x_n, u_n) and approximate (7.1.1) by

$$\dot{x} \cong A_1 x + Bu + A_2 x(t-\tau) \tag{7.1.2}$$

where A_1, B, and A_2 are appropriate Jacobian matrices.

Next, one may assume that the time delay τ is negligible, i.e., $\tau = 0$ and hence the system model (7.1.2) is further approximated by

$$\dot{x} \cong Ax + Bu \tag{7.1.3}$$

where $A = A_1 + A_2$. The system model, even though it is now linear and

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

without delay, has high dimension, hence to simplify the analysis and synthesis problems, we can decompose (7.1.3) into N subsystems as

$$\dot{x}_i \cong A_i x_i + B_i u_i + g_i(x,u) \tag{7.1.4}$$

where $g_i(x,u) \triangleq A_j x + B_j u,\ j \neq i$ is the interaction between ith and the other N-1 subsystems. Here, A_j and B_j are the original A and B matrices of (7.1.3) after the exclusion of block-diagonal submatrices A_i and B_i in (7.1.4). The dimensions of each subsystem is $n_i << n$ and, hence, it is desirable to work with each subsystem by ignoring the interaction vector function $g_i(\cdot)$, i.e., analyze and synthesize the n_i-dimensional linear decoupled nondelayed system

$$\dot{\bar{x}}_i \cong A_i \bar{x}_i + B_i u_i \tag{7.1.5}$$

where the over bar ($\bar{\ }$) represents the isolated (decoupled) subsystem state.

Now, let us review the process through which the mathematical model of a large-scale system (7.1.1) has been reduced to (7.1.5). Through this process by virtue of linearization, delay approximation, decomposition, and model reduction, each step and/or assumption has introduced a degree of *uncertainty* into the system, moving the model away from the true physical situation. The above discussion brings up another point—by making frequent, simplifying assumptions, the problem at hand has become too uncertain to be of practical use. The design and analysis of a large-scale control system should, in our opinion, be based on the best available *knowledge* instead of the simplest available model to treat uncertainties in the system. Therefore, a large-scale system is better treated by knowledge-based methods such as fuzzy logic, neural networks, genetic algorithms, expert systems, etc. The reader should note that by the above statement, we are *not* advocating that the designer should throw away the model and proceed with operational knowledge of the system. Quite to the contrary, the availability of a model, even an oversimplified one, is considered as an important part of the overall *knowledge base* of the system. The designer should pursue further information on the system and its operational practices to complete the knowledge base.

Like some model-based control systems whose designs are by trial and error, knowledge-based controller design is ad hoc at the present time. A gap exists between solid theoretical results such as stability, controllability, etc. and the real-time practical implementations of intelligent control systems, using the above techniques. Thus, a new school of thought—*intelligent control theory*— needs to be opened in control systems.

The aims of this chapter and the next one are to define one such area—

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

fuzzy control systems—and cover the latest results and development. Traditionally, an intelligent control system is defined as one in which classical control theory is combined with artificial intelligence (AI) and possibly OR (Operations Research). Due to this early definition, two approaches to intelligent control have been in use thus far. One approach combines expert systems in AI with differential equations to create the so-called expert control, while the other integrates discrete event systems (Markov chains) and differential equations (Wang, 1994a). The first approach, although practically useful, is rather difficult to analyze because of the different natures of differential equations (based on mathematical relations) and AI expert systems (based on symbolic manipulations). The second approach, on the other hand, has well-developed and solid theory, but is too complex for many practical applications. It is clear, therefore, that a new approach and a change of course are called for here. We begin with another definition of an intelligent control system. An intelligent control system is one in which a physical system or a mathematical model of it is being controlled by a combination of a knowledge base, approximate (humanlike) reasoning, and/or a learning process structured in a hierarchical fashion. Under this simple definition any control system which involves fuzzy logic, neural networks, expert learning schemes, genetic algorithms, genetic programing or any combination of these would be designated as *intelligent control*. Due to a vast body of results on fuzzy control systems and limited space, it has been decided to cover only fuzzy control systems in this text and leave the applications of neural networks and evolutionary algorithms to future books, although a very brief consideration of these appear at the end of the next chapter.

The present chapter is structured as follows. Section 7.2 is devoted to fuzzy control structures and basic features. In Section 7.3, stability of fuzzy control systems is discussed. The next chapter, Chapter 8, will treat fuzzy systems' adaptation and hierarchy.

7.2 Fuzzy Control Structures

Among the many applications of fuzzy sets and fuzzy logic, *fuzzy control* is perhaps the most common. Most industrial applications in Japan, the U.S., and Europe, fall under fuzzy control. The reasons for the success of fuzzy control are both theoretical and practical (Wang, 1994a).

From a theoretical point of view, a fuzzy logic rule base, as discussed in Sections 1.4.2, 2.6, and Appendix A, can be used to identify both a model, as a "universal approximation," as well as a nonlinear controller. The most relevant information about *any* system comes in one of three ways—a mathematical model, sensory input/output data, and human expert knowl-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

edge. The common factor in all these three sources is *knowledge*. For many years, classical control designers began their effort with a mathematical model and did not go any further in acquiring more knowledge about the system, i.e., designers put their entire trust in a mathematical model whose accuracy may sometimes be in question. Today, control engineers can use all of the above sources of information. Aside from a mathematical model whose utilization is clear, numerical (input/output) data can be used (see Algorithm 2.4 in Section 2.6) to develop an approximate model (input/output nonlinear mapping) as well as a controller, based on the acquired fuzzy IF-THEN rules.

Some researchers and teachers of fuzzy control systems subscribe to the notion that fuzzy controls should always use a model-free design approach and, hence, give the impression that a mathematical model is irrelevant. As indicated before, this author, however, believes strongly that if a mathematical model does exist, it would be the first source of knowledge used in building the entire knowledge base. From a mathematical model, through simulation, for example, one can further build the knowledge base. Through utilization of the expert operator's knowledge which comes in the form of a set of linguistic or semi-linguistic IF-THEN rules, the fuzzy controller designer would get a big advantage in using every bit of information about the system during the design process.

On the other hand, it is quite possible that a system, such as the large-scale systems of this text, is so complex that a reliable mathematical tool either does not exist or is very costly to attain. This is where fuzzy control or intelligent control comes in. Fuzzy control approaches these problems through a set of local humanistic (expert-like) controllers governed by fuzzy linguistic IF-THEN rules. In short, fuzzy control falls into the category of *intelligent controllers* which are not solely model-based, but also *knowledge-based*.

From a practical point of view, fuzzy controllers which have appeared in industry and in manufactured consumer products, are easy to understand, simple to implement, and inexpensive to develop. Because fuzzy controllers emulate human control strategies, they are easily understood even by those who have no formal background in control. These controllers are also very simple to implement. In fact, there are at least six easy approaches to realize a fuzzy controller through hardware and software in a real-time control application. These range from an implementation of a fuzzy look-up table all the way to an application-specific integrated circuit chip on an embedded fuzzy controller board. Finally, the development of a fuzzy controller is rather low, because fuzzy rules can be inferred through an 8-bit processor rather than 32-bit processors or DSP chips. Today, numerous cost-effective software and hardware tools are available (Jamshidi

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

et al., 1992). Moreover, since fuzzy controllers are task-oriented design approaches (unlike the set-point approach of classical control), cheaper (less accurate) sensors can be used.

7.2.1 Basic Definitions and Architectures

Having presented the above introduction to fuzzy control from both conceptual as well as practical points of view, we can now present details of fuzzy control.

A common definition of a fuzzy control system is that it is a system which emulates a human expert. In this situation, the knowledge of the human operator would be put in the form of a set of fuzzy linguistic rules. These rules would produce an approximate decision, just as a human would. Consider Figure 7.1 where a block diagram of this definition is shown. As seen, the human operator observes quantities by making observation of the inputs, i.e., reading a meter or measuring a chart, and performs a definite action, e.g., pushes a knob, turns on a switch, closes a gate, or replaces a fuse, i.e., leading to a crisp action, shown here by the output variable $y(t)$. The human operator can be replaced by a combination of a fuzzy rule-based system (FRBS) and a block called defuzzifier (see also Section 1.4.2). The sensory (crisp or numerical) data is fed into FRBS where physical quantities are represented or compressed into linguistic variables with appropriate membership functions. These linguistic variables are then used in the "antecedents" (IF-Part) of a set of fuzzy rules within an inference

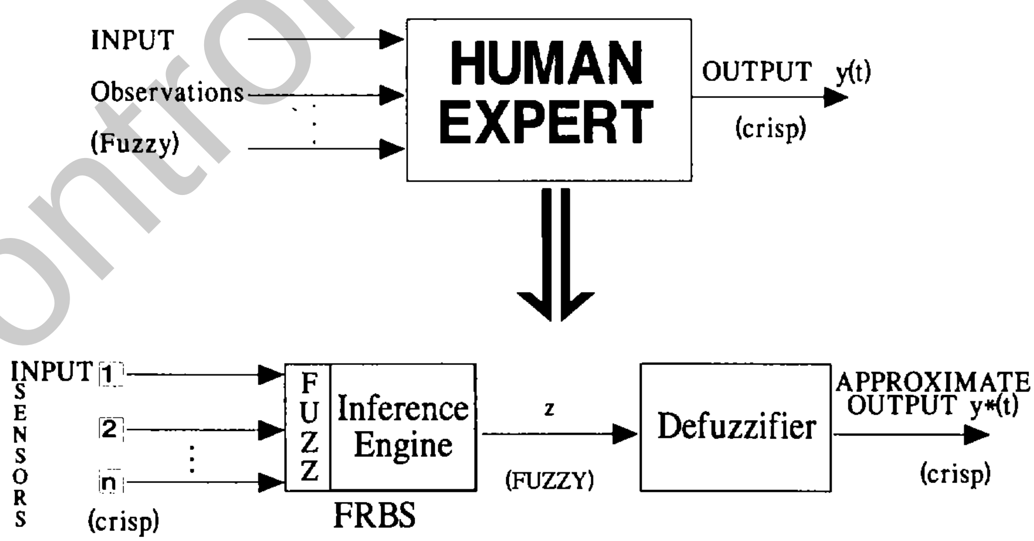


Figure 7.1 A conceptual definition of a fuzzy control system.

controlengineers.ir

controlengineers.ir
@controlengineers

engine to result in a new set of fuzzy linguistic variables or "consequents" (THEN-Part) as variables are then, denoted in this figure by z, combined and changed to a crisp (numerical) output $y^*(t)$ which represents an approximation to actual output $y(t)$.

It is therefore noted that a fuzzy controller consists of three operations: (1) fuzzification, (2) inference engine, and (3) defuzzification. Figure 7.2 shows a pictorial representation of two fuzzy rules of the inference engine firing simultaneously. It is seen that in Rule 1 the two antecedents have an "AND" operator between them, while Rule 2 has an "OR." This would mean that the output of Rule 1 is the minimum of the two antecedent outputs, while in Rule 2, the maximum of the two antecedent outputs are passed on. These two outputs are then combined via a union (or MAX) operation and through a centroid (or weighted average) of the combined distribution, crisp numerical output is yielded.

Before a formal description of fuzzification and defuzzification processes are made, let us consider a typical structure of a fuzzy control system which was first presented in Section 1.4.2 (see Figure 1.7) which is repeated here in Figure 7.3. As shown, the sensory data goes through two levels of interface, i.e., the usual analog to digital and crisp to fuzzy and at the other end in reverse order, i.e. fuzzy to crisp and digital to analog.

Another structure for a fuzzy control system is a fuzzy inference, con-

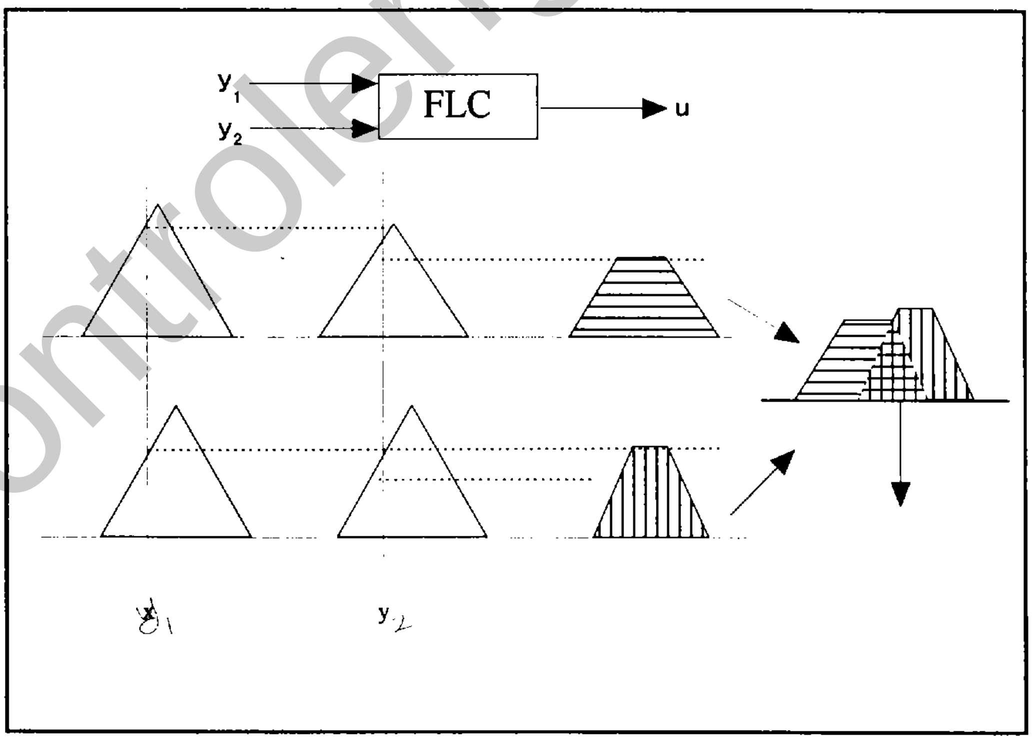


Figure 7.2 A pictorial representation of two fuzzy rules in a fuzzy logic controller.

controlengineers.ir

controlengineers.ir
@controlengineers

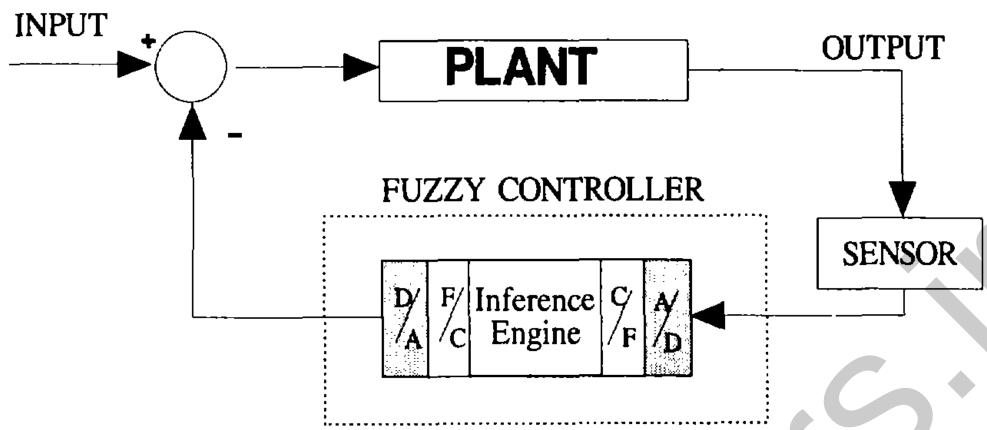


Figure 7.3 Block diagram for a laboratory implementation of a fuzzy controller.

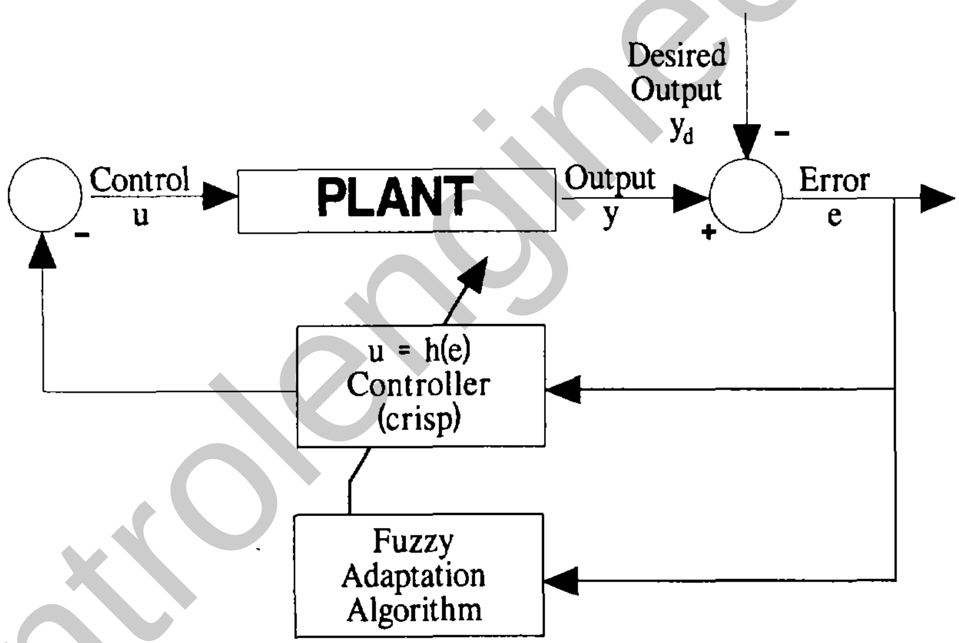


Figure 7.4 An adaptive (tuner) fuzzy control system.

nected to a knowledge base, in a supervisory or adaptive mode. The structure is shown in Figure 7.4. As shown, a classical crisp controller (often an existing one) is left unchanged, but through a fuzzy inference engine or a fuzzy adaptation algorithm the crisp controller is altered to cope with the system's unmodeled dynamics, disturbances, or plant parameter changes—much like a standard adaptive control system. Here the function $h(\cdot)$ represents the unknown nonlinear controller or mapping function $h: e \to u$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

which along with any two input components e_1 and e_2 of e represent a nonlinear surface, sometimes known as the control surface (Jamshidi *et al.*, 1993). More on the control surface is discussed in Appendix A. Adaptive fuzzy control systems will be discussed in Section 8.2.

7.2.2 *Fuzzification*

The fuzzification operation or the fuzzifier unit, as it is sometimes called, represents a mapping from a crisp point $x = (x_1 \;\; x_2 \; \ldots \; x_n)^T \in X$ into a fuzzy set $A \in X$, where X is the universe of discourse (see Appendix A) and T is vector or matrix transposition. There are normally two categories of fuzzifiers in use. They are 1) singleton and 2) nonsingleton. As discussed in Appendix A, a singleton fuzzifier has one point (value) x_p as its fuzzy set support, i.e., the membership function is governed by the following relation:

$$\mu_A(x) = \begin{cases} 1, & x = x_p \in X \\ 0, & x \neq x_p \in X \end{cases} \tag{7.2.1}$$

The nonsingleton fuzzifiers are those in which the support is more than a point. Examples of these fuzzifiers (see Appendix A) are triangular, trapezoidal, Gaussian, left-shoulder, right-shoulder, etc. In these fuzzifiers, $\mu_A(x) = 1$ at $x = x_p$, where x_p may be one or more than one point, and then $\mu_A(x)$ decreases from 1 as x moves away from x_p or the "core" region to which x_p belongs such that $\mu_A(x_p)$ remains 1. For example, the following relation represents a Gaussian-type fuzzifier

$$\mu_A(x) = \exp\left\{-\frac{\left(x - x_p\right)^T \left(x - x_p\right)}{\sigma^2}\right\} \tag{7.2.2}$$

where σ^2 is a parameter characterizing the shape of $\mu_A(x)$ (Wang, 1994a). Today, all fuzzy control software programs (see Appendix B) give the user many choices of fuzzifiers (or shapes of membership functions).

7.2.3 *Inference Engine*

The cornerstone of any expert controller is its inference engine which consists of a set of expert rules which reflect the knowledge base and reasoning structure of the solution of any problem. A fuzzy (expert) control system is no exception and its rule base is the heart of the nonlinear fuzzy

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controller. A typical fuzzy rule can be composed as

$$\text{IF } A \text{ is } A_1 \text{ and } B \text{ is } B_1 \text{ or } C \text{ is } C_1$$
$$\text{THEN } U \text{ is not } U_1 \tag{7.2.3}$$

where A, B, C, and U are fuzzy variables (fuzzy sets or fuzzified crisp variables), A_1, B_1, C_1, and U_1 *are fuzzy linguistic values* (membership functions or fuzzy linguistic labels), "AND," "OR," and "NOT" are *connectives* of the rule. The rule in (7.2.3) has three antecedents and one consequence. Typical fuzzy variables may be "temperature," "pressure," "output," "elevation," etc. and typical fuzzy linguistic values (labels) maybe "hot," "very high," "low," etc. The portion "very" in the label very high is called a *linguistic hedge*. Other examples of a hedge are "much," "slightly," "more or less," etc.

In general in fuzzy system theory, there are many forms and variations of fuzzy rules some of which will be introduced here and throughout this chapter.

Consider the following rule whose consequent is not a fuzzy implication

$$\text{IF } e \text{ is } A \text{ and } \Delta e \text{ is } B$$
$$\text{THEN } u = f(e, \Delta e) \tag{7.2.4}$$

where $f(\cdot)$ is a predefined function of error and change of the error. A special case of (7.2.4) can be written by

$$\text{IF } e_1 \text{ is } A_1,\ e_2 \text{ is } A_2, \ldots, e_n \text{ is } A_n$$
$$\text{THEN } u = a_o + a_1 e_1 + \ldots + a_n e_n \tag{7.2.5}$$

This rule has a linear consequence. It is noted that rules like (7.2.4) and (7.2.5) do not need a defuzzifier to decode any linguistic fuzzy values into numerical values. These rules will be used later within the context of fuzzy control systems. The following example shows how the linguistic type of rule inferencing takes place.

Example 7.2.1 Consider a set of 2 fuzzy rules given by

$$\text{IF } x \text{ is } A_1 \text{ and } y \text{ is } B_1$$
$$\text{THEN } z \text{ is } C_1 \tag{7.2.6}$$

and

$$\text{IF } x \text{ is } A_2 \text{ or } y \text{ is } B_2$$
$$\text{THEN } z \text{ is } C_2$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where the membership functions for fuzzy sets A_i, B_i, an C_i, $i = 1, 2$ are triangular and are described below:

$$\mu_{A_1}(x) = \begin{cases} x & \cdots \quad 0 \le x \le 1 \\ (3-x)/2 & \cdots \quad 1 \le x \le 3 \end{cases} \tag{7.2.7}$$

$$\mu_{A_2}(x) = \begin{cases} (x+2)/2 & \cdots \quad -2 \le x \le 0 \\ (2-x)/2 & \cdots \quad 0 \le x \le 2 \end{cases} \tag{7.2.8}$$

$$\mu_{B_1}(y) = \begin{cases} (y-1) & \cdots \quad 1 \le y \le 2 \\ (4-y)/2 & \cdots \quad 2 \le y \le 4 \end{cases} \tag{7.2.9}$$

$$\mu_{B_2}(y) = \begin{cases} y/2 & \cdots \quad 0 \le y \le 2 \\ (4-y)/2 & \cdots \quad 2 \le y \le 4 \end{cases} \tag{7.2.10}$$

$$\mu_{C_1}(z) = \begin{cases} (z-4)/2 & \cdots \quad 4 \le z \le 6 \\ (9-z)/3 & \cdots \quad 6 \le z \le 9 \end{cases} \tag{7.2.11}$$

$$\mu_{C_2}(z) = \begin{cases} (z-5)/3 & \cdots \quad 5 \le z \le 8 \\ (10-z)/2 & \cdots \quad 8 \le z \le 10 \end{cases} \tag{7.2.12}$$

If the sensory values of x and y are $x_s = 1.5$ and $y_s = 2.5$, respectively, find the output value z from rules (7.2.6) using the "center of gravity," i.e.,

$$z_c = \left(z_1 \cdot \mu_{c_1}(z_1) + z_2 \cdot \mu_{c_2}(z_2)\right) \Big/ \left(\mu_{c_1}(z_1) + \mu_{c_2}(z_2)\right) \tag{7.2.13}$$

where subscript c corresponds to a "crisp" value.

SOLUTION: The membership functions of A_i and B_i, $i = 1, 2$ corresponding to $x_s = 1.5$ and $y_s = 2.5$ are $\mu_{A_1}(x_s) = 0.75$, $\mu_{A_2}(x_s) = 0.25$, $\mu_{B_1}(y_s) = 0.75$, and $\mu_{B_2}(y_s) = 0.75$. Hence, by virtue of the rules given by Equation (7.2.6), we have

Rule 1:
$$\mu_{c_1}(z) = \min\left(\mu_{A_1}(x_s),\ \mu_{B_1}(y_s)\right) = \min(0.75,\ 0.75) = 0.75$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Rule 2: $$\mu_{c_2}(z)=\max\left(\mu_{A_2}(x_s),\ \mu_{B_2}(y_s)\right)$$
$$=\max(0.25,\ 0.75)=0.75$$

Note that because of the connective "or" in Rule 2 we have a maximization in place of a "minimization" which accommodates the "and" connective in Rule 1. Now, corresponding to the value $\mu_{c_i}(z)$, we have two z values (see Figure 7.2). The possible solutions are given by $z_1 =$ 5.5 and 6.75, and $z_2 =$ 7.25 and 8.5. The defuzzified value by center of gravity relation (7.2.13) results in

$$z_c=\frac{((6.125)(0.75)+(7.875)(0.75))}{(0.75+0.75)}=7$$

Standard multiterm controllers such as PI, PD, and PID can easily be represented in terms of fuzzy rules and their associated structures. Figure 7.5 shows the structures for a fuzzy PI, fuzzy PD, and fuzzy PID controllers. Typical fuzzy rules for these controllers are as follows:

(a) Fuzzy PI rule:
IF e is A and Δe is B THEN Δu is D
b) Fuzzy PD rule: (7.2.14)
IF e is A and Δe is B THEN u is D
c) Fuzzy PID rule:
IF e is A and Δe is B and $\Delta^2 e$ is C THEN Δu is D

where A, B, C, and D are fuzzy sets and $\Delta^2 e$ represents an approximate second derivative (double forward difference) of error e.

7.2.4 Defuzzification Methods

Defuzzification is the third important element of a fuzzy controller (see Figures 7.1 and 7.3). In this section, a few of the most common schemes of defuzzification are described.

The above rules' antecedents would remain unchanged, but their consequents will be changed in the form of a polynomial, e.g., for the fuzzy PID controller one has

IF e is A and Δe is B and $\Delta^2 e$ is C
THEN $\Delta u = k_i e + k_p \Delta e + k_d \Delta^2 e$ (7.2.15)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

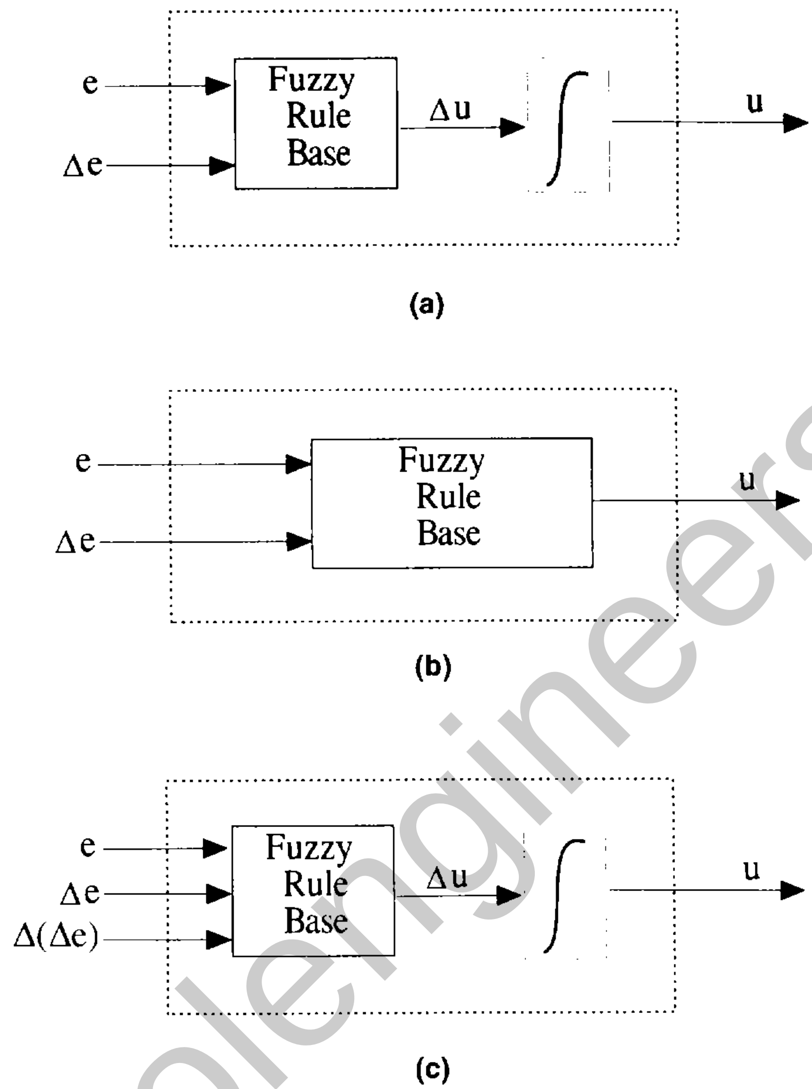


Figure 7.5 Three multiterm fuzzy controllers.

Maximum Defuzzifier Whenever the union of fuzzy outputs of several rules has a unique maximum, the crisp value corresponding to the peak value is taken as the defuzzified value (see Figure 7.6), i.e.,

$$y_c = \arg\left\{\max_{y \varepsilon Y}\left[\mu_A(y)\right]\right\}$$

where $\mu_A(y) = \bigcup_i \left\{\mu_{A_i}(y)\right\}$ is the union of all rules' output fuzzy sets and, once again, subscript c stands for crisp.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

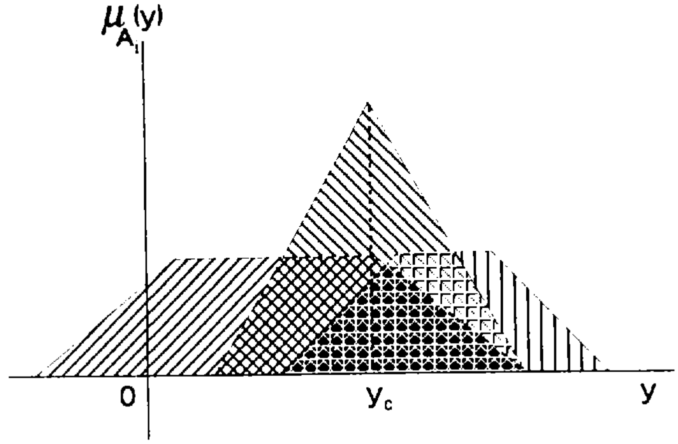


Figure 7.6 Maximum method of defuzzification.

Center of Gravity Defuzzifier In this method (see also Equation (7.2.13) the weighted average of the membership function or the center of gravity of the area bounded by the membership function curve is computed as the most typical crisp value of the union of all output fuzzy sets. Figure 7.7 and the following relation illustrate this method,

$$y_c = \frac{\sum_{i=1}^{n} y_i \mu_{A_i}(y_i)}{\sum_{i=1}^{n} \mu_{A_i}(y_i)} \tag{7.2.16}$$

This scheme is perhaps the most common method and is supported by all software environments.

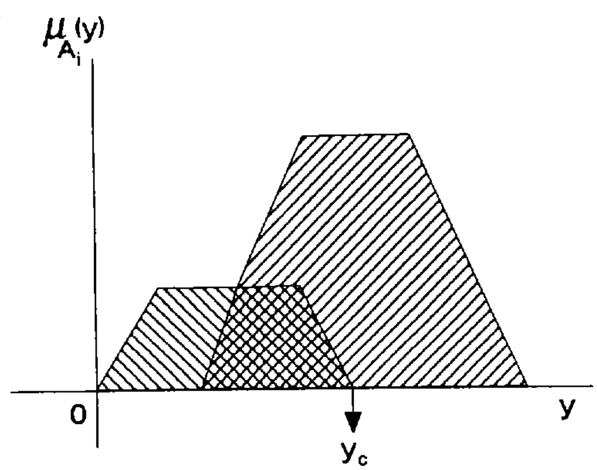


Figure 7.7 Center of gravity (centroid) method of defuzzification.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

There are a few more methods such as the "height" and "center average" or "mean of maximum" methods which are sometimes restricted to membership functions with special properties, etc. The interested reader can refer to Kruse *et al.* (1994) for more information.

The defuzzification of output functional rules like those in Equation (7.2.4) or the ones repeated below:

$$\begin{aligned} &\textit{Rule 1:} \quad \text{IF } e \text{ is } A_1 \text{ and } \Delta e \text{ is } B_1 \text{ THEN} \\ &\qquad\qquad u = f_1(e, \Delta e) \\ &\quad \vdots \\ &\textit{Rule n:} \quad \text{IF } e \text{ is } A_n \text{ and } \Delta e \text{ is } B_n \text{ THEN} \\ &\qquad\qquad u = f_n(e, \Delta e) \end{aligned} \tag{7.2.17a}$$

is given by

$$u_c = \frac{\sum_{i=1}^{n} \mu_u(e_i, \Delta e_i) \cdot f_i(e_i, \Delta e_i)}{\sum_{i=1}^{n} \mu_u(e_i, \Delta e_i)} \tag{7.2.17b}$$

where $\mu_u(e_i, \Delta e_i)$ is the firing strength of the ith rule and n is the total number of rules in the inference engine.

Example 7.2.2 Let us modify the two rules in (7.2.6) of Example 7.2.1 by

$$\text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ THEN } z = \sqrt{x^2 + 2y^2}$$

and

$$\text{IF } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ THEN } z = \sqrt{2x^2 + y^2}$$

If $x_s = 1.5$ and $y_s = 2.5$, find the numerical value of z.

SOLUTION: Using the fuzzy membership functions (7.2.7) through (7.2.10) for A_i and B_i, $i = 1, 2$, the corresponding firing strengths for $\mu(z_i)$, $i = 1, 2$ will be

$$\begin{aligned} \mu(z_1) &= \min\left(\mu_{A_1}(1.5),\ \mu_{B_1}(2.5)\right) \\ &= \min(0.75,\ 0.75) = 0.75 \end{aligned}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\mu(z_2) = \min\left(\mu_{A_2}(1.5),\ \mu_{B_2}(2.5)\right)$$
$$= \min(0.25,\ 0.75) = 0.25$$

Then by virtue of the above rules we have two values for z, i.e., $z_1 = 3.84$ and $z_2 = 3.28$. To infer the numerical value of z from both rules, the defuzzification relation (7.2.17b) can be used to find $z = ((0.75)(3.84) + (0.25)(3.28) / (0.75 + 0.25) = 3.7$.

7.2.5 The Inverted Pendulum Problem

In this section, the much discussed inverted pendulum problem will be first introduced and then two fuzzy control structures will be introduced for it. A third structure is given in Section 8.2.4. The first structure is based on the linguistic-type rules (see Equation (7.2.14), sometimes called the *Mamdani-type* controller (Mamdani and Assilian, 1975). The second structure is one in which the consequence of the rules is replaced by a dynamic relation such as a difference equation, or those in Equations (7.2.4) or (7.2.15) or simply a function. This structure is sometimes referred to as *Takagi-Sugeno-type* controller (Jamshidi *et al.*, 1993).

Consider an inverted pendulum or the cart pole system shown in Figure 7.8. This system is commonly treated as either a 4-state $(\theta,\ \dot{\theta},\ x, \text{ and } \dot{x})$ or

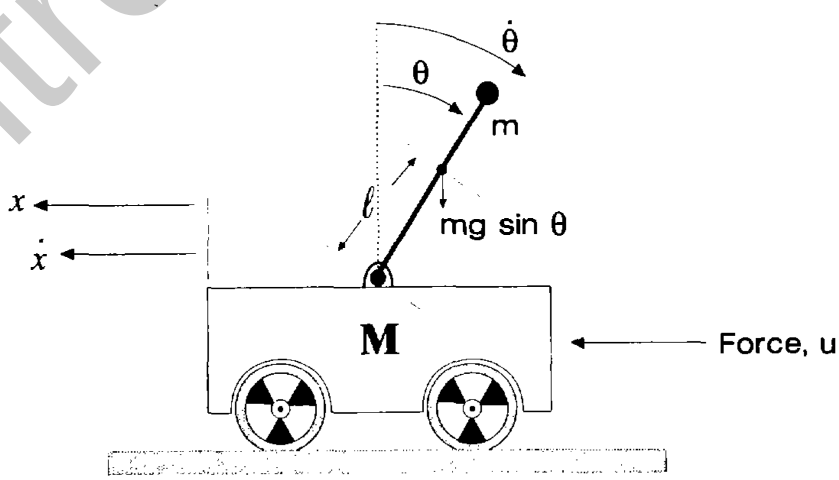


Figure 7.8 A schematic for an inverted pendulum.

controlengineers.ir

controlengineers.ir
@controlengineers

a 2-state (θ and $\dot{\theta}$) space model. Below are both dynamic models of the inverted pendulum (Jamshidi *et al.*, 1992; Wang, 1994a).

Inverted Pendulum with 4 states:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{-\frac{7}{3}m^2\ell^3 x_4^2 \sin x_3 + (m\ell)^2 g \sin x_3 \cos x_3 - \frac{7}{3}m\ell^2 u}{\left[m\ell \cos x_3\right]^2 - \left[\frac{7}{3}m\ell^2(M+m)\right]}$$

(7.2.18)

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{(m\ell)^2 x_4^2 \sin x_3 \cos x_3 - mg\ell \sin x_3 (M+m) + m\ell \cos x_3}{\left[m\ell \cos x_3\right]^2 - \left[\frac{7}{3}m\ell^2(M+m)\right]} u$$

where $\mathrm{x}^T = (\theta, \dot{\theta}, x, \dot{x})$, m and M are the masses of the pole and cart, respectively, usually assumed that $m << M$, ℓ is one-half of the length of the pole, and $g = 9.8$ m/sec^2 is the gravitational acceleration constant, and $u = F$ is the force acting on the cart.

Inverted pendulum with 2 states:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{g \sin x_1 - \dfrac{m\ell x_2^2 \cos x_1 \sin x_1}{m+M}}{\ell\left(\dfrac{4}{3} - \dfrac{m\cos^2 x_1}{m+M}\right)} + \frac{\dfrac{\cos x_1}{m+M}}{\ell\left(\dfrac{4}{3} - \dfrac{m\cos^2 x_1}{m+M}\right)} u \qquad (7.2.19)$$

where the state vector $\mathrm{x}^T = (\theta, \dot{\theta})$ and $u = F$. For the first numerical analysis of the system, the following values are used: $\ell = 0.5$ m, $m = 100$ grams, and $M = 1$ kgram (Wang, 1994a). Using these values, the two models will

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

become

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{-0.003x_4^2 \sin x_3 + 0.025 \sin x_3 \cos x_3 - 0.06u}{0.0025 \cos^2 x_3 - 0.064}
\end{aligned} \tag{7.2.20}
$$

$$
\begin{aligned}
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{0.0025x_4^2 \sin x_3 - 0.49 \sin x_3 + 0.05 \cos x_3}{0.0025 \cos^2 x_3 - 0.064} u
\end{aligned}
$$

and for the 2-dimensional pendulum we have

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= 1.58 \sin x_1 - 0.07 x_2 \sin x_1 \cos x_1 + 1.5u
\end{aligned}
$$

Since most classical control methods require a linear model, we can assume a set of nominal conditions $x_n = (0\ \ 0\ \ 0\ \ 0)^T$ or $x_n = (0\ \ 0)^T$, depending on which model we choose. Then, the linearized models will be given by

$$
\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 7.97 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0.97 \\ 0 \\ -0.8 \end{pmatrix} u \tag{7.2.21}
$$

or

$$
\dot{x} = \begin{pmatrix} 0 & 1 \\ 15.79 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1.46 \end{pmatrix} u \tag{7.2.22}
$$

As one expects, these linearized models represent two unstable systems.

Example 7.2.3 Consider the 4th-order model of the inverted pendulum system of Equation (7.2.21). It is desired to stabilize it such that the new closed-loop poles are at -1, -2, -3, and -4.

SOLUTION: Since the states of the inverted pendulum represent position and velocity of both angular and linear positions, then a state feedback law

$$
u = -kx = -k_1x_1 - k_2x_2 - k_3x_3 - k_4x_4
$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is effectively a PD controller and would suffice to stabilize the system provided that the system is controllable. A quick check on controllability shows that the system is, in fact, controllable. The application of any pole placement program would give a gain vector for k. This system's state feedback law will be used with its nonlinear model in Section 8.2 to design an adaptive fuzzy tuner controller.

One of the first steps in the design of any fuzzy controller is to develop a knowledge base for the system to eventually lead to an initial set of rules. There are at least five different methods to generate a fuzzy rule base:

i) simulate the closed-loop system through its mathematical model,
ii) interview an operator who has had many years of experience controlling the system,
iii) generate rules through an algorithm using numerical input/output data of the system,
iv) use learning or optimization methods such as neural networks (NN) or genetic algorithms (GA) to create the rules, and
v) in the absence of all of the above, if a system does exist, experiment with it in the laboratory or factory setting and gradually gain enough experience to create the initial set of rules.

In sequel, the second-order model of the inverted pendulum will be used to help the designer obtain a knowledge base leading to a set of linguistic rules.

Example 7.2.4 For the linearized model of the inverted pendulum described by (7.2.22), develop a knowledge base through simulation.

SOLUTION: Since the system (7.2.22) is unstable but controllable, we can design either a PD or a PID controller to first stabilize it, and then through variations of the system's desired poles, study it under different circumstances. We offer both approaches here.

PD Controller The application of a PD controller $u = \theta_d - k_p x_1 - k_d x_2$ to (7.2.22) leads to a closed-loop system

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ a - bk_p & -bk_d \end{pmatrix} x + \begin{pmatrix} 0 \\ b \end{pmatrix} \theta_d \tag{7.2.23}$$

where $a = 15.79$, $b = 1.46$, and the desired angular position is $\theta_d(t) = 1 - e^{-2t}$. Assuming that the desired closed-loop poles are at λ_1 and λ_2, a desired

controlengineers.ir
@controlengineers

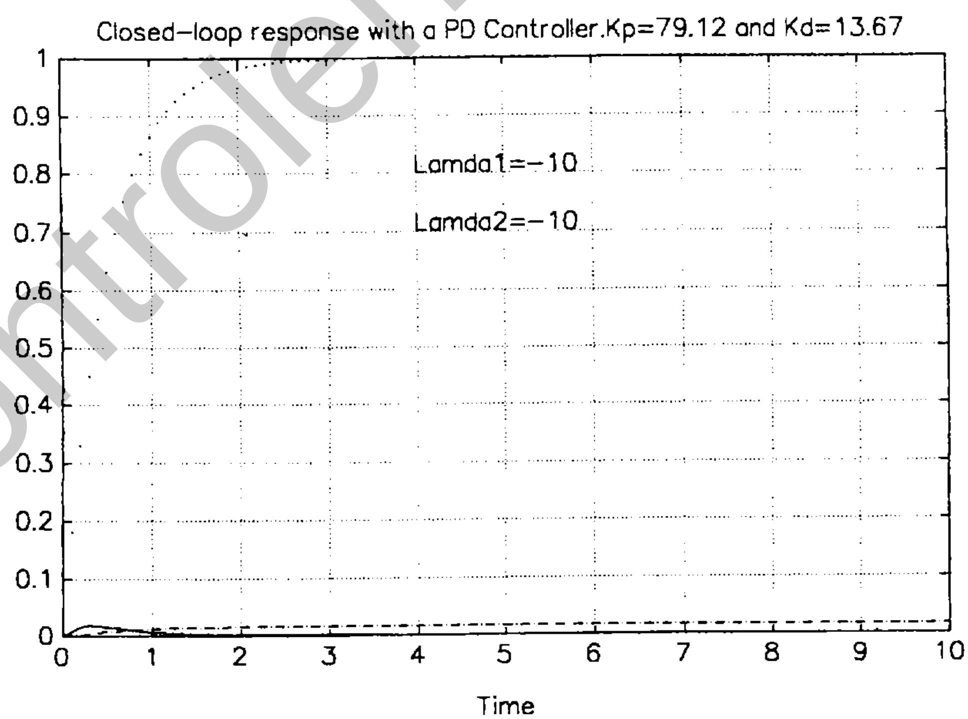
characteristic polynomial $\Delta(\lambda) = \lambda^2 - (\lambda_1 + \lambda_2)\,\lambda + \lambda_1\lambda_2$ is obtained. Equating the coefficients of this polynomial to the system's $\Delta(\lambda) = \det(\lambda I - A_c)$, where A_c is the closed-loop system matrix of (7.2.23), leads to:

$$k_p = \frac{\lambda_1\lambda_2 + a}{b}, \quad k_d = -\frac{(\lambda_1 + \lambda_2)}{b}$$

Using standard software environments such as Matlab (Jamshidi *et al.*, 1992), several simulation rules for different values of λ_1 and λ_2 were deduced. Figure 7.9 shows the simulation results.

As shown in this figure, the best results are obtained for low values of $\lambda_1 = \lambda_2 = -1.2$ or $k_p = 11.78$ and $k_d = 1.64$ with a 1% error. As k_p and k_d increase to very large values of $k_p = 79.12$ and $k_d = 13.67$, the error jumps up to 99%. This study gives the designer an insight into rules for the variations of k_p and k_d as a function of the system's responses. For example, to avoid large tracking errors, one can compose a rule such as:

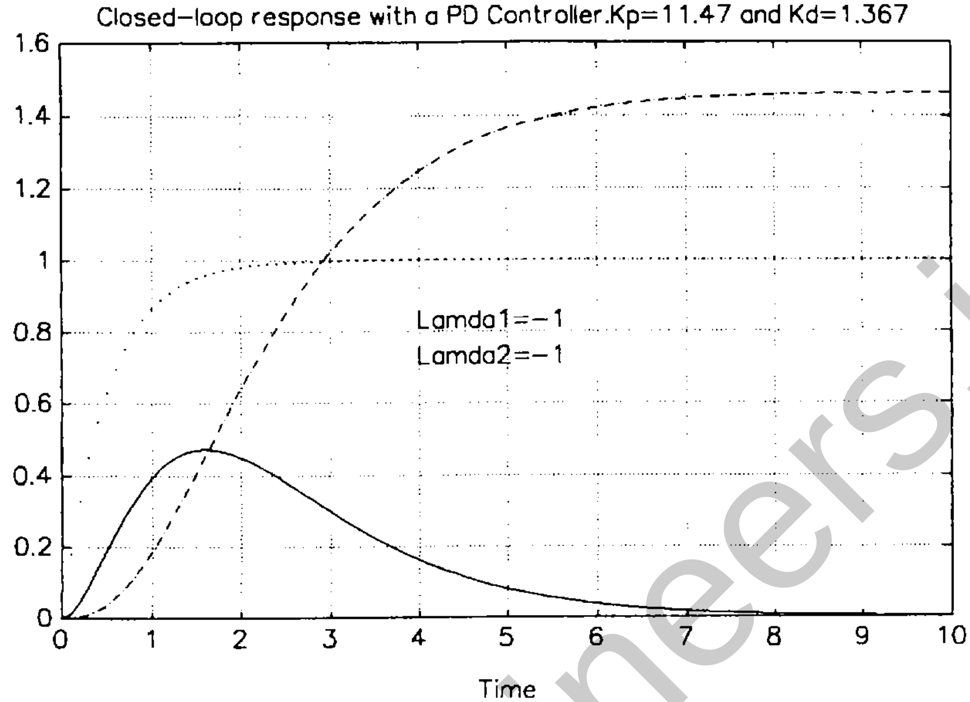
IF θ is small and $\dot{\theta}$ is very small, THEN k_p is medium and k_d is small.



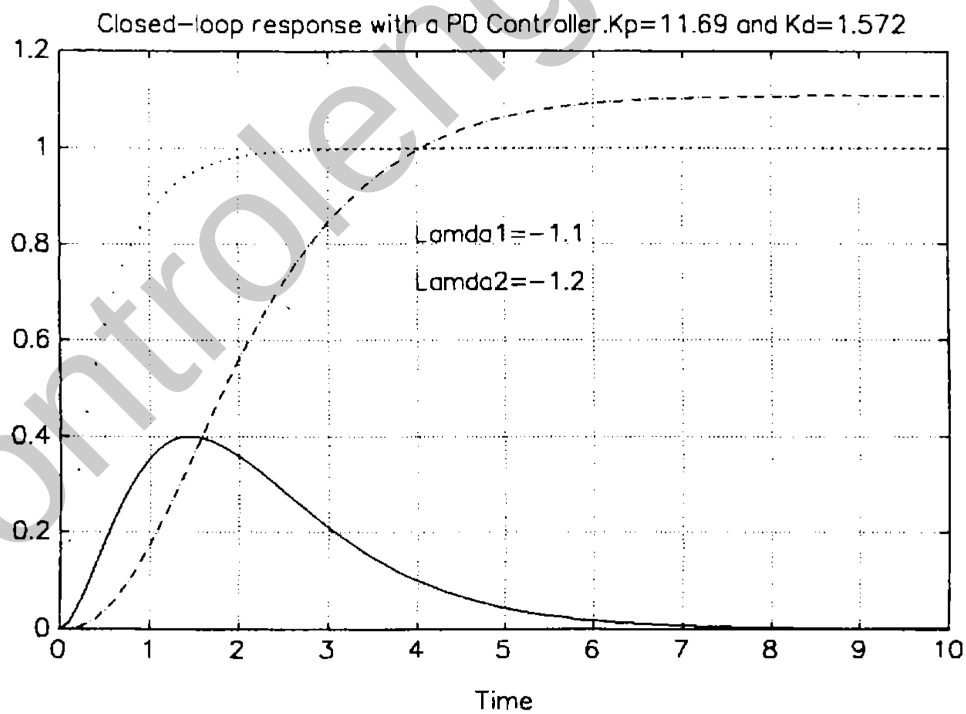
(a) Simulation Run No. 1.

controlengineers.ir

controlengineers.ir
@controlengineers



(b) Simulation Run No. 2.



(c) Simulation Run No. 3.

Figure 7.9 (opposite page and above) Simulation results for PD control of the inverted pendulum; ------ represents θ (position) and —— represents $\dot{\theta}$ (velocity).

controlengineers.ir @controlengineers

Such a rule would help avoid the drastic 99% error situation of Figure 7.9a. The results of examples like this will be used for adaptive fuzzy tuning of the standard crisp PD or PID controllers in Section 8.2.

PID Controller Here the control law is given by

$$\begin{aligned} u &= -k_p x_1 - k_i \int (x_1 - \theta_d)dt - k_d x_2 \\ &= (-k_p - k_d - k_i)x \end{aligned} \tag{7.2.24}$$

where $x = (x_1 \;\; x_2 \;\; x_3)^T$, $x_3 \triangleq \int (x_1 - \theta_d)dt$ is a third (pseudo) state variable as a result of the integral action of the controller, and $y = \theta$. Using the controller (7.2.24) for the system of Equation (7.2.22), one gets

$$\begin{aligned} \dot{x} &= A_c x + B\theta_d \\ &= \begin{pmatrix} 0 & 1 & 0 \\ a - bk_p & -bk_d & -bk_i \\ 1 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ b \\ -1 \end{pmatrix} \theta_d \end{aligned}$$

where, once again, $a = 15.79$ and $b = 1.46$. Assuming that the new poles of the system are located at λ_1, λ_2 and λ_3, equating the coefficients of the desired and system's characteristic polynomial, i.e., $\Delta_d(\lambda) = \Delta(\lambda) = \det(\lambda I - A_c)$, would, after some arithmetic operations, result in

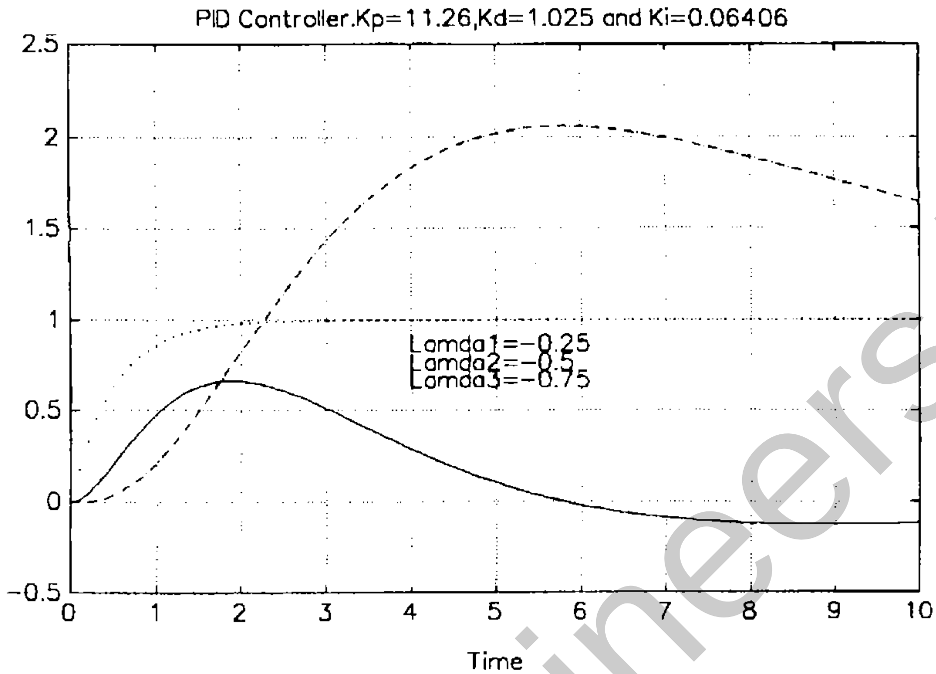
$$\begin{aligned} k_p &= (a + \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3)/b \\ k_i &= -(\lambda_1\lambda_2\lambda_3)/b \\ k_d &= -(\lambda_1 + \lambda_2 + \lambda_3)/b \end{aligned}$$

The simulation results are shown in Figure 7.10. Once again, four different sets of responses are shown here. Typical rules from such a study can be deduced as

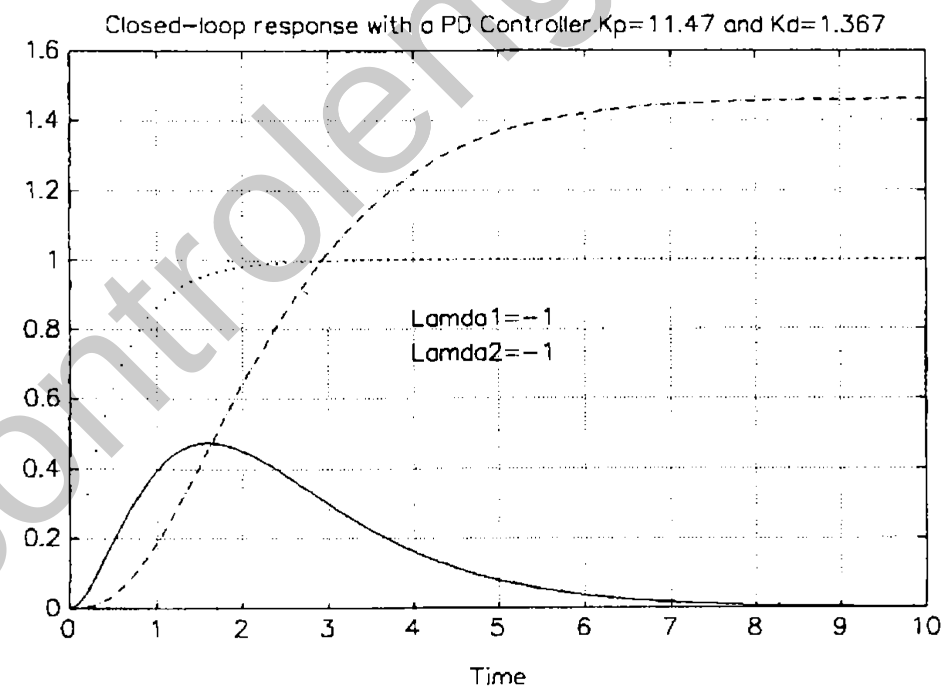
> IF error is small and change in error is zero,
> THEN k_p is high, k_i is very high and k_d is medium.

This rule would characterize the response shown in Figure 7.10c. Note that the case in Figure 7.10d is essentially a proportional controller which

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) Simulation Run No. 1.

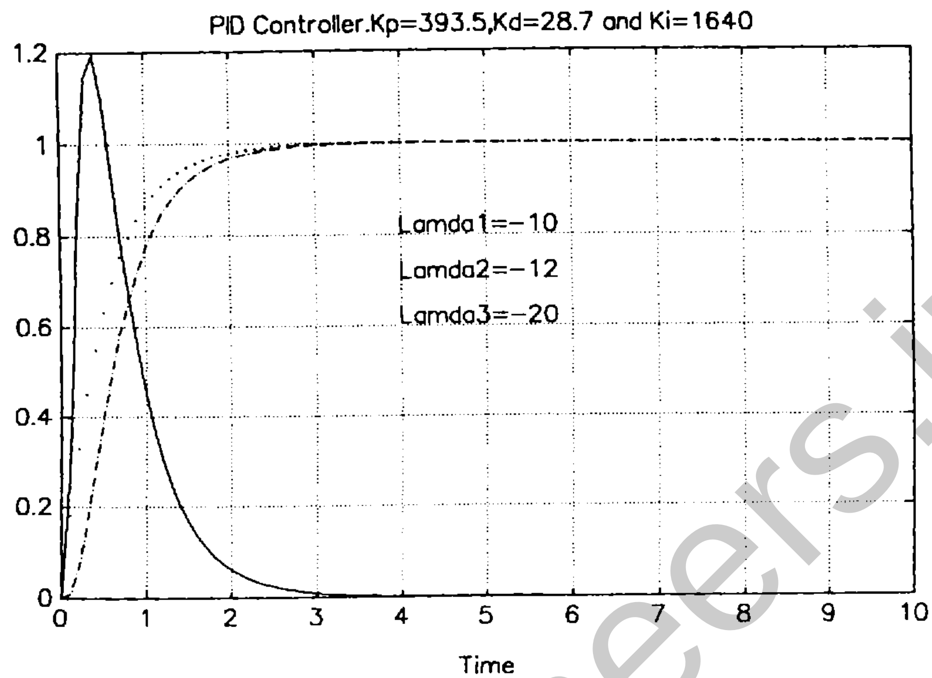


(b) Simulation Run No. 2.

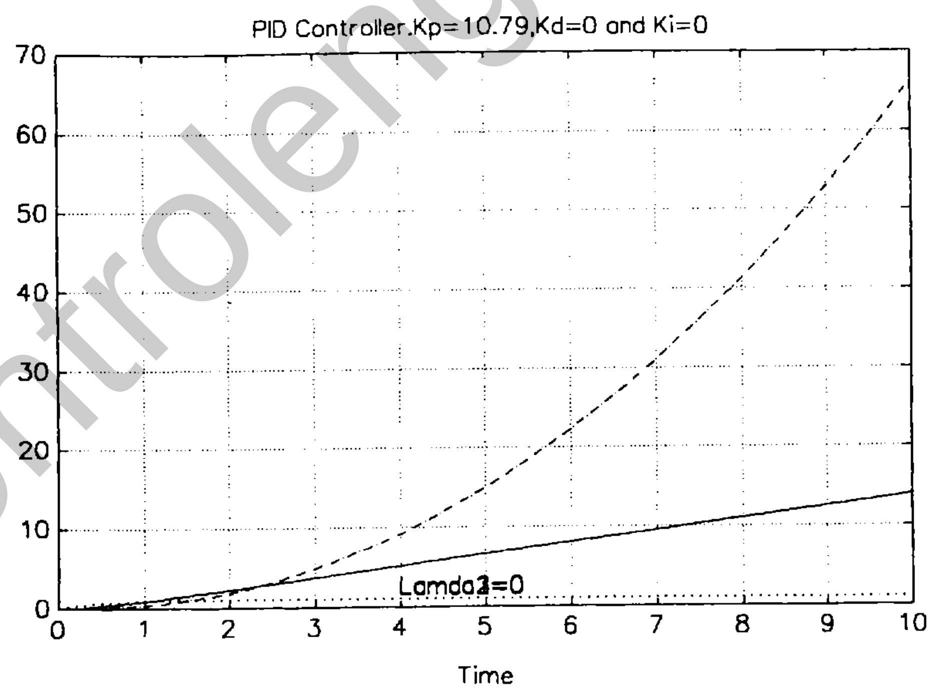
Figure 7.10a & b Simulation results for PID control of the inverted pendulum; ------- represents θ (position) and —— represents $\dot{\theta}$ (velocity).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(c) Simulation Results No. 3



(d) Simulation Results No. 4

Figure 7.10c & d Simulation results for PID control of the inverted pendulum; -·-·-·- represents θ (position) and —— represents $\dot{\theta}$ (velocity).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

we knew was not suitable for the unstable open-loop inverted pendulum problem.

Example 7.2.5 Now, let us reconsider the second-order inverted pendulum problem and simulate it using the FULDEK program (see Appendix B) with a sequence of rules, i.e., 7, 5, 3, and 2 rules. The last case is essentially a fuzzy version of a bang-bang controller.

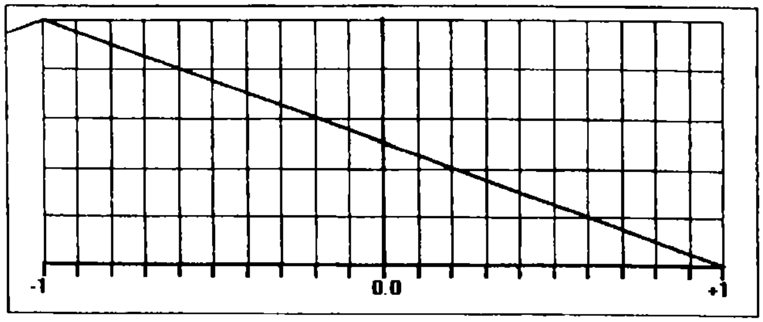
SOLUTION: In order to simulate the inverted pendulum using FULDEK, one needs to create a ".dat" file which would define the linear model of the pendulum (see Appendix B). This file is shown below:

```
1
2, 1, 1, 1,
0.0 , 1.0
15.79171   ,   0.0
0
1.46341
0.
0
1.0 , 0.0
0.0
0.0
0.5
0.0
```

The membership functions for the pendulum's states (position and velocity) and control (torque) variables are shown in Figure 7.11. Three triangular and three singletons have been chosen. As required, four different fuzzy inference engines were composed. These inference engines are shown in Table 7.1. The terms in the first two columns are the antecedents of the fuzzy IF-THEN rules connected by an "and," while the third column is the consequence of the rules. The four fuzzy controllers were separately simulated on FULDEK. Figure 7.12 shows four sets of simulation results for the inverted pendulum. As seen, all simulations have resulted in stable responses for both states. All have reasonable responses even for the two-rule case, i.e., fuzzy bang-bang control. With a quick look at the settling times of both θ and $\dot{\theta}$ and percent overshoot of $\dot{\theta}$, it becomes evident that although the 2-rule case has the largest overshoot, it has the shortest settling time. This simulation exercise indicates that too few rules are not necessarily unacceptable and that too many rules may not always result in or guarantee good responses. In fact, in Section 8.3, the reduction of the size of the rule set will be a major topic under the heading of hierarchical fuzzy control systems. Figure 7.13 shows the control surfaces of two of the four cases. The five- and seven-rule surfaces were very much similar.

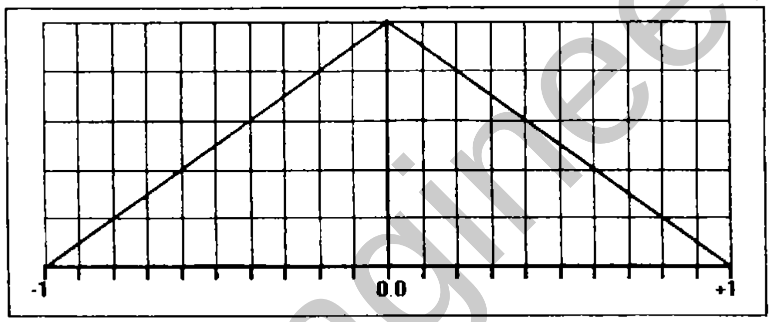
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



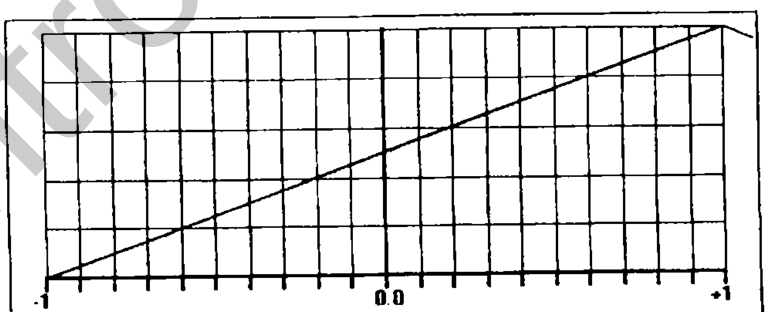
TTH_NEG1

Index	Triang X	Triang Y
1	-3.00000	0.00000
2	-1.00000	1.00000
3	1.00000	0.00000



TTH_NULL

Index	Triang X	Triang Y
1	-1.00000	0.00000
2	0.00000	1.00000
3	1.00000	0.00000



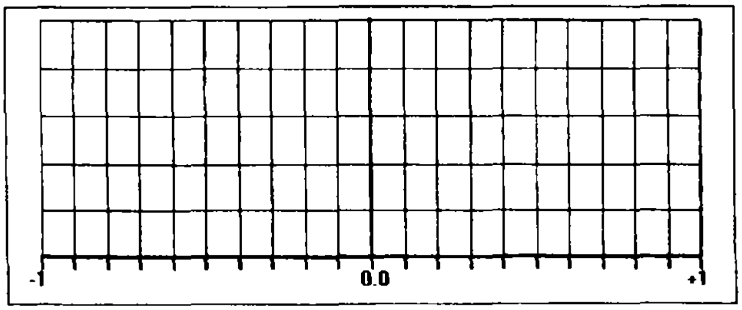
TTH_POS1

Index	Triang X	Triang Y
1	-1.00000	0.00000
2	1.00000	1.00000
3	3.00000	0.00000

Figure 7.11(a) Membership functions for Example 7.2.5, velocity and position.

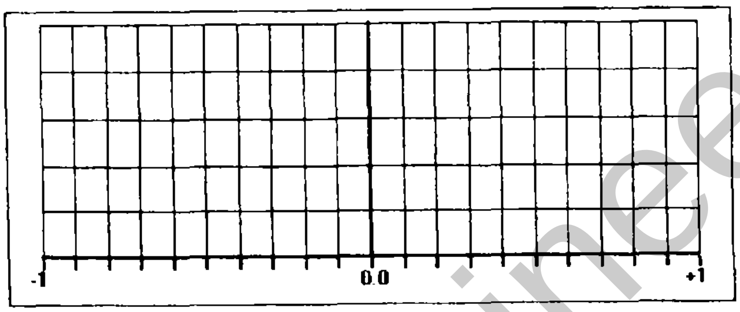
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



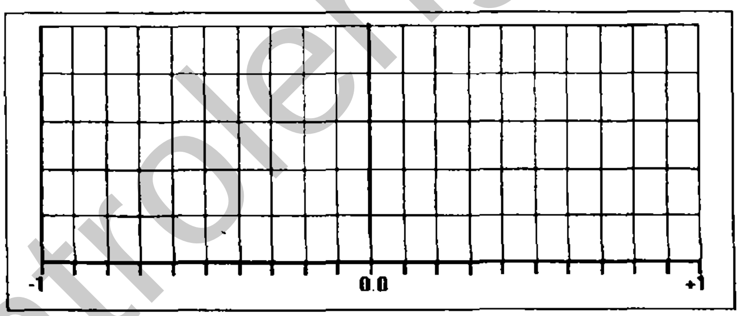
S_POS1

Index	Singl X	Singl Y
1	1.00000	1.00000



S_NULL

Index	Singl X	Singl Y
1	0.00000	1.00000



S_NEG1

Index	Singl X	Singl Y
1	-1.00000	1.00000

Figure 7.11(b) Membership functions for Example 7.2.5, torque.

As discussed, the inverted pendulum has been one the most popular testbeds for both classical and fuzzy control system design. One experimental setup of real-time fuzzy control with 11 rule chips and 1 defuzzifier chip of note is the one developed by Yamakawa (1994), whose setup is shown in Figure 7.14. In Figure 7.14a. A wine glass is being stabilized on

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 7.1 Four Fuzzy Inferences for Inverted Pendulum Problem

1st case :

Theta	Theta Dot	Torque
TTH_NEG1	TTH_NEG1	S_POS1
TTH_POS1	TTH_POS1	S_NEG1

2nd case :

Theta	Theta Dot	Torque
TTH_NEG1	TTH_NEG1	S_POS1
TTH_POS1	TTH_POS1	S_NEG1
TTH_NULL	TTH_NULL	S_NULL

3th case :

Theta	Theta Dot	Torque
TTH_NEG1	TTH_NEG1	S_POS1
TTH_POS1	TTH_POS1	S_NEG1
TTH_NULL	TTH_NULL	S_NULL
TTH_NEG1	TTH_POS1	S_NULL
TTH_POS1	TTH_NEG1	S_NULL

4th case :

Theta	Theta Dot	Torque
TTH_NEG1	TTH_NEG1	S_POS1
TTH_POS1	TTH_POS1	S_NEG1
TTH_NULL	TTH_NULL	S_NULL
TTH_NEG1	TTH_NULL	S_POS1
TTH_NULL	TTH_NEG1	S_POS1
TTH_POS1	TTH_NULL	S_NEG1
TTH_NULL	TTH_POS1	S_NEG1

controlengineers.ir

top of the pendulum with the same fuzzy controller which stabilizes the pendulum, without the wine glass and the wine itself. Robustness of the fuzzy controller is further demonstrated here in Figure 7.14a by placing a free-walking chick on the flat platform situated on top of the pendulum. More details on this experiment and corresponding fuzzy controller architecture are given in Section 8.3.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

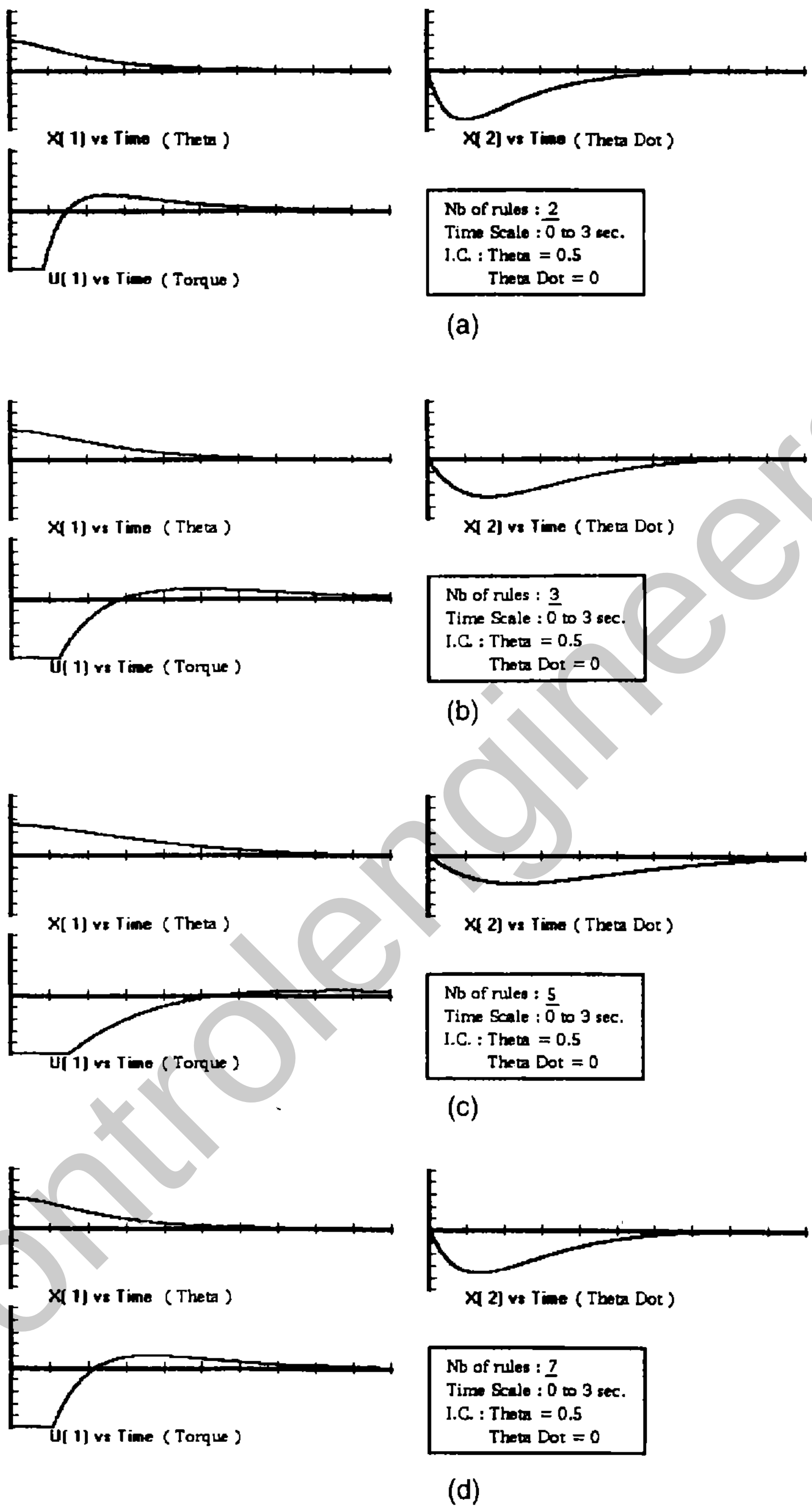
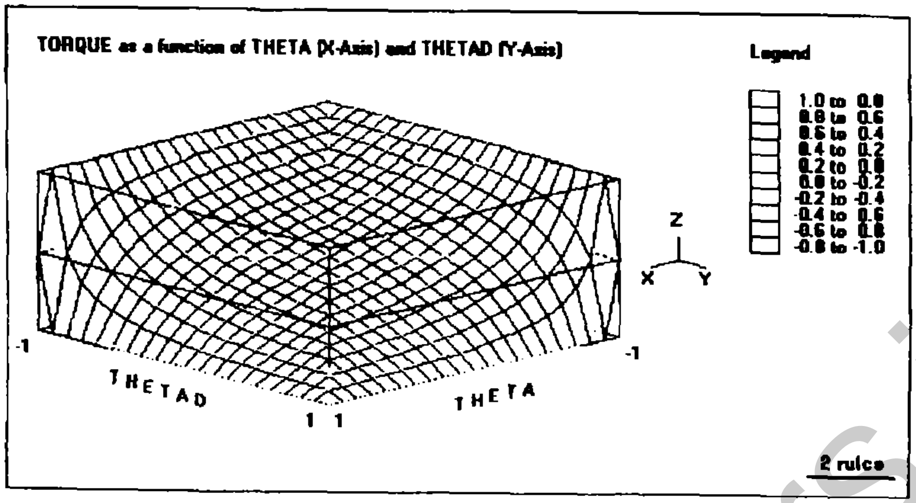


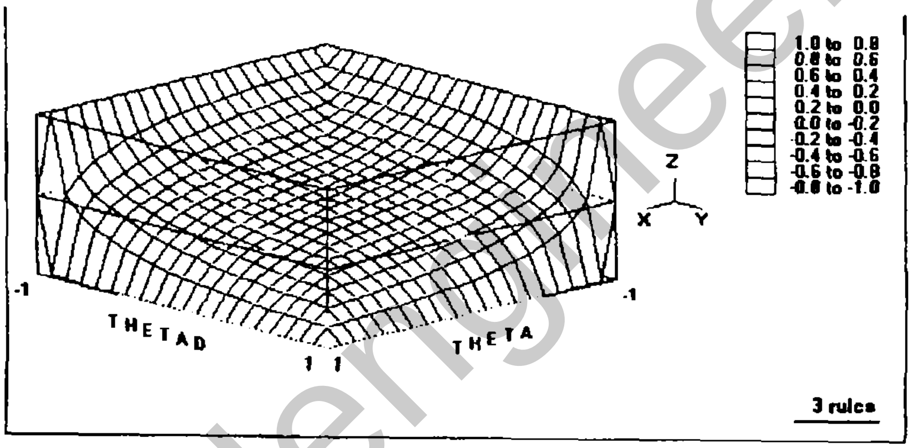
Figure 7.12 Four simulations of the inverted Pendulum: (a) two rules, (b) three rules, (c) five rules, and (d) seven rules.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a)



(b)

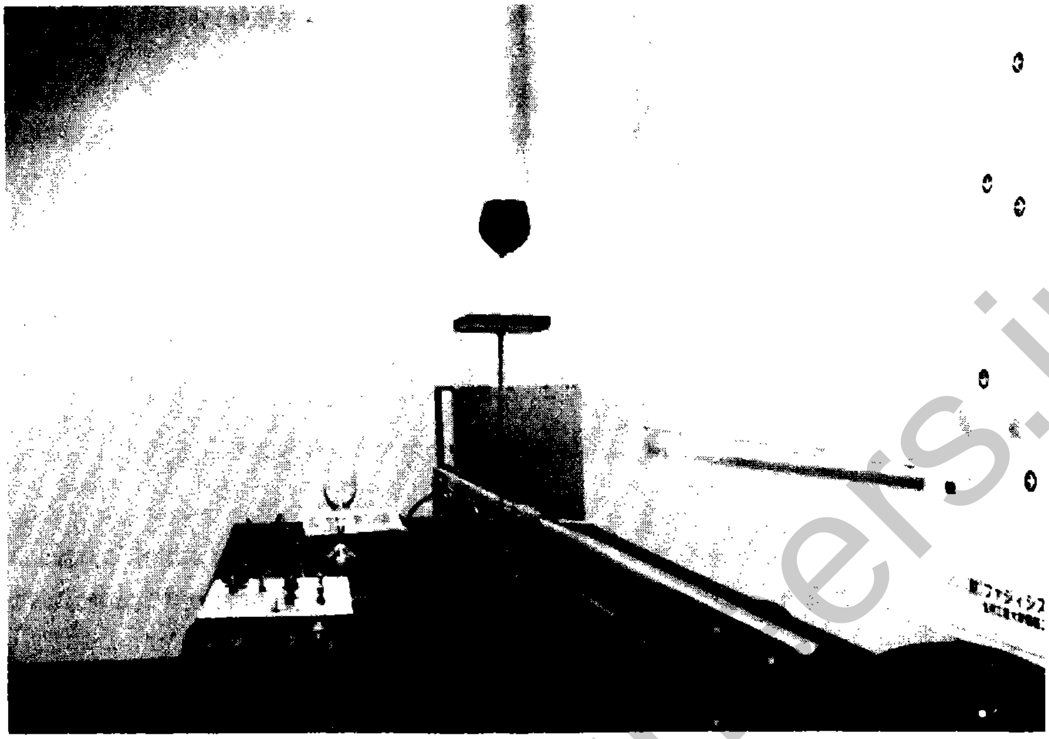
Figure 7.13 Control surfaces for two of the four simulation cases of the inverted pendulum: (a) two rules and (b) three rules.

7.2.6 *Overshoot—Suppressing Fuzzy Controllers*

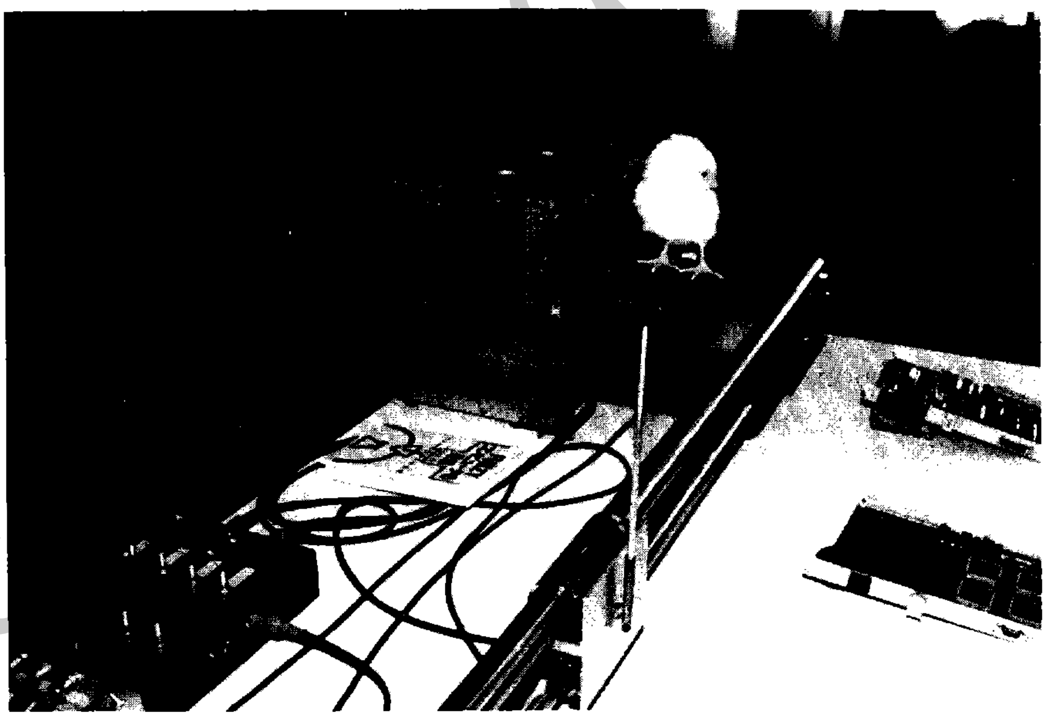
It is well known that in classical control systems the speed of the response and the amount of overshoot are in conflict with each other. On one hand, a system requiring fast rise time may undergo large overshoots. On the other hand, requiring minimal overshoot would result in a fairly slow response. In order to have both fast response and little or no overshoot, a new strategy has to be devised.

Fuzzy logic inferencing can be used to achieve minimal overshoot without sacrificing the speed of the system response. A controller with this characteristic was first introduced by the Japanese corporation Yokogawa

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) Wine.



(b) A chick.

Figure 7.14 The wine glass balancing experiment on the inverted pendulum.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

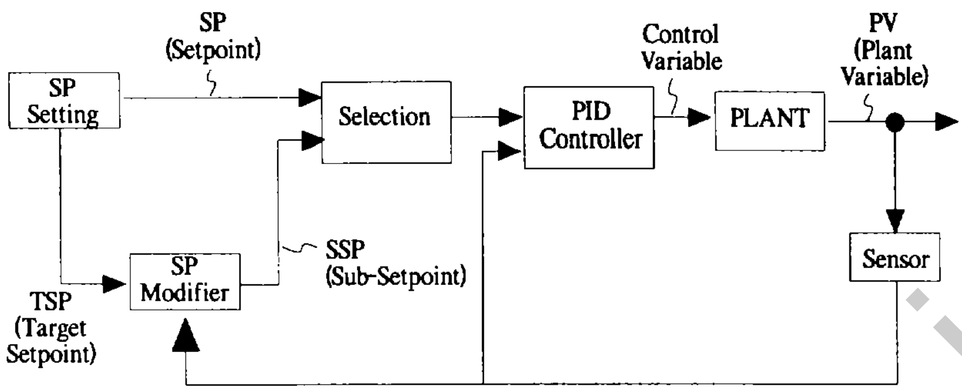


Figure 7.15 Block diagram of an overshoot suppressor controller.

in 1991. Figure 7.15 shows a block diagram of such a control system. Under normal situations, the set point of the system has a fixed numerical value. However, here the setpoint would be changed in accordance with a fuzzy inference engine which is incorporated within the block, called Selection in Figure 7.15. A pictorial characteristic of the overshoot suppression algorithm is shown in Figure 7.16. The sub-setpoint (SSP) which is the output of the block designated as SP Modifier in Figure 7.15, is computed in the background as the plant undergoes initial operation. At the instant when the plant variable (PV) and sub-setpoint value are equal, the fuzzy inference will begin (see Figure 7.16). Yokogawa Corporation engineers have composed 40 fuzzy rules to take care of every possible way of sup-

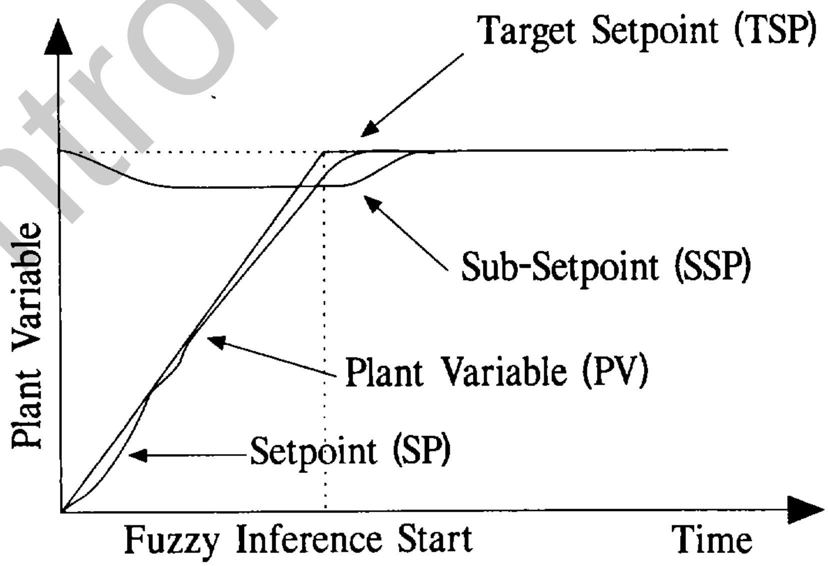


Figure 7.16 Overshoot suppression algorithm principle.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

pressing the overshoot. The result is that beyond this point the fuzzy inference will determine a new setpoint for the system which would force the plant's response not to exceed the target setpoint (TSP).

Example 7.2.6 Consider a unity feedback SISO system described by the transfer function $G(s) = 1/[(1 + 5s)^2 (1 + 0.5s)]$ whose closed-loop transfer function is $T(s) = 0.08/(s^3 + 2.4s^2 + 0.84s + 0.16)$ whose state model is given by

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.16 & -0.84 & -2.4 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0.08 \end{pmatrix} u$$

$$y = (1 \quad 0 \quad 0)x$$

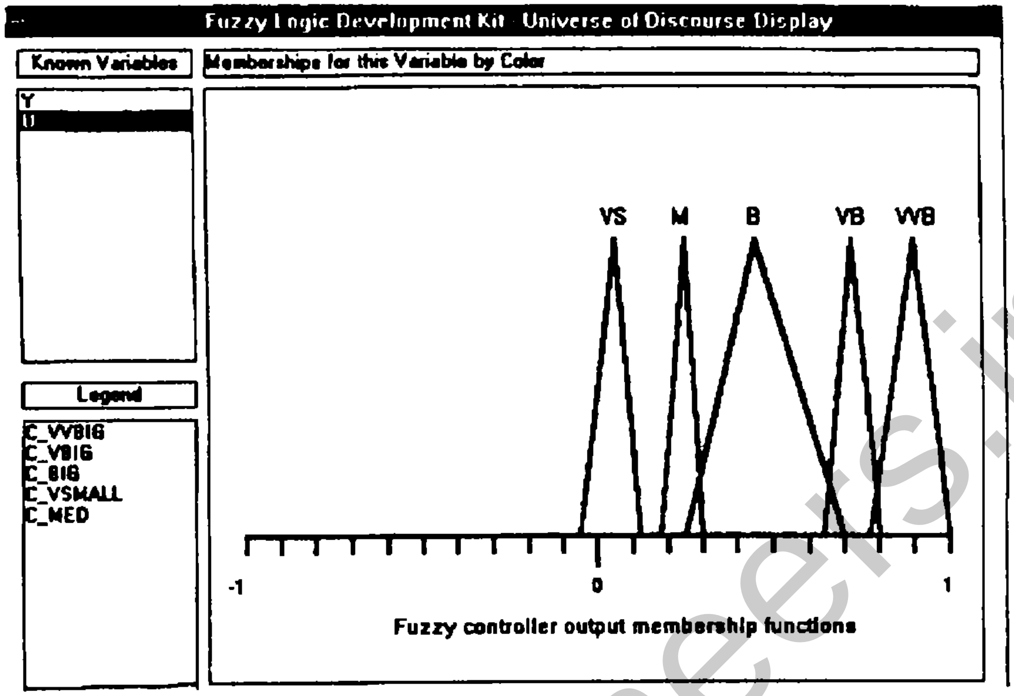
It is desired to obtain a minimal overshoot controller for this system.

SOLUTION: The system was simulated on FULDEK (see Appendix B) with fuzzy rules suppressing overshoot. The controller was a simple fuzzy output feedback whose membership functions are shown in Figure 7.17. Note that the output variable does not have any negative labels which would correspond to an all-positive response of the plant. The eight fuzzy rules devised for this controller are listed below:

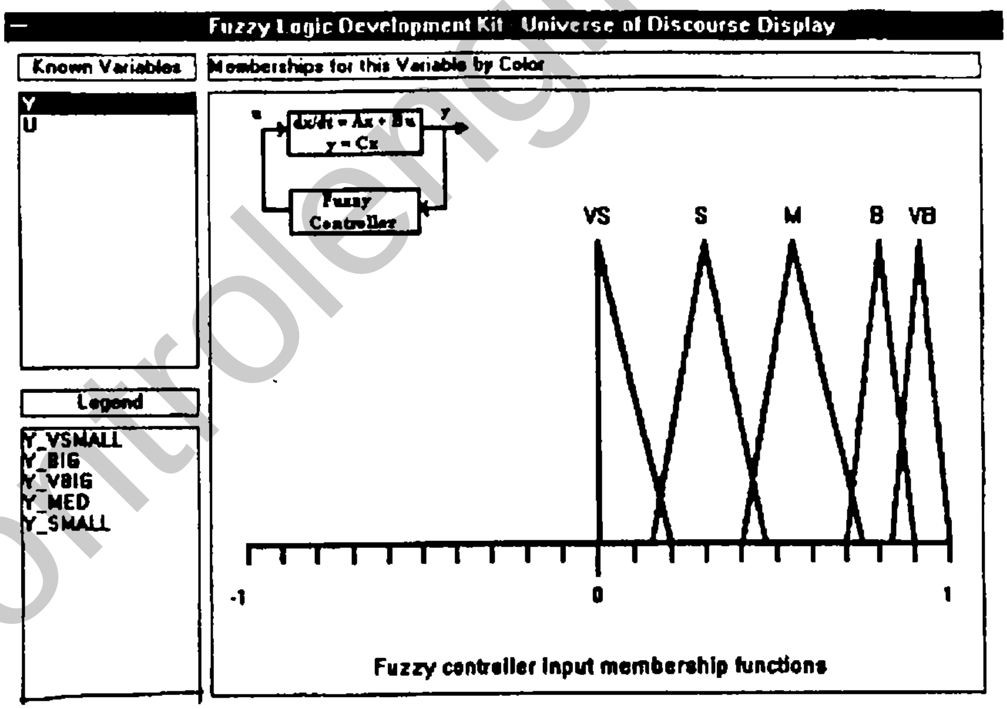
1. IF Y is Y_VSMALL THEN U is C_VVBIG
2. IF Y is Y_BIG THEN U is C_VBIG
3. IF Y is Y_VBIG AND Y is Y_BIG, THEN U is C_VBIG
4. IF Y is Y_VBIG THEN U is C_BIG
5. IF Y is Y_BIG OR Y is Y_MED THEN U is C_VSMALL
6. IF Y is Y_MED THEN U is C_MED
7. IF Y is Y_BIG AND Y is Y_MED THEN U is C_VSMALL
8. IF Y is Y_SMALL THEN U is C_VBIG

The step response of the system for a conventional output feedback controller is shown in Figure 7.18. It is noted here that the system has a relatively small overshoot of 6.25% with a slow rise time of $t_r = 6.5$ seconds, a peak time of $t_p = 16$ seconds, and a long settling time of $t_s \cong 30$ seconds. Clearly, the system is not very desirable. Figure 7.19 shows the control and output time responses under the fuzzy logic-based overshoot suppressing controller. As seen, the rise time and settling time have been

controlengineers.ir
@controlengineers



(a) input variable labels



(b) output variable labels.

Figure 7.17 Membership functions (labels) for input and output variables of Example 7.2.6.

controlengineers.ir
@controlengineers

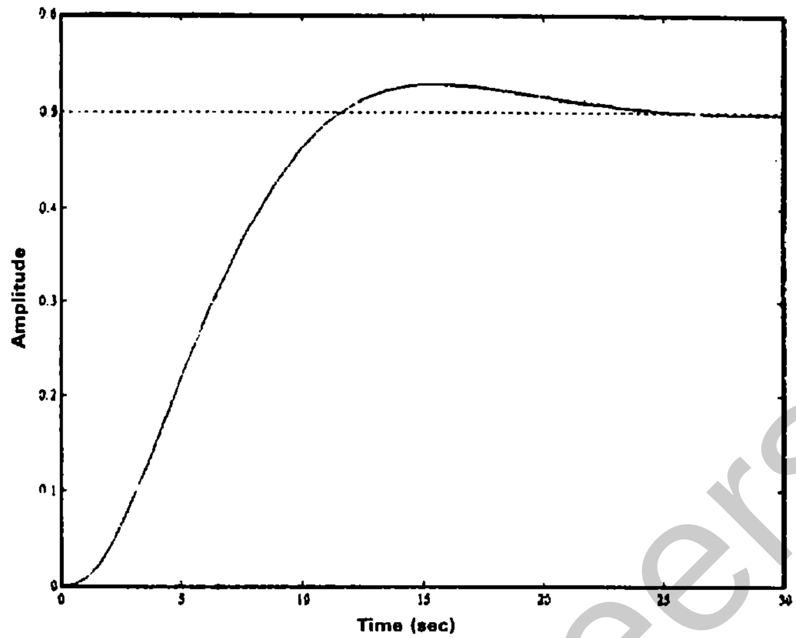


Figure 7.18 Output feedback conventional control step response for system of Example 7.2.6.

reduced to 5 seconds and 11 seconds, respectively, resulting in an improvement of 20% and 60% in those times, while the overshoot is nearly zero. As a follow-up, one may modify (or edit) the rules to speed up the response even further.

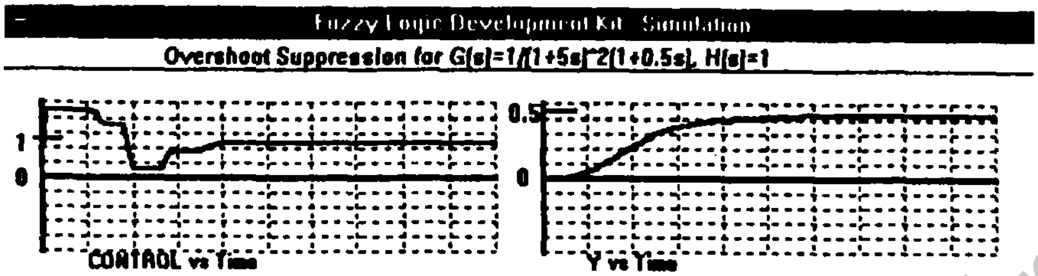
7.2.7 *Analysis of Fuzzy Control Systems*

In this section, some results of Tanaka and Sugeno (1992) with respect to analysis of feedback fuzzy control systems will be briefly discussed. This section would use Takagi-Sugeno models to develop fuzzy block diagrams and fuzzy closed-loop models.

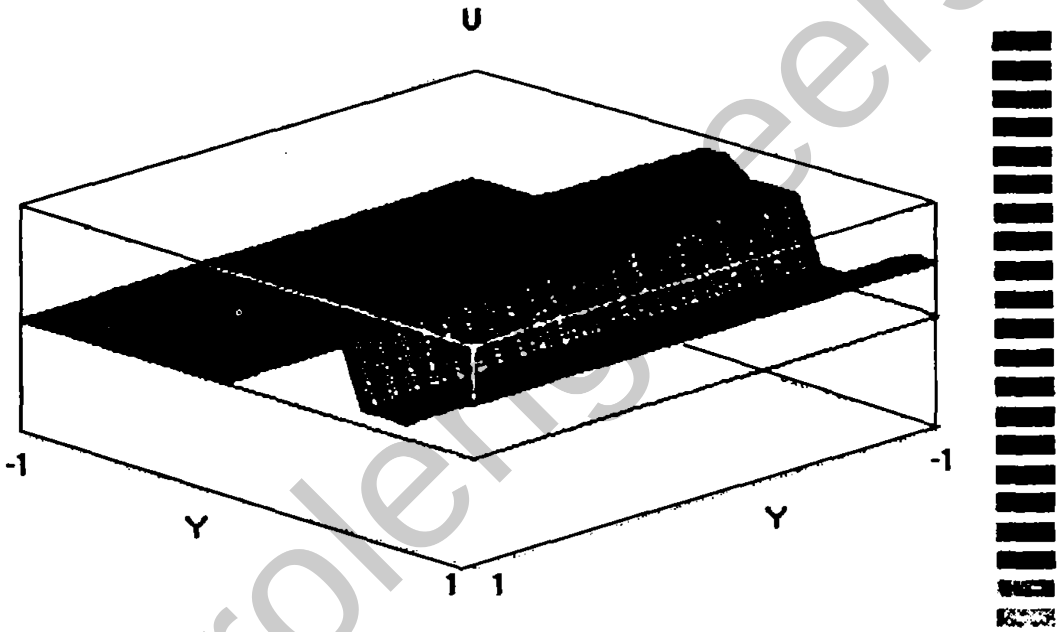
Consider a typical Takagi-Sugeno fuzzy plant model represented by implication P^i in Figure 7.20.

$$\begin{aligned} P^i:\quad & \mathit{IF}\ x(k) \text{ is } A_1^i \text{ and} \ldots \text{and } x(k-n+1) \text{ is } A_n^i \text{ and} \\ & u(k) \text{ is } B_1^i \text{ and} \ldots \text{and } u(k-m+1) \text{ is } B_m^i \\ \text{THEN } & x^i(k+1) = a_0^i + a_1^i x(k) + \ldots + a_n^i x(k-n+1) + \\ & \qquad b_1^i u(k) + \ldots + b_m^i u(k-m+1) \end{aligned} \tag{7.2.25}$$

controlengineers.ir
@controlengineers



(a)



(b)

Figure 7.19 Responses of overshoot suppressing fuzzy controller for the system of Example 7.2.6.

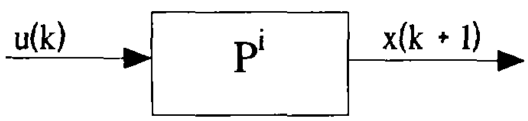


Figure 7.20 A single-input, single-ouput fuzzy block represented by *i*th implication P^i.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $P^i (i = 1, \ldots, \ell)$ is the ith implication, ℓ is the total number of implications, $a_p^i, p = 1, \ldots, n$ and $b_q^i, q = 1, 2, \ldots, m$ are constant consequent parameters, k is time sample, $x(k), \ldots, x(k - n + 1)$ are input variables, n and m are the number of antecedents for states and inputs, respectively. The terms A_p^i and B_q^i are fuzzy sets whose membership functions are defined as follows.

Definition 7.1 A fuzzy set A satisfying the following properties is said to have a piecewise-continuous polynomial (PCP) membership function $A(x)$:

$$\text{(i)} \quad A(x) = \begin{cases} \mu_1(x) & , \quad x \in [p_0, p_1] \\ \vdots & \\ \mu_s(x) & , \quad x \in [p_{s-1}, p_s] \end{cases} \tag{7.2.26}$$

where $\mu_i(x) \in [0,1]$ for $x \in [p_{i-1}, p_i]$, $i = 1, 2, \ldots, s$, and $-\infty = p_0 < p_1 < \ldots < p_{s-1} < p_s = \infty$.

$$\text{(ii)} \quad \mu_i(x) = \sum_{j=0}^{n_i} c_j^i x^j \tag{7.2.27}$$

where c_j^i are known parameters of polynomials $\mu_i(x)$.

Note that the convexity has not been brought into this definition, because, as it will be seen later, this property may not be preserved in the analysis of fuzzy block diagrams.

Given the inputs

$$\mathbf{x}(k) \triangleq [x(k) \quad x(k-1) \ldots x(k-n+1)]^T$$

and (7.2.28)

$$\mathbf{u}(k) \triangleq [u(k) \quad u(k-1) \ldots u(k-m+1)]^T$$

the final defuzzified output of the inference is given by a weighted average of $x^i(k + 1)$ values:

$$x(k+1) = \sum_{i=1}^{\ell} w^i x^i(k+1) \Big/ \sum_{i=1}^{\ell} w^i \tag{7.2.29}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where it is assumed that the denominator of (7.2.29) is positive, and $x^i(k+1)$ is calculated from the ith implication, and the weight w^i refers to the overall truth value of the ith implication premise for the inputs (7.2.28), i.e.,

$$w^i = \prod_{p=1}^{n} A_p^i\left(x(k-p+1)\right) \times \prod_{q=1}^{m} B_q^i\left(u(k-q+1)\right) \tag{7.2.30}$$

where "×" stands for a Cartesian product. Note that regardless of the number of fuzzy implications, the rules (7.2.25) represent a highly nonlinear and complicated mapping function.

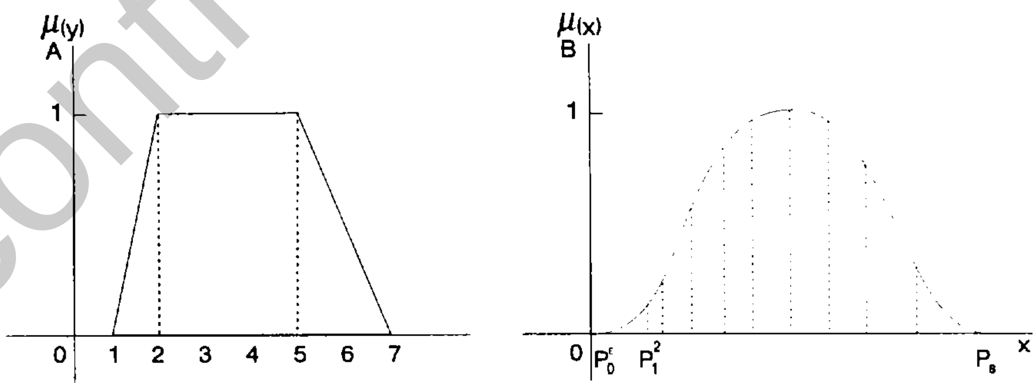
In view of vector notations in Equation (7.2.28), the implication (7.2.25) can be represented in the following compact form,

$$P^i: \text{ IF } \mathbf{x}(k) \text{ is } \mathbf{A}^i \text{ and } \mathbf{u}(k) \text{ is } \mathbf{B}^i$$

$$\text{THEN } x^i(k+1) = a_0^i + \sum_{p=1}^{n} a_p^i x(k-p+1) + \sum_{q=1}^{m} b_q^i u(k-q+1) \tag{7.2.31}$$

where $\mathbf{A}^i \triangleq \left[A_1^i \; A_2^i \cdots A_n^i\right]^T$, $\mathbf{B}^i \triangleq \left[B_1^i \; B_2^i \cdots B_m^i\right]^T$ and "$\mathbf{x}(k)$ is $\mathbf{A}^i$" are equivalent to antecedent $x(k)$ is "A_1^i and . . . and $x(k-n+1)$ is A_n^i."

Example 7.2.7 Consider the trapezoidal and Gaussian membership functions shown in Figure 7.21. It is desired to check if they are piecewise continuous polynomials (PCP).



(a) Trapezoidal (b) Gaussian

Figure 7.21 Two example membership functions.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

SOLUTION: For the trapezoidal function in Figure 7.21a, the set $A(x)$ can be written as

$$A(x)=\begin{cases} 0 & x\in(-\infty,1] \\ x-1 & x\in[1,\ 2] \\ 1 & x\in[2,\ 5] \\ -0.5x+3.5 & x\in[5,\ 7] \\ 0 & x\in[7,\infty) \end{cases} \tag{7.2.32}$$

which satisfies Definition 7.1. Note that all the sets (triangular shapes) in Example 7.2.1 are PCP type. However, for the Gaussian type of functions, as in Figure 7.21b, Definition 7.1 does not apply. One way to create a continuous-piecewise membership function from Gaussian function is to approximate it by a finite number of linear segments by, say, Taylor series expansion.

Now, consider the following proposition (Tanaka and Sugeno, 1992).

Proposition 7.1 *Given two PCP fuzzy sets $A^1(x)$ and $A^2(x)$ show that their product $B(x) = A^1(x) \times A^2(x)$ is also PCP.*

PROOF: In view of Equation (7.2.26), let $A^i(x)$, $i = 1, 2$ be represented by

$$A^1(x)=\begin{cases} \mu_1^1(x) & x\in\left[p_0^1,\ p_1^1\right] \\ \vdots & \vdots \\ \mu_s^1(x) & x\in\left[p_{s-1}^1,\ p_s^1\right] \end{cases}$$

$$A^2(x)=\begin{cases} \mu_1^2(x) & x\in\left[p_0^2,\ p_1^2\right] \\ \vdots & \vdots \\ \mu_\upsilon^2(x) & x\in\left[p_{\upsilon-1}^2,\ p_\upsilon^2\right] \end{cases} \tag{7.2.33}$$

then $B(x) = A^1(x) \times A^2(x)$ can be calculated from

$$B(x)=\mu_i^1(x)\times\mu_j^2(x) \tag{7.2.34}$$

Here x satisfies $x\in\left[p_{i-1}^1,\ p_i^1\right]\cap\left[p_{j-1}^2,\ p_j^2\right]$ for $i = 1, \ldots, s; j = 1, \ldots, \upsilon$ and the set for x is assumed to be nonempty. It is obvious that $B(x)$ is a piecewise polynomial membership function. Moreover, $B(x)$ which is the product of two continuous functions is itself continuous. Thus, $B(x)$ is PCP. Q.E.D.■

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Example 7.2.8 Consider the PCP fuzzy set $B(x)$ described by

$$B(x)=\begin{cases} 0 & x\in(-\infty, 2] \\ x-2 & x\in[2, 3] \\ -0.5x+2.5 & x\in[3, 4] \\ 0.5 & x\in[4, 5] \\ -0.25x+1.75 & x\in[5, 7] \\ 0 & x\in[7, \infty) \end{cases} \tag{7.2.35}$$

and PCP fuzzy set $A(x)$ described by Equation (7.2.32). It is desired to find their products $C(x) = A(x) \times B(x)$.

SOLUTION: Moving along the universe of discourse of the two sets starting from $-\infty$ on to ∞ and multiplying the functions during every definable interval for both sets, the following product set $C(x)$ results.

$$C(x)=\begin{cases} 0 & x\in(-\infty, 2] \\ x-2 & x\in[2, 3] \\ -0.5x+2.5 & x\in[3, 4] \\ 0.5 & x\in[4, 5] \\ 0.125x^2-1.75x+6.125 & x\in[5, 7] \\ 0 & x\in[7, \infty) \end{cases}$$

Figure 7.22 shows the sets $A(x)$, $B(x)$, and $C(x)$. It is evident that $C(x)$ is also PCP, but it is not convex even though both $A(x)$ and $B(x)$ are convex sets.

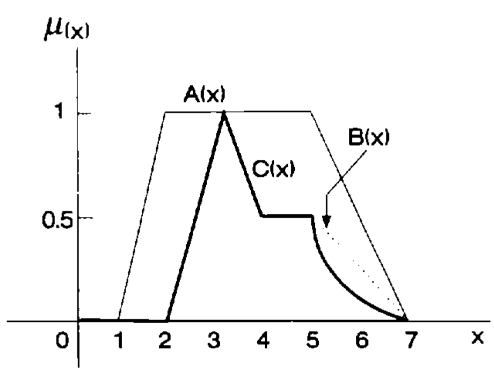


Figure 7.22 Product of two piecewise-continuous polynomial fuzzy sets.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

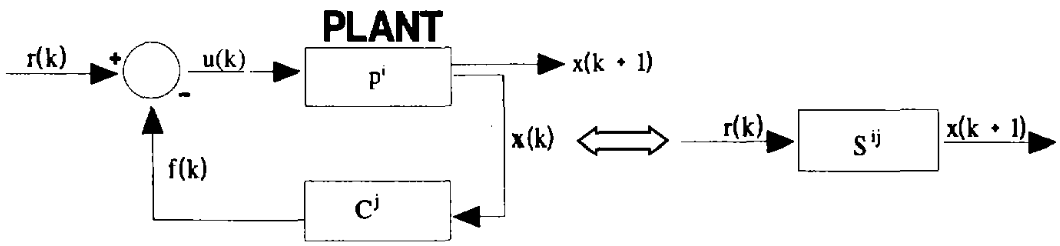


Figure 7.23 A fuzzy control system depicted by two implications and its equivalent implication.

Since the product of two PCP fuzzy sets can be considered as a series connection of two fuzzy blocks of the type in Figure 7.20, it is concluded that the convexity of fuzzy sets in succession is not preserved in general. Now let us consider a fuzzy control system whose plant model and controller are represented by fuzzy implications as depicted in Figure 7.23. In this figure, $r(k)$ represents a reference input. The plant implication p^i is already defined by Equation (7.2.31), while the controller's jth implication is given by

$$C^j:\ \text{IF } \mathbf{x}(k) \text{ is } \mathbf{D}^j \text{ and } \mathbf{u}(k) \text{ is } \mathbf{F}^j$$
$$\text{THEN } f^j(k)=c_0^j+\sum_{p=1}^{n}c_p^j x(k-p+1) \tag{7.2.36}$$

where $\mathbf{D}^j \triangleq \left[D_1^j\ D_2^j \ldots D_n^j\right]^T$, $\mathbf{F}^j \triangleq \left[F_1^j\ F_2^j \ldots F_m^j\right]^T$, and of course $u(k) = r(k) - f(k)$. The equivalent implication S^{ij} is given by

$$S^{ij}:\ \text{IF } \mathbf{x}(k) \text{ is } \left(\mathbf{A}^i \text{ and } \mathbf{D}^j\right) \text{ and } v^*(k) \text{ is } \left(\mathbf{B}^i \text{ and } \mathbf{F}^j\right)$$
$$\text{THEN } x^{ij}(k+1)=a_0^i-b^ic_0^j+b^ir(k)+\sum_{p=1}^{n}\left(a_p^i-b^ic_p^j\right)x(k-p+1) \tag{7.2.37}$$

where $i = 1, \ldots, \ell_1$ and $j = 1, \ldots, \ell_2$; the antecedent "$\mathbf{x}(k)$ is ($\mathbf{A}^i$ and $\mathbf{D}^j$)" is equivalent to $x(k)$ is $\left(A_1^i \text{ and } D_1^j\right)$ and … and $x(k-n+1)$ is $\left(A_n^i \text{ and } D_n^j\right)$. The membership function of $\left(A_1^i \text{ and } D_1^j\right)$ is defined as $A_1^i\,(x(k))\times D_1^j(x(k))$ as discussed in Proposition 7.1. The term $v^*(k)$ is

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

defined by

$$v^*(k) = \left[r(k) - e^*(x(k)),\ r(k-1) - e^*(x(k-1)), \right.$$
$$\left. \ldots, r(k-m+1) - e^*(x(k-m+1)) \right]^T$$

where $e^*(\cdot)$ is the input-output mapping function of block C^j in Figure 7.23, i.e., $f(k) = e^*(x(k))$. The evaluation of $e^*(\cdot)$ is dependent on the fuzzy sets $\mathbf{A}^i$, $\mathbf{B}^i$, $\mathbf{D}^j$, and $\mathbf{F}^j$ and can be found in Tanaka and Sugeno (1992). Note that if $\ell_1 = \ell_2 = \ell$ and $A_1^i = D_1^i, \ldots, A_n^i = D_n^i$, $B_1^i = F_1^i, \ldots, B_m^i = F_m^i$ for $i = 1, 2, \ldots, \ell$, then the fuzzy blocks of Figure 7.23 and the implication in Equation (7.2.37) is reduced to

$$\begin{aligned} S^{ij}:\ & \text{IF } \mathbf{x}(k) \text{ is } \left(\mathbf{A}^i \text{ and } \mathbf{A}^j\right) \text{ and } v^*(k) \text{ is } \left(\mathbf{B}^i \text{ and } \mathbf{B}^j\right) \\ & \text{THEN } x^{ij}(k+1) = a_0^i - b^i c_0^j + b^i r(k) + \sum_{p=1}^{n} \left(a_p^i - b_p^i c_p^j\right) x(k-p+1) \end{aligned} \tag{7.2.38}$$

for $i, j = 1, \ldots, \ell$. The following numerical example illustrates how S^{ij} can be obtained.

Example 7.2.9 Consider a fuzzy feedback control system of the type shown in Figure 7.23 with the following implications:

P^1: IF $x(k)$ is A^1 THEN $x^1(k+1) = 1.85x(k) - 0.65x(k-1) + 0.35u(k)$

P^2: IF $x(k)$ is A^2 THEN $x^2(k+1) = 2.56x(k) - 0.135x(k-1) + 2.22u(k)$

C^1: IF $x(k)$ is D^1 THEN $f^1(k) = k_1^1 x(k) + k_2^1 x(k-1)$

C^2: IF $x(k)$ is D^2 THEN $f^2(k) = k_1^2 x(k) + k_2^2 x(k-1)$

It is desired to find the closed-loop implications S^{ij}, $i = 1, 2$, and $j = 1, 2$.

SOLUTION: Noting that $u(k) = r(k) - f(k)$ in Figure 7.23 and the relations of (7.2.37), we have

$$\begin{aligned} S^{11}:\ & \text{IF } x(k) \text{ is } (A^1 \text{ and } D^1), \text{ THEN } x^{11}(k+1) = (1.85 - 0.35k_1^1)\, x(k) + \\ & (-0.65 - 0.35k_2^1)\, x(k-1) + 0.35r(k) \end{aligned} \tag{7.2.39}$$

controlengineers.ir
@controlengineers

S^{12}: IF $x(k)$ is (A^1 and D^2), THEN $x^{12}(k+1) = (1.85 - 0.35k_1^2)\, x(k) + (-0.65 - 0.35k_2^2)\, x(k-1) + 0.35r(k)$

(7.2.40)

S^{21}: IF $x(k)$ is (A^2 and D^1), THEN $x^{21}(k+1) = (2.56 - 2.22\,k_1^1)\, x(k) + (-0.135 - 2.22k_2^1)\, x(k-1) + 2.22r(k)$

(7.2.41)

S^{22}: IF $x(k)$ is (A^2 and D^2), THEN $x^{22}(k+1) = (2.56 - 2.22\,k_1^2)\, x(k) + (-0.135 - 2.22k_2^2)\, x(k-1) + 2.22r(k)$

(7.2.42)

Note that for the case where $A^i = D^i$, $i = 1, 2$, the fuzzy sets within the antecedents of (7.2.39) and (7.2.42) would be changed to (A^i and A^i) for $i =$ 1 and 2, respectively. Similarly, the fuzzy sets in cross-coupled terms of Equations (7.2.40) and (7.2.41) will become (A^1 and A^2) for both implications. This fact would allow one to combine the middle two implications to result in the following one:

$$
\begin{aligned}
2S^{12*}:\ & \text{IF } x(k) \text{ is } (A^1 \text{ and } A^2) \\
& \text{THEN } x^{12*}(k+1) = \left[2.205 - (2.22k_1^1 + 0.35k_1^2)/2\right] x(k) \\
& + \left[-0.3925 - (2.22k_2^1 + 0.35k_2^2)/2\right] x(k-1) + 1.285r(k)
\end{aligned}
\qquad (7.2.43)
$$

where the coefficients within the brackets are averages of those in (7.2.40) and (7.2.41). Therefore, our reduced implications are S^{11} in (7.2.39), $2S^{12*}$ in (7.2.43) and S^{22} in (7.2.42). The number "2" in front of $2S^{12*}$ symbolizes the fact that this implication is counted as double and should be so counted in evaluating the output,

$$
x(k+1) = \frac{w^1 x^{11}(k+1) + 2w^2 x^{12*}(k+1) + w^3 x^{22}(k+1)}{w^1 + 2w^2 + w^3}
$$

where w^1, w^2, and w^3 are the weights associated with implications S^1, S^{12*}, and S^{22}, respectively. More discussions and a parallel connection of fuzzy blocks will come forth in the problem section at the end of the chapter.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

7.3 Stability of Fuzzy Control Systems

7.3.1 Introduction

One of the most fundamental issues in any control system—fuzzy or others—is the stability. Briefly, a system is said to be stable if it would come to its equilibrium state after any external inputs, initial conditions, and/or disturbances have impressed the system. The issue of stability is of even greater relevance when questions of safety, lives, and environment are at stake as in such systems as nuclear reactors, traffic systems and airplane autopilots. The stability test for fuzzy control systems, or lack of it, has been a subject of criticism by many control engineers in some control engineering literature (*IEEE Control Systems* Magazine, 1993).

Almost any linear or nonlinear system under the influence of a closed-loop crisp controller has one type of stability test or other. For example, the stability of a linear time-invariant system can be tested by a wide variety of methods such as Routh-Hurwitz, R. root locus, Bode plots, Nyquist criterian, and even through traditionally nonlinear systems methods of Lyapunov, Popov, and circle criterion. The common requirement in all these tests is the availability of a mathematical model—be it in time or frequency domain. A reliable mathematical model for a very complex and large-scale system may, in practice, be unavailable or unfeasible. In such cases, a fuzzy controller may be designed based on expert knowledge or experimental practice. However, the issue of the stability of a fuzzy control system still remains and must be addressed. The aim of this section is to present an up-to-date survey of available techniques and tests for fuzzy control systems stability.

Fuzzy controllers represent static nonlinearities (Bretthauer and Opitz, 1994) and as such the stability problems belong to nonlinear control systems. In this section, a survey of fuzzy control systems' stability will be given and a few more promising approaches will be described in more detail.

controlengineers.ir

7.3.2 Fuzzy Control Systems' Stability Classes

From the viewpoint of stability a fuzzy controller can be either acting as a conventional (low-level) controller or as a supervisory (high-level) controller as described in Section 7.2 and to be discussed in Section 8.2. Depending on the existence and nature of a system's mathematical model and the level in which fuzzy rules are being utilized for control and robustness, four classes or fuzzy control stability problems can be

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

distinguished. These four classes are:

Class 1: Process model is crisp and linear and fuzzy controller is low level.
Class 2: Process model is crisp and nonlinear and the fuzzy controller is low level.
Class 3: Process model (linear or nonlinear) is crisp and a fuzzy tuner or an adaptive fuzzy controller is present at high level.
Class 4: Process model is fuzzy and fuzzy controller is low level.

Figures 7.24–7.27 show all the four classes of fuzzy control systems whose stability is of concern. In this presentation we are concerned mainly with the first three classes. For the last class, traditional nonlinear control theory would fail and is beyond the scope of this section. It will be discussed very briefly. The techniques for testing the stability of the first two classes of systems (Figures 7.24 and 7.25) are shown in Table 7.2. As shown, the methods are divided into two main groups—time and frequency.

Time-Domain Methods

The state-space approach has been considered by many authors (Aracil *et al.*, 1988; Chen and Tsao, 1989; Wang *et al.*, 1990; Hojo *et al.*, 1991; Hwang and Liu, 1992; Driankov *et al.*, 1993; Kang, 1993; Demaya *et al.*, 1994). The basic approach here is to subdivide the state space into a finite number of cells based on the definitions of the membership functions. Now, if a separate rule is defined for every cell, a cell-to-cell trajectory can be constructed from the system's output induced by the new outputs of the fuzzy controller. If every cell of the modified state space is checked, one can identify all the equilibrium points, including the system's stable region. This method should be used with some care since the inaccuracies in the modified description could cause oscillatory phenomenon around the equilibrium points.

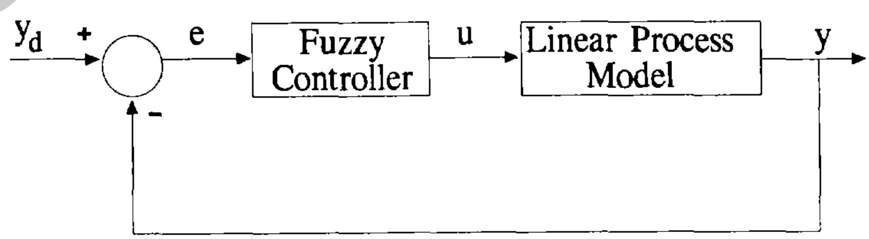


Figure 7.24 Class 1 of fuzzy control system stability problem.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

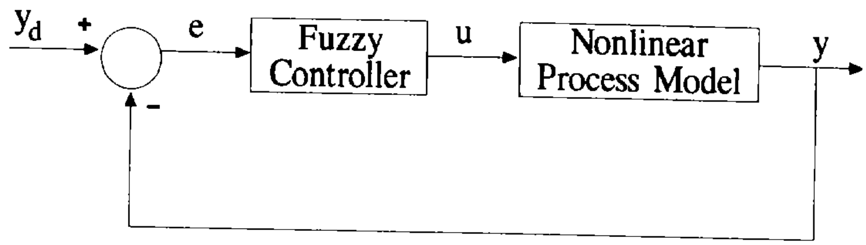


Figure 7.25 Class 2 of fuzzy control system stability problem.

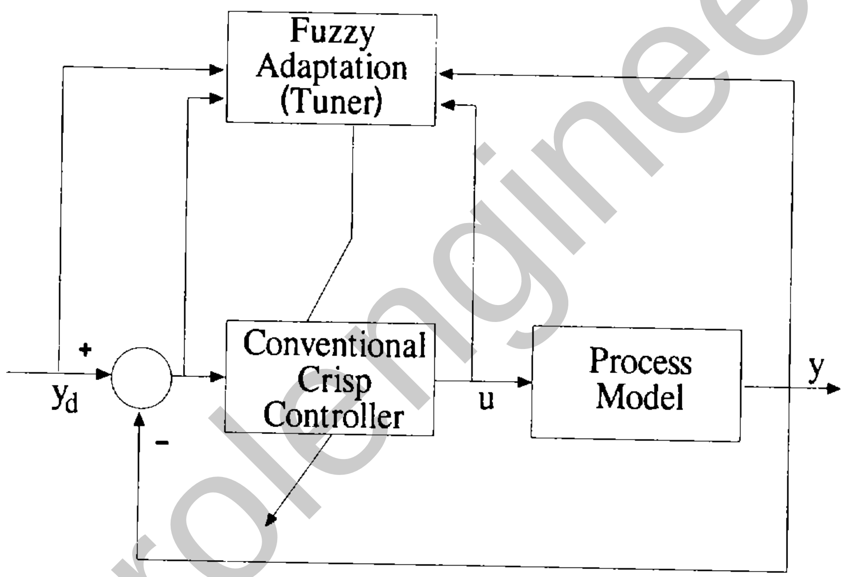


Figure 7.26 Class 3 of fuzzy control system stability problem.

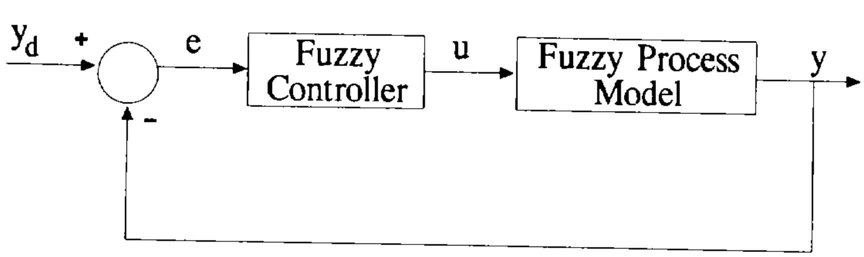


Figure 7.27 Class 4 of fuzzy control system stability problem.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 7.2 Stability Analysis Methods for Fuzzy Control Systems with Known Model

Stability Analysis Methods

Time Domain	Frequency Domain
State-space	Harmonic Balance
Lyapunov Theory	Circle Criterion
Hyperstability Theory	Popov Criterion
Bifurcation Theory	
Graph Theory	

The second class of methods is based on Lyapunov's method. Several authors (Langari and Tomizuka, 1990; Bouslama and Ichikawa, 1992; Chen *et al.*, 1993; Chen, 1987; Driankov *et al.*, 1993; Franke, 1993; Gertler and Chang, 1986; Hojo *et al.*, 1991; Kiszka *et al.*, 1985; Tanaka and Sugeno, 1992; Wang, 1993, 1994a,b; Tahani and Sheikholeslam, 1994) have used this theory to come up with criterion for stability of fuzzy control systems. The approach is along the same lines as in Section 3.2, i.e., show that the time derivative of the Lyapunov function at the equilibrium point is negative semi-definite. Many approaches have been proposed. One approach is to define a Lyapunov function and then derive the fuzzy controller's architecture out of the stability conditions. Another approach uses Aiserman's method (Bretthauer and Opitz, 1994) to find an adopted Lyapunov function, while representing the fuzzy controller by a nonlinear algebraic function $u = f(y)$, when y is the system's output. A third method calls for the use of so-called *facet functions*, where the fuzzy controller is realized by boxwise multilinear facet functions with the system being described by a state space model. To test stability, a numerical parameter optimization scheme is needed.

The *hyperstability* approach, considered by other authors (Barreiro and Aracil, 1992; Opitz, 1993, 1994) has been used to check stability of systems depicted in Figure 7.24. The basic approach here is to restrict the input-output behavior of the nonlinear fuzzy controller by

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

inequality and to derive conditions for the linear part of the closed-loop system to be satisfied for stability.

Bifurcation theory (Driankov *et al.*, 1993) can be used to check stability of fuzzy control systems of the class described in Figure 7.25. This approach represents a tool in deriving stability conditions and robustness indices for stability from small gain theory. The fuzzy controller, in this case, is described by a nonlinear vector function. The stability, in this scheme, could only be lost if one of the following conditions become true: (i) the origin becomes unstable if a pole crosses the imaginary axis into the right half-plane—static bifurcation, (ii) the origin becomes unstable if a pair of poles would cross over the imaginary axis and assumes positive real parts—Hopf bifurcation—or (iii) new additional equilibrium points are produced.

The last time-domain method is the use of *graph theory* (Driankov *et al.*, 1993). In this approach conditions for special nonlinearities are derived to test the BIBO stability.

Frequency-Domain Methods

There are three primary groups of methods which have been considered here (see Table 7.2). The *harmonic balance* approach, considered by Braee and Rutherford (1978, 1979), and Kickert and Mamdani (1978), among others, has been used to check the stability of the first two classes of fuzzy control systems (see Figures 7.24 and 7.25). The main idea is to check if permanent oscillations occur in the system and whether these oscillations with known amplitude or frequency are stable. The nonlinearity (fuzzy controller) is described by a complex-valued describing function and the condition of harmonic balance is tested. If this condition is satisfied, then a permanent oscillation exists. This approach is equally applicable to MIMO systems.

The *circle criterion* (Aracil *et al.*, 1988, 1991; Opitz, 1994; Ray and Majumder, 1984; Ray *et al.*, 1984) and Popov criterion (Böhm, 1992; Bühler, 1993) have been used to check stability of the first class of systems (Figure 7.24). In both criteria, certain conditions on the linear process model and static nonlinearity (controller) must be satisfied. It is assumed that the characteristic value of the nonlinearity remains within certain bounds, and the linear process model must be open-loop stable with proper transfer function. Both criteria can be graphically evaluated in simple manners.

The stability of adaptive fuzzy control systems has been treated in detail by Wang (1994a, 1994b) and is best used when it augments the design process as is described in Section 8.2.3.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

7.3.3 *Lyapunov Stability of Fuzzy Control Systems*

As mentioned before, one of the most fundamental criterion of any control system is to ensure stability as a part of the design process. In this section, some theoretical results on this important topic are detailed, followed by a few stability-guaranteeing design examples.

As in Section 7.2.7 we begin with the ith implication of a fuzzy system:

$$\begin{aligned} P^i:\quad & \text{IF } x(k) \textit{ is } A_1^i \textit{ and} \ldots \textit{and } x(k-n+1) \textit{ is } A_n^i \\ & \text{THEN } x^i(k+1) = a_1^i x(k) + \ldots + a_n^i x(k-n+1) \end{aligned} \tag{7.3.1}$$

with $i = 1, \ldots, l$. It is noted that this implication is the same as (7.2.25) except since we are dealing with Lyapunov stability the inputs $u(k)$ are absent. The stability of a fuzzy control system with the presence of the inputs will be considered shortly. The consequent part of (7.3.1) represents a set of linear subsystems and can be rewritten as (Tanaka and Sugeno, 1992), i.e.,

$$\begin{aligned} & \text{IF } x(k) \textit{ is} A_1^i \text{ and} \ldots \text{and } x(k-n+1) \text{ is} A_n^i \\ & \text{THEN } \mathbf{x}(k+1) = \mathbf{A}_i\, \mathbf{x}(k) \end{aligned} \tag{7.3.2}$$

where $\mathbf{x}(k)$ is defined by Equation (7.2.28)and $n \times n$ matrix $\mathbf{A}_i$ is

$$\mathbf{A}_i = \begin{bmatrix} a_1^i & a_2^i & \cdots & a_{n-1}^i & a_n^i \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & & \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{7.3.3}$$

The output of the fuzzy system described by Equations (7.3.1)–(7.3.3) is given by

$$\mathbf{x}(k+1) = \frac{\sum_{i=1}^{l} w^i\, \mathbf{A}_i\, \mathbf{x}(k)}{\sum_{i=1}^{l} w^i} \tag{7.3.4}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where w^i is the overall truth value of the ith implication defined by Equation (7.2.30) and l is the total number of implications. Lyapunov stability of large-scale systems was described in Section 7.3.2. As a sequel, a theorem and a lemma for crisp discrete-time systems are first given before Lyapunov stability of fuzzy control systems is presented.

Theorem 7.1 *Consider a discrete-time system described by*

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) \tag{7.3.5}$$

where $\mathbf{x}(k) \in R^n$, $\mathbf{f}(\cdot)$ *is an n-dimensional nonlinear function with a property* $\mathbf{f}(0) = 0$ *for all k. Assume that a scalar continuous function* $v(\mathbf{x}(k))$ *exists which has the following properties*

a) $v(0) = 0$

b) $v(\mathbf{x}(k)) > 0$ *for* $\mathbf{x}(k) \neq 0$

c) $\lim_{\|x(k)\| \to \infty} v\left(\mathbf{x}(k)\right) = \infty$

d) $\Delta v(\mathbf{x}(k)) < 0$ for all $\mathbf{x}(k) \neq \mathbf{0}$

Then the equilibrium point $\mathbf{x}(k) = 0$ *of* (7.3.5), *obtained by solving* $\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k))$, *is asymptotically stable in the large for all k and* $v(\mathbf{x}(k))$ *is a Lyapunov function.*

Clearly, this theorem is the discrete version of the standard Lyapunov stability theorem (Jamshidi *et al.*, 1992). Now, consider the following:

Lemma 7.1 *If* **P** *is a positive definite matrix such that*

$$\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} < 0 \quad \textit{and} \quad \mathbf{B}^T\mathbf{P}\mathbf{B} - \mathbf{P} < 0$$

where **A**, **B**, *and* **P** *are n × n matrices, then*

$$\mathbf{A}^T\mathbf{P}\mathbf{B} + \mathbf{B}^T\mathbf{P}\mathbf{A} - 2\mathbf{P} < 0$$

PROOF: Evaluating the left side of the inequality would result in

$$\begin{aligned}\mathbf{A}^T\mathbf{P}\mathbf{B} + \mathbf{B}^T\mathbf{P}\mathbf{A} - 2\mathbf{P} &= -(\mathbf{A} - \mathbf{B})^T\mathbf{P}(\mathbf{A} - \mathbf{B}) + \mathbf{A}^T\mathbf{P}\mathbf{A} + \mathbf{B}^T\mathbf{P}\mathbf{B} - 2\mathbf{P}\\ &= -(\mathbf{A} - \mathbf{B})^T\mathbf{P}(\mathbf{A} - \mathbf{B}) + (\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P}) + (\mathbf{B}^T\mathbf{P}\mathbf{B} - \mathbf{P})\end{aligned}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

since the last two terms are assumed to be positive-definite and since $\mathbf{P}$ is a positive-definite matrix,

$$-(\mathbf{A}-\mathbf{B})^{T}\mathbf{P}(\mathbf{A}-\mathbf{B}) \leq 0$$

and the lemma's conclusion follows. We now present the first stability result of fuzzy control systems, due to Tanaka and Sugeno (1992), among others.

Theorem 7.2 *The equilibrium point of a fuzzy system (7.3.4) is globally asymptotically stable if there exists a common positive definite matrix* $\mathbf{P}$ *for all subsystems such that*

$$\mathbf{A}_i^T \mathbf{P}\mathbf{A}_i - \mathbf{P} < \mathbf{0} \tag{7.3.6}$$

for $i = 1, \ldots, l$.

PROOF: Let the scalar Lyapunov function $v(\mathbf{x}(k)) = \mathbf{x}^{T}(k)\mathbf{P}\mathbf{x}(k)$, where $\mathbf{P}$ is an $n \times n$ positive definite matrix. The function $v(\mathbf{x}(k))$ satisfies the following properties: (i) $v(\mathbf{0}) = 0$, (ii) $v(\mathbf{x}(k)) > 0$ for $\mathbf{x}(k) \neq 0$, and (iv) $\lim_{\|x(k)\|\to\infty} v(\mathbf{x}(k)) = \infty$.

Next, we evaluate

$$\Delta v(\mathbf{x}(k)) = v(\mathbf{x}(k+1)) - v(\mathbf{x}(k)) = \mathbf{x}^{T}(k+1)\ \mathbf{P}\ \mathbf{x}(k+1) - \mathbf{x}^{T}(k)\ \mathbf{P}\ \mathbf{x}(k)$$

$$= \left(\frac{\sum_{i=1}^{l} w^i \mathbf{A}_i\, \mathbf{x}(k)}{\sum_{i=1}^{l} w^i}\right)^{T} \mathbf{P} \left(\frac{\sum_{i=1}^{l} w^i \mathbf{A}_i\, \mathbf{x}(k)}{\sum_{i=1}^{l} w^i}\right) - \mathbf{x}^{T}(k)\ \mathbf{P}\ \mathbf{x}(\mathrm{k})$$

$$= \mathbf{x}^{T}(k)\left[\left(\frac{\sum_{i=1}^{l} w^i \mathbf{A}_i^T}{\sum_{i=1}^{l} w^i}\right)\mathbf{P}\left(\frac{\sum_{i=1}^{l} w^i \mathbf{A}_i}{\sum_{i=1}^{l} w^i}\right) - \mathbf{P}\right]\mathbf{x}(k)$$

$$= \frac{\sum_{i,j=l}^{l} w^i w^j \mathbf{x}^{T}(k)\left(\mathbf{A}_i^T \mathbf{P}\mathbf{A}_j - \mathbf{P}\right)\mathbf{x}(k)}{\sum_{i,j=1}^{l} w^i w^j}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$= \frac{\left[\sum_{i=1}^{l}(w^i)^2 \mathbf{x}^T(k)\left(\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P}\right)\mathbf{x}(k) + \sum_{i<j}^{l} w^i w^j \mathbf{x}^T(k)\left(\mathbf{A}_i^T\mathbf{P}\mathbf{A}_j + \mathbf{A}_j^T\mathbf{P}\mathbf{A}_i - 2\mathbf{P}\right)\mathbf{x}(k)\right]}{\sum_{i,j=1}^{l} w^i w^j} \tag{7.3.7}$$

where $w^i \geq 0$ for $i = 1, \ldots, l$ and $\sum_{i=1}^{l} w^i > 0$. Now, by assumption of the theorem, Equation (7.3.6), the first matrix is negative-definite and the second matrix relation is negative-definite by Lemma 7.1. Hence, we have $\Delta v(\mathbf{x}(k)) < 0$. Now, in lieu of Theorem 7.1, the fuzzy system (7.3.4) is globally asymptotically stable. Q.E.D. ■

It is noted that the above theorem can be applied to any nonlinear system which can be approximated by a piecewise linear function if the stability condition (7.3.6) is satisfied. Moreover, if there exists a common positive definite matrix $\mathbf{P}$, then all the $\mathbf{A}_i$ matrices are stable. Since Theorem 7.2 is a sufficient condition for stability, it is possible not to find a $\mathbf{P} > 0$ even if all A_i matrices are stable. In other words, a fuzzy system may be globally asymptotically stable even if a $\mathbf{P} > 0$ is not found. The fuzzy system is not always stable even if all the $\mathbf{A}_i$'s are stable. This is illustrated by the following example.

Example 7.3.1 Consider the following fuzzy system

P^1: IF $x(k-1)$ is A^1 THEN $x^1(k+1) = x(k) - 0.2x(k-1)$
P^2: IF $x(k-1)$ is A^2 THEN $x^2(k+1) = -x(k) - 0.2x(k-1)$

where the fuzzy sets A^i, $i = 1, 2$ are shown in Figure 7.28. It is desired to check its stability. Use $x(0) = -0.8$ and $x(1) = 0.8$.

SOLUTION: The system consists of two linear subsystems whose matrices are given by

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -0.2 \\ 1 & 0 \end{bmatrix}, \qquad \mathbf{A}_2 = \begin{bmatrix} -1 & -0.2 \\ 1 & 0 \end{bmatrix}$$

Using the initial conditions and simulating the linear discrete-time systems $\mathbf{x}(k+1) = \mathbf{A}_i\mathbf{x}(k)$ for $i = 1,2$ we can note from Figure 7.29a and

controlengineers.ir
@controlengineers

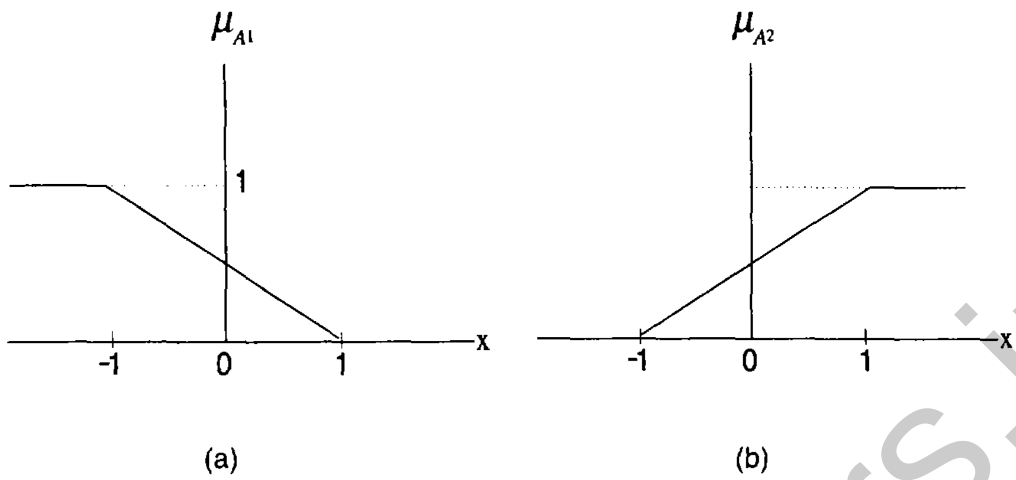


Figure 7.28 Fuzzy sets for Example 7.3.1.

7.29b that these linear subsystems (rules) are in fact stable. However, using Equation (7.3.4), the fuzzy system would become

$$x(k+1)=\left[\left(w^1-w^2\right)/\left(w^1+w^2\right)\right]x(k)-0.2x(k-1)$$

The fuzzy system's output along with the membership functions weights w^i, $i = 1, 2$ are shown in Figure 7.29c and 7.29d, respectively.

As it can be easily seen, while the two individual linear subsystems are stable, the fuzzy system is unstable. The conclusion here is that for this example system, no common **P** matrix can be found indicating that the given fuzzy control system is in fact unstable. It is thus clear that a necessary condition for the existence of common **P** is critical.

Theorem 7.3 *Let* $\mathbf{A}_i$ *be stable and nonsingular matrices for* $i = 1, \ldots, l$. *Then* $\mathbf{A}_i\mathbf{A}_j$ *are stable matrices for* $i, j = 1, \ldots, l$, *if there exists a common positive definite matrix* **P** *such that*

$$\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i-\mathbf{P}<0 \tag{7.3.8}$$

PROOF: Rewriting (7.3.8), we obtain

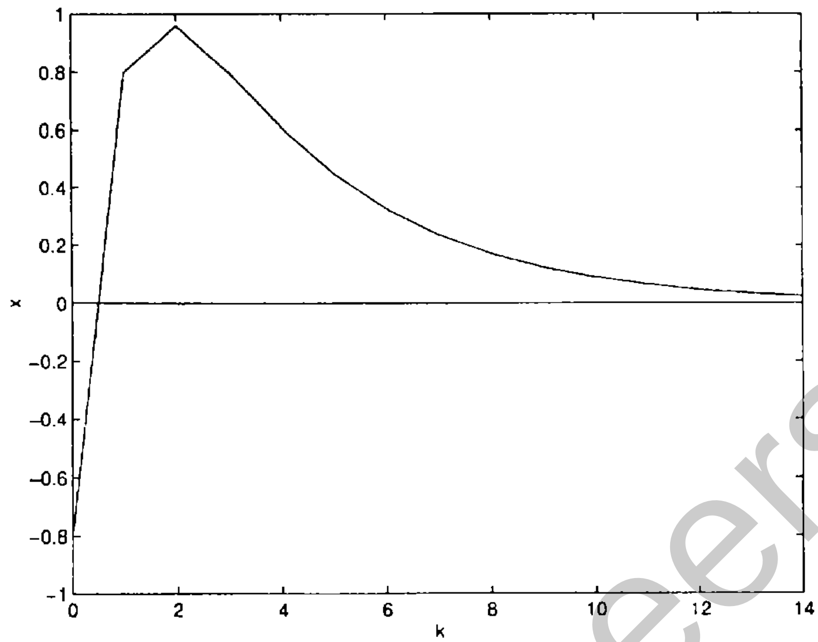
$$\mathbf{P}-\left(\mathbf{A}_i^{-1}\right)^T\mathbf{P}\mathbf{A}_i^{-1}<0$$

since $\left(\mathbf{A}_i^{-1}\right)^T=\left(\mathbf{A}_i^T\right)^{-1}$. Thus, $\mathbf{P}<\left(\mathbf{A}_i^{-1}\right)^T\mathbf{P}\left(\mathbf{A}_i^{-1}\right)$ for $i = 1, \ldots, l$. In view of this inequality and the one **P** in (7.3.8), it follows that

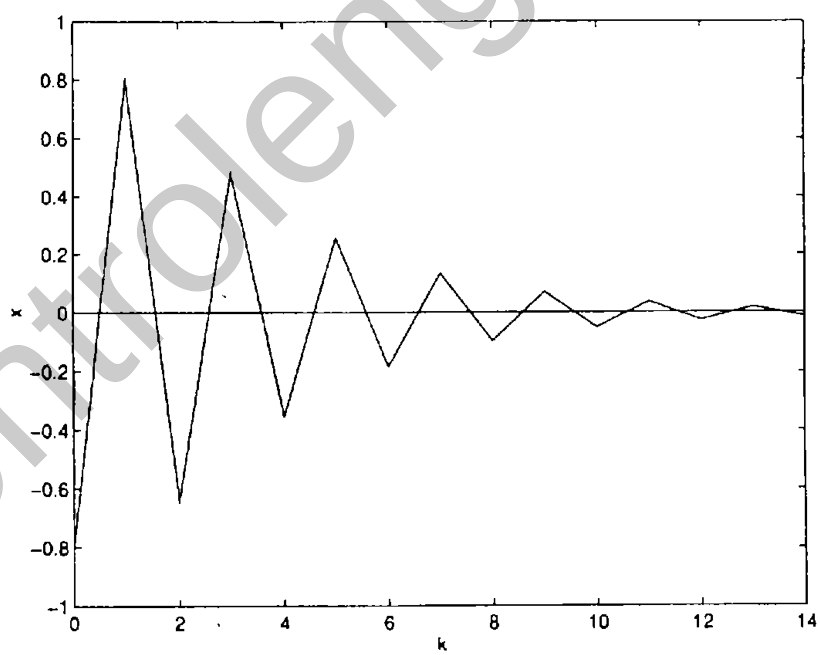
$$\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i<\left(\mathbf{A}_j^{-1}\right)^T\mathbf{P}\left(\mathbf{A}_j^{-1}\right)$$

controlengineers.ir

controlengineers.ir
@controlengineers



(a) Subsystem 1

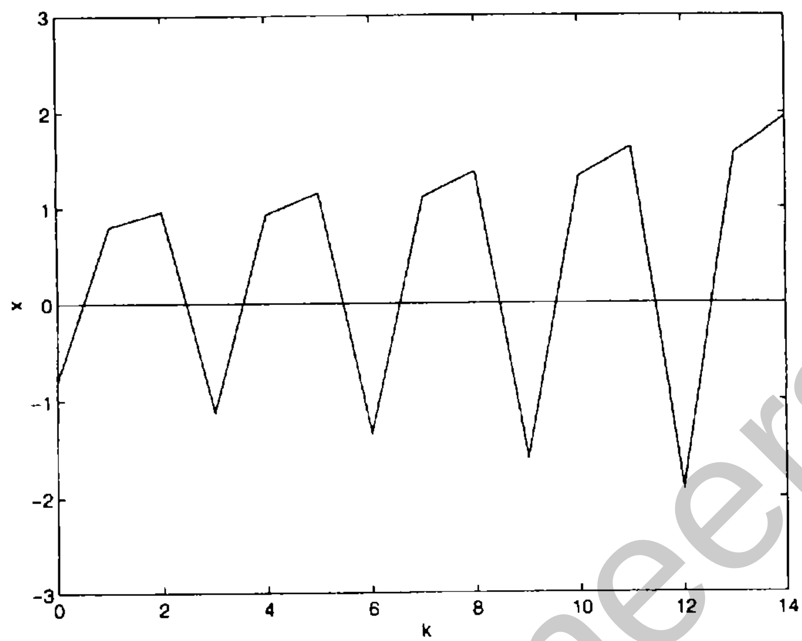


(b) Subsystem 2

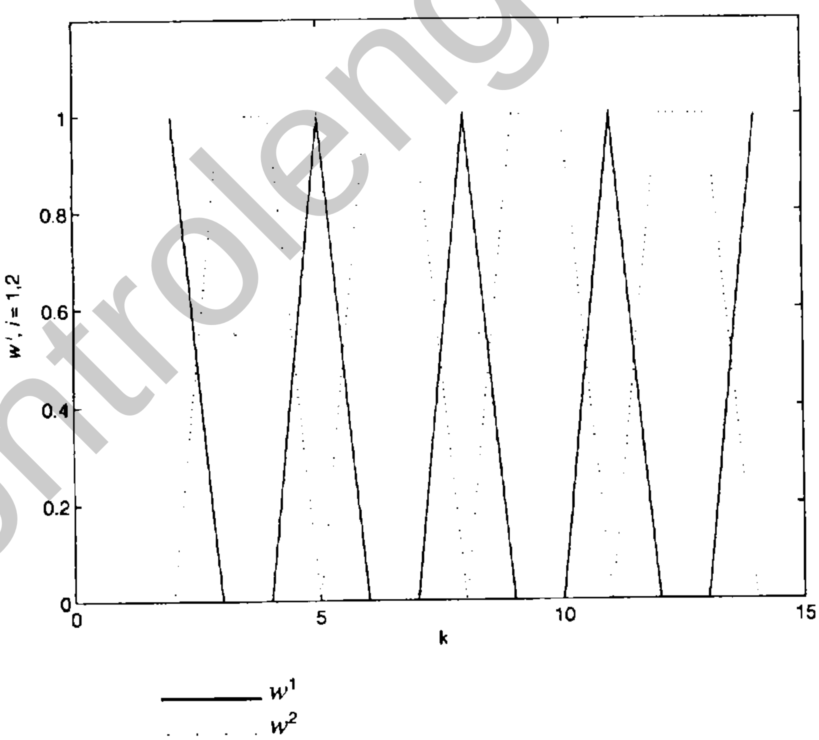
Figure 7.29 (a & b) Simulation results for Example 7.3.1.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(c) Fuzzy System



(d) Membership Function weights

Figure 7.29 (c & d) Simulation results for Example 7.3.1.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

or by pre- and post-multiplying by $\mathbf{A}_j^T$ and $\mathbf{A}_j$, respectively we have $\mathbf{A}_j^T\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i\mathbf{A}_j - \mathbf{P} < 0$. Thus, $\mathbf{A}_i\mathbf{A}_j$ must be a Hurwitz matrix for i, j = 1, ... , l. Q.E.D. ■

In view of this theorem in the system of Example 7.3.1, we have

$$\mathbf{A}_1\mathbf{A}_2 = \begin{pmatrix} -1.2 & -0.2 \\ -1 & -0.2 \end{pmatrix}$$

whose eigenvalues are $\lambda_1 = -1.3708$ and $\lambda_2 = -0.0292$ which clearly indicates that the matrix is unstable.

In summary, to check the stability of a fuzzy system, one must find a common positive definite $\mathbf{P}$. This task may not be an easy one. A possible algorithm to check the stability is to

(1) Find a $\mathbf{P}_i > 0$ such that $\mathbf{A}_i^T\mathbf{P}_i\mathbf{A}_i - \mathbf{P}_i < 0$ for $i = 1, \ldots, l$. If $\mathbf{A}_i$ is stable, it is always possible to find a $\mathbf{P}_i$.

(2) Check if a $\mathbf{P}_j \in \{\mathbf{P}_i | i = 1, \ldots, l\}$ exists such that $\mathbf{A}_i^T\mathbf{P}_j\mathbf{A}_i - \mathbf{P}_j < 0$. If so, a common $\mathbf{P}$ has been obtained. Otherwise go to step 1 for the next value of i.

Example 7.3.2 Consider the following fuzzy system:

P^1: IF $x(k)$ is A^1 THEN $x^1(k+1) = 1.2x(k) - 0.6x(k-1)$
P^2: IF $x(k)$ is A^2 THEN $x^2(k+1) = x(k) - 0.4x(k-1)$

where A^i are fuzzy sets similar to those in Figure 7.28. It is desired to check the stability of this system.

SOLUTION: The two linear subsystems' matrices are

$$\mathbf{A}_1 = \begin{bmatrix} 1.2 & -0.6 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 1 & -0.4 \\ 1 & 0 \end{bmatrix}$$

The product matrix $\mathbf{A}_1\mathbf{A}_2$ is

$$\mathbf{A}_1\mathbf{A}_2 = \begin{bmatrix} 0.6 & -0.48 \\ 1 & -0.4 \end{bmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

whose eigenvalues are $\lambda_{1,2} = 0.1 \pm j\,0.48$ which indicates that $\mathbf{A}_1\mathbf{A}_2$ is a stable matrix. Thus, by Theorem 7.3 a common $\mathbf{P}$ can be found. Now, by using the two-step algorithm, we have a $\mathbf{P}$ matrix

$$\mathbf{P} = \begin{bmatrix} 2 & -1.2 \\ -1.2 & 1 \end{bmatrix}$$

which would satisfy both equations $\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P} < 0$, $i = 1, 2$ simultaneously. Hence, the fuzzy system is globally asymptotically stable.

Stability of Nonautonomous Fuzzy Systems

Thus far, the criteria which have been presented treat autonomous (either closed-loop or no input) systems. Consider the following nonautonomous fuzzy system

$$\begin{aligned} P^i:\quad & \text{IF } x(k) \text{ is } A_1^i \text{ and ... and } x(k-n+1) \text{ is } A_n^i \text{ and } u(k) \text{ is } B_1^i \\ & \text{and ... and } u(k-m+1) \text{ is } B_m^i \\ & \text{THEN } x^i(k+1) = a_o^i + a_1^i\,x(k) + \ldots + a_n^i\,x(k-n+1) \\ & \quad + b_1^i\,u(k) + \ldots + b_m^i\,u(k-m+1) \end{aligned} \tag{7.3.9}$$

In this section, we use some results from Tahani and Sheikholeslam (1994) to test the stability of the above system. We begin with a definition.

Definition 7.2 The nonlinear system

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k]\,, \qquad \mathbf{y} = \mathbf{g}\,[\mathbf{x}(k), \mathbf{u}(k), k]$$

is *totally stable* if and only if for any bounded input $\mathbf{u}(k)$ and bounded initial state $\mathbf{x}_0$, the state $\mathbf{x}(k)$ and the output $\mathbf{y}(k)$ of the system are bounded, i.e., we have

$$\begin{aligned} &\text{For all } \|\mathbf{x}_0\| < \infty \text{ and for all } \|\mathbf{u}\| < \infty \Rightarrow \|\mathbf{x}(k)\| < \infty \\ &\quad \textit{and } \|\mathbf{y}(k)\| < \infty \end{aligned} \tag{7.3.10}$$

Now, consider the following theorem:

Theorem 7.4 *The fuzzy system (7.3.9) is totally stable if there exists a*

controlengineers.ir
@controlengineers

common positive definite matrix **P** *such that the following inequalities hold*

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < 0 \tag{7.3.11}$$

for $i = 1, \ldots, l$, and $\mathbf{A}_i$ *is defined by (7.3.3). The proof of this theorem can be found in Sheikholeslam (1994). Note that the conditions (7.3.11) are the same as those in (7.3.6) of Theorem 7.2 for fuzzy system (7.3.1).*

Example 7.3.3 Let us consider the following fuzzy system

P^1: IF $x(k)$ is A^1 THEN $x^1(k+1) = 0.85x(k) - 0.25x(k-1) + 0.35\,u(k)$
P^2: IF $x(k)$ is A^2 THEN $x^2(k+1) = 0.56x(k) - 0.25x(k-1) + 2.22\,u(k)$

It is designed to check if the system is totally stable. The fuzzy sets A^i, $i = 1, 2$ are shown in Figure 7.30. The input $u(k)$ is bounded.

SOLUTION: The $\mathbf{A}_i$, $i = 1, 2$ matrices are

$$\mathbf{A}_1 = \begin{bmatrix} 0.85 & -0.25 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0.56 & -0.25 \\ 1 & 0 \end{bmatrix}$$

If we choose the positive definite matrix **P**,

$$\mathbf{P} = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix}$$

then it can be easily verified that the system is totally stable. A simulation

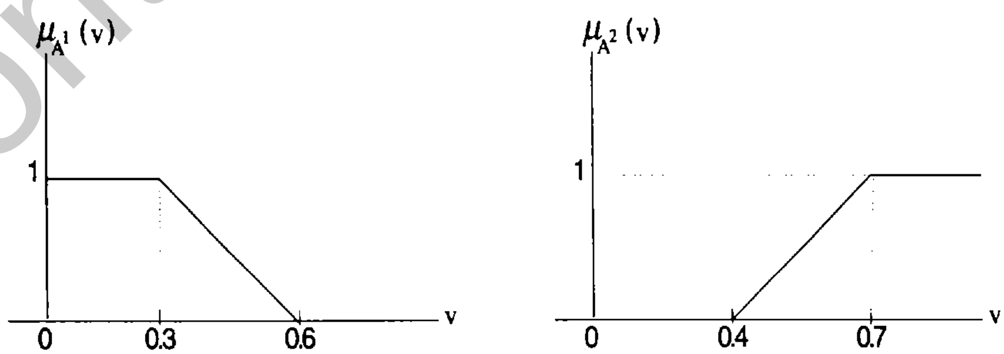


Figure 7.30 Fuzzy set membership functions for Example 7.3.3.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

study of the two subsystems and the fuzzy system revealed this fact. Note that the criterion on $\mathbf{A}_i\mathbf{A}_j$ matrices in Theorem 7.3 checks here as well. In fact, the eigenvalues of $\mathbf{A}_1\mathbf{A}_2$ are $\lambda\{\mathbf{A}_1\mathbf{A}_2\} = 0.012 \pm\ j\ 0.25$.

Example 7.3.4 Reconsider the implications of Example 7.2.9, repeated here again

P^1: IF $x(k)$ is A^1 THEN $x^1(k+1) = 1.85x(k) - 0.65x(k-1) + 0.35\ u(k)$

P^2: IF $x(k)$ is A^2 THEN $x^2(k+1) = 2.56x(k) - 0.135x(k-1) + 2.22\ u(k)$

C^1: IF $x(k)$ is A^1 THEN $f^1(k+1) = k_1^1 x(k) + k_2^1 x(k-1)$

C^2: IF $x(k)$ is A^2 THEN $f^2(k+1) = k_1^2 x(k) + k_2^2 x(k-1)$

where sets A^i are shown in Figure 7.30. It is desired to find appropriate feedback gains k_i^j, i, $j = 1,2$ such that the overall closed-loop fuzzy system would have closed loop poles at 0.9 and 0.7 within the unit circle.

SOLUTION: The closed-loop system's implications were already obtained in Example 7.2.9, i.e.,

S^{11}: IF $x(k)$ is (A^1 and A^1)
THEN $x^{11}(k+1) = (1.85 - 0.35k_1^1)\ x(k) + (-0.65 - 0.35k_2^1)\ x(k-1)$

$2S^{12*}$: IF $x(k)$ is (A^1 and A^2)
THEN $x^{12*}(k+1) = [2.205 - (2.22k_1^1 + 0.35k_1^2)/2]\ x(k)$
$+ (-0.3925 - (2.22k_2^1 + 0.35k_2^2)/2]\ x(k-1)$

S^{22}: IF $x(k)$ is (A^2 and A^2)
THEN $x^{22}(k+1) = (2.56 - 2.22k_1^2)\ x(k)$
$+ [-0.135 - 2.22k_2^2)\ x(k-1)$

where it is assumed that the external input $r(k) = 0$. Clearly, the above implications have four parameters between them and we only have two conditions. Thus, we have a nonunique solution to the problem. We proceed by finding parameters k_1^1 and k_1^2 for S^{11} so that $\lambda_1 = 0.7$ and $\lambda_2 = 0.9$. We will have $k_1^1 = 0.71$and $k_2^1 = -0.06$. Similarly, one can find k_1^2 and k_2^2 using S^{22} and the same two pole conditions. The feedback gains are $k_1^2 = 0.43$ and $k_2^2 = 0.22$. What is important at this point

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is to make sure that for this set of k_i^j values, the fuzzy subsystem S^{12*} is also stable. The characteristic equation of S^{12*} for this set of gains is $\lambda_2 - 1.39\lambda + 0.364 = 0$, leading to two eigenvalues $\lambda_1 = 0.96$ and $\lambda_2 = 0.38$ within the unit circle. One can, at this point, formulate closed-loop matrices $\mathbf{A}_{11}$, $\mathbf{A}_{12*}$, and $\mathbf{A}_{22}$ and check the stability of the overall system (see the problem section for a CAD problem for this). The matrices $\mathbf{A}_{11}\mathbf{A}_{12}$ and $\mathbf{A}_{11}\mathbf{A}_{22}$ are all stable, hence a common $\mathbf{P} > 0$ is possible.

The next section presents an alternative (non-Lyapunov) criterion for fuzzy control systems.

7.3.4 Fuzzy System Stability via Interval Matrix Method

Recent results on the stability of time-varying discrete interval matrices by Han and Lee (1994) can lead us to some more conservative, but computationally more convenient, stability criteria for fuzzy systems of the Takagi-Sugeno type shown by Equation (7.3.1). Before we can state these new criteria some preliminary discussions will be necessary.

Consider a linear discrete-time system described by a difference equation in state form:

$$\mathbf{x}(k+1) = (\mathbf{A} + \mathbf{G}(k))\,\mathbf{x}(k)\ , \qquad \mathbf{x}(0) = \mathbf{x}_0 \tag{7.3.12}$$

where $\mathbf{A}$ is an $n \times n$ constant asymptotically stable matrix, $\mathbf{x}$ is the $n \times 1$ state vector, and $\mathbf{G}(k)$ is an unknown $n \times n$ time-varying matrix on the perturbation matrix's maximum modulus, i.e.,

$$|\mathbf{G}(k)| \le \mathbf{G}_m\ , \quad \text{for all } k \tag{7.3.13}$$

where the $|\cdot|$ respresents the matrix with modulus elements and the inequality holds element-wise. Now, consider the following theorem.

Theorem 7.5 *The time-varying discrete-time system (7.3.12) is asymptotically stable if*

$$\rho\left(|\mathbf{A}| + \mathbf{G}_m\right) < 1 \tag{7.3.14}$$

where $\rho(\cdot)$ stands for spectral radius of the matrix. The proof of this theorem is straightforward, based on the evaluation of the spectral norm $\|x(k)\|$ of $x(k)$ and showing that if condition (7.3.14) holds, then $\lim_{k\to\infty}\|x(k)\| = 0$. The proof can be found in Han and Lee (1994).

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Definition 7.3 An interval matrix $\mathbf{A}_I(k)$ is an $n \times n$ matrix whose elements consist of intervals $[b_{ij}, c_{ij}]$ for $i, j = 1, \ldots, n$, i.e.,

$$\mathbf{A}_I(k) = \begin{bmatrix} [b_{11}, c_{11}] & \cdots & [b_{1n}, c_{1n}] \\ \vdots & & \vdots \\ & [b_{ij}, c_{ij}] & \cdots \\ [b_{n1}, c_{n1}] & \cdots & [b_{nn}, c_{nn}] \end{bmatrix} \tag{7.3.15}$$

Definition 7.4 The *center matrix*, $\mathbf{A}_c$ and the *maximum difference matrix*, $\mathbf{A}_m$ of $\mathbf{A}_I(k)$ in (7.3.15) are defined by

$$\mathbf{A}_c = \frac{\mathbf{B}+\mathbf{C}}{2}, \quad \mathbf{A}_m = \frac{\mathbf{C}-\mathbf{B}}{2} \tag{7.3.16}$$

where $\mathbf{B} = \{b_{ij}\}$ and $\mathbf{C} = \{c_{ij}\}$. Thus, the interval matrix $\mathbf{A}_I(k)$ in (7.3.15) can also be rewritten as

$$\mathbf{A}_I(k) = [\mathbf{A}_c - \mathbf{A}_m, \mathbf{A}_c + \mathbf{A}_m] = \mathbf{A}_c + \Delta\mathbf{A}(k) \tag{7.3.17}$$

with $|\Delta\mathbf{A}(\mathrm{k})| \le \mathbf{A}_m$.

Lemma 7.2 *The interval matrix $\mathbf{A}_I(k)$ is asymptotically stable if matrix $\mathbf{A}_c$ is stable and*

$$\rho\left(|\mathbf{A}_c| + \mathbf{A}_m\right) < 1 \tag{7.3.18}$$

or in canonical form,

$$\rho\left(|\mathbf{T}^{-1}\mathbf{A}_c\mathbf{T}| + |\mathbf{T}^{-1}|\mathbf{A}_m|\mathbf{T}|\right) < 1 \tag{7.3.19}$$

The proof can be found in Han and Lee (1994). The above lemma can be used to check the sufficient condition for the stability of fuzzy systems of Takagi-Sugeno type given in Equation (7.3.2). Consider a set of m fuzzy rules like (7.3.2),

$$\begin{aligned} &\text{IF } x(k) \text{ is } A_1^1 \text{ and} \ldots \text{ and } x(k-n+1) \text{ is } A_n^1 \\ &\qquad \text{THEN } \mathbf{x}^1(k+1) = \mathbf{A}_1\mathbf{x}(k) \\ &\vdots \\ &\text{IF } x(k) \text{ is } A_1^m \text{ and} \ldots \text{ and } x(k-n+1) \text{ is } A_n^m \\ &\qquad \text{THEN } \mathbf{x}^m(k+1) = \mathbf{A}_m\mathbf{x}(k) \end{aligned} \tag{7.3.20}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $\mathbf{A}_i$ matrices for $i = 1, \ldots, m$ are defined by (7.3.3). One can now formulate all the m matrices $\mathbf{A}_i$, $i = 1, \ldots, m$ as an interval matrix of the form (7.3.17) by simply finding the minimum and maximum of all the elements at the top row of all the $\mathbf{A}_i$ matrices. In other words, we have

$$\mathbf{A}_I(k) = \begin{bmatrix} [\underline{a}_1, \bar{a}_1] & \cdots & [\underline{a}_n, \bar{a}_n] \\ 1 \quad 0 & \cdots & 0 \\ 0 \quad \ddots & & \vdots \\ \vdots & & \\ 0 \quad \cdots & 1 & 0 \end{bmatrix} \tag{7.3.21}$$

where $\underline{a}_i$ and $\bar{a}_i$, for $i = 1, \ldots, n$ are the minimum and maximum of the respective elements of the first rows of $\mathbf{A}_i$ in (7.3.3), taken element by element.

Using the above definitions and observations, the fuzzy system (7.3.20) can be rewritten by

$$\begin{aligned} &\text{IF } x(k) \text{ is } \mathbf{A}_1^i \text{ and} \ldots \text{and } x(k-n+1) \text{ is } \mathbf{A}_n^i \\ &\text{THEN } \mathbf{x}(k+1) = \mathbf{A}_I^i \mathbf{x}(k) \end{aligned} \tag{7.3.22}$$

where $i = 1, \ldots, m$ and $A_I^i(k)$ is an interval matrix of form (7.3.21), except that $\underline{a}_i = \bar{a}_i = a_i$. Now, finding the weighted average equation (7.2.29), one has

$$\mathbf{x}(k+1) = \sum_{i=1}^{m} w^i A_I^i(k)\mathbf{x}(k) \Big/ \sum_{i=1}^{m} w^i \triangleq A_I(k)\mathbf{x}(k) \tag{7.3.23}$$

where

$$A_I(k) = \left[\sum_{i=1}^{m} w^i A_I^i(k)\right] \Big/ \sum_{i=1}^{m} w^i \tag{7.3.24}$$

is the overall interval matrix for the fuzzy system (7.3.20).

Theorem 7.6 *The fuzzy system (7.3.23) is asymptotically stable if the interval matrix $\mathbf{A}_I(k)$ in (7.3.24) is asymptotically stable, i.e., the conditions of Lemma 7.2 are satisfied.*

Example 7.3.5 Reconsider the fuzzy system of Example 7.3.1. It is

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

desired to check its stability.

SOLUTION: The system's two canonical matrices are written in the form of an interval matrix (7.3.21) as

$$\mathbf{A}_I(k) = \begin{pmatrix} [-1,1] & -0.2 \\ 1 & 0 \end{pmatrix}$$

The center and maximum difference matrices (see Equation (7.3.16)) are

$$\mathbf{A}_c = \begin{pmatrix} 0 & -0.2 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{A}_m = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Then, condition (7.3.18) would become,

$$\rho\left(\left|\mathbf{A}_c\right| + \mathbf{A}_m\right) = \rho\begin{pmatrix} 1 & 0.2 \\ 1 & 0 \end{pmatrix} = 1.17 > 1$$

Thus, the stability of the fuzzy system under consideration is inconclusive. In fact, it was shown to be unstable.

Example 7.3.6 Consider a two-rule fuzzy system

IF $x(k)$ is A_1 THEN $x^1(k+1) = 0.3x(k) + 0.5\ x(k-1)$
IF $x(k)$ is A_2 THEN $x^2(k+1) = 0.2\ x(k) + 0.2\ x(k-1)$

where $\mathbf{A}_i$, $i = 1, 2$ are the piecewise continuous fuzzy sets in Figure 7.30. It is desired to check its stability by both Lyapunov and interval matrix methods.

SOLUTION: The $\mathbf{A}_i$ matrices are

$$\mathbf{A}_1 = \begin{pmatrix} 0.3 & 0.5 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{A}_2 = \begin{pmatrix} 0.2 & 0.2 \\ 1 & 0 \end{pmatrix}$$

Using Theorem 7.2, a common $\mathbf{P} > 0$ matrix

$$\mathbf{P} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

would result in

$$\lambda\left(\mathbf{A}_1^T\mathbf{P}\mathbf{A}_1 - \mathbf{P}\right) = \{-0.037, -1.88\}$$
$$\lambda\left(\mathbf{A}_2^T\mathbf{P}\mathbf{A}_2 - \mathbf{P}\right) = \{-0.218, -2.02\}$$

which indicates that the system is asymptotically stable.

Now, consider the center and the maximum difference matrices for this system

$$\mathbf{A}_c = \begin{pmatrix} 0.25 & 0.35 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{A}_m = \begin{pmatrix} 0.05 & 0.15 \\ 0 & 0 \end{pmatrix}$$

and matrix

$$\left|\mathbf{A}_c\right| + \mathbf{A}_m = \begin{pmatrix} 0.3 & 0.5 \\ 1 & 0 \end{pmatrix}$$

and $\rho\left(\left|\mathbf{A}_c\right| + \mathbf{A}_m\right) = 0.873 < 1$. Hence, in view of the above inequality and the fact that $\mathbf{A}_c$ is stable, by Theorem 7.6 the interval matrix

$$\mathbf{A}_I(k) = \begin{pmatrix} [0.2, 0.3] & [0.2, 0.5] \\ 1 & 0 \end{pmatrix}$$

is asymptotically stable and the fuzzy system is stable.

In this section a number of sufficiency conditions have been presented to check for the asymptotic stability of fuzzy control systems with Takagi-Sugeno type rules, i.e., Equation (7.3.1). All the criterion presented here are somewhat conservative. It is noted that if a given condition, say Equations (7.3.6, (7.3.11), or (7.3.18) is not satisfied, it does not mean that the system is necessarily unstable. On the other hand, if the condition is true, then the system is, in fact, stable. It is hoped that the content of this section will serve as a starting point for many new results that will lead toward a solid stability theory for fuzzy control systems. This challenge still exists for both control engineers and mathematicians.

Problems

7.1. Consider a set of two fuzzy rules given by

IF x is A_1 and y is B_1 THEN z is C_1

IF x is A_2 and y is B_2 THEN z is C_2

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where A_i, B_i, and C_i, $i = 1, 2$ are trapezoidal functions described below:

$$\mu_{A_1}(x)=\begin{cases} x+3 & -3\le x\le -2 \\ 1 & -2\le x\le 0 \\ 1-x & 0\le x\le 1 \end{cases} \qquad \mu_{B_2}(y)=\begin{cases} \dfrac{y-3}{2} & 3\le y\le 5 \\ 1 & 1\le y\le 6 \\ 0 & y\ge 6 \end{cases}$$

$$\mu_{A_2}(x)=\begin{cases} x-2 & 2\le x\le 3 \\ 1 & 3\le x\le 4 \\ 5-x & 4\le x\le 5 \end{cases} \qquad \mu_{C_1}(z)=\begin{cases} z-5 & 5\le z\le 6 \\ 1 & 6\le z\le 8 \\ 9-z & 8\le z\le 9 \end{cases}$$

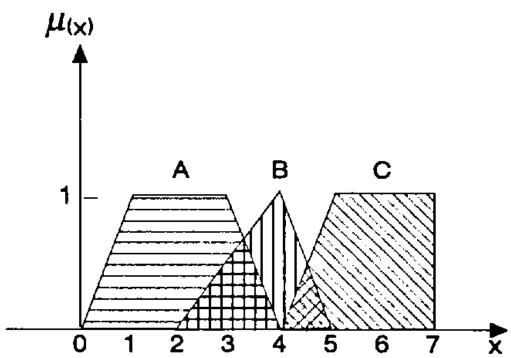
$$\mu_{B_1}(y)=\begin{cases} y & 0\le y\le 1 \\ 1 & 1\le y\le 3 \\ 4-y & 3\le y\le 4 \end{cases} \qquad \mu_{C_2}(z)=\begin{cases} \dfrac{z-7}{2} & 7\le z\le 9 \\ 1 & 9\le z\le 10 \\ 0 & z>10 \end{cases}$$

If the incoming values are $x_s = 0.5$ and $y_s = 2.25$, find z using the maximum defuzzification method.

7.2. Repeat Problem 7.1 using the center of gravity method and $(x_s, y_s) = (2.5, 4.2)$.

7.3. Repeat Example 7.2.1 using the MOM method for defuzzification.

7.4. Given three output fuzzy sets shown below.



Find the center of gravity and the MOM.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

7.5. Repeat Example 7.2.2 for the following two rules:

IF x is A_1 and y is B_1 THEN $z = \sin(x + y)$
IF x is A_2 and y is B_2 THEN $z = \cos(x + y)$

7.6. CADLAB For the inverted pendulum of Equation (7.2.21), use your favorite computer program to place the poles at −2, -2, -5, -5. Use these nominal values of k in $u = -kx$ and assess the effects of k_2 and k_4 on the system's overshoot by further simulations.

7.7. Derive the relations for k_P, k_D, and k_I for both the PD and PID controllers of Example 7.2.4.

7.8. Devise three Takagi-Sugeno type rules to tune the parameters k_P, k_I, and k_D for Example 7.2.4.

7.9. CADLAB Use FULDEK to simulate the 4th-order system

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u$$

using three arbitrary linguistic rules and triangular membership functions for $x = [\ 0,\ 10\ ]$, $u = [\ -5,\ 5\ ]$, and $x(0) = (1\ \ 0\ \ 0\ \ 0)^T$.

7.10. CADLAB Repeat problem 7.6 for the 2nd-order model of the inverted pendulum (7.2.22) to place the poles at −2 and −4.

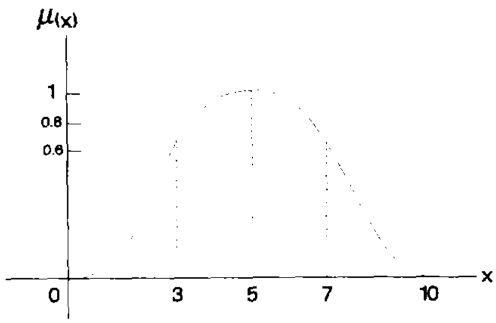
7.11. CADLAB Use FULDEK or a similar program to simulate both a conventional and a fuzzy controller for the 2nd-order inverted pendulum problem in Equation (7.2.22) *simultaneously*.
Hint: Augment the model twice into two 2nd-order systems and use fuzzy rules for the lower half of the system only.

7.12. For the inverted pendulum problem compose five rules to eliminate the overshoot in terms of the approach of Section 7.2.6.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

7.13. Consider the fuzzy sets of Problem 7.4. Check if they are continuous-piecewise polynomial type.

7.14. Consider a Gaussian membership function



Approximate the function by four line segments and check if it is PCP type.

7.15. Consider the fuzzy sets A, B, and C of Problem 7.4. Find the Cartesian products A × B and B × C.

7.16. Repeat Problem 7.15 for the product A × B × C.

7.17. Consider a parallel connection of two fuzzy blocks shown below. Let these blocks be represented by Equation (7.2.31) for $i = 1$ and 2. Then the rules for p^{ij} are given by

P^{12}: IF $\underset{\sim}{x}(k)$ is (A^1 and A^2) and

$\underset{\sim}{y}(k)$ is (B^1 and B^2) THEN

$$x^{12}(k+1) = \left(a_0^1 + a_0^2\right) + \sum_{p=1}^{2}\left(a_p^1 + a_p^2\right)$$

$$x(k-p+1) + \sum_{q=1}^{2}\left(b_q^1 + b_q^2\right)u(k-q+1)$$

Now, using P^i, $i = 1, 2$ from Example 7.2.9, find P^{12} block representation.

7.18. CADLAB A fuzzy system is represented by

P^1: IF $x(k-1)$ is A_1 THEN $x^1(k+1) = 2x(k) - 0.5x(k-1)$

P^2: IF $x(k-1)$ is A_2 THEN $x^2(k+1) = -x(k) - 0.5\,x(k-1)$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where A_i, $i = 1, 2$ are shown in Figure 7.28. Check if the system is stable by Lyapunov's method. Use $x(0) = -1$ and $x(1) = 0$ and verify your answer by simulation.

7.19. If the rules in Example 7.3.3 are changed to

P^1: IF $x(k)$ is A^1 THEN $x^1(k+1) = 0.15x(k-1) + 0.25u(k)$

P^2: IF $x(k)$ is A^2 THEN $x^2(k+1) = 0.25x(k) - 0.25x(k-1) + u(k)$

would the system be stable?

7.20. Repeat Example 7.3.4 for the following

P^i rules:

P^1: If $x(k)$ is A^1 THEN $x^1(k+1) = 0.8x(k) + 0.5x(k-1) + 0.25u(k)$

P^2: If $x(k)$ is A^2 THEN $x^2(k+1) = 1.5x(k) - 0.2x(k-1) + 0.2u(k)$

7.21. Repeat Problem 7.18 using the interval matrix method.

7.22. Repeat Problem 7.20 using the interval matrix method.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Chapter 8

Fuzzy Control Systems
Adaptation and Hierachy

controlengineers.ir

8.1 Introduction

The last chapter dealt with the basic architectures of fuzzy control systems and one of the most important issues concerning them—*stability*. The current state of fuzzy control systems calls for two additional important issues—one is the notion of *adaptation* and the other is that of *hierarchy*.

Briefly, in dealing with the inexact methods of creating fuzzy rules and the arbitrary shaping of the membership function, the designer would almost always confront potential variations of one, the other, or both. Therefore, in Section 8.2, means for "adapting" the structure of the fuzzy control to accommodate plant uncertainties, unmodeled dynamics, disturbances, and the like, are sometimes very critical. This section will attempt to review some adaptive schemes for fuzzy control systems.

The other issue discussed in this chapter is the notion of hierarchy within the context of fuzzy control systems. The applications of fuzzy logic and fuzzy control, thus far, have been to low-order or small-scale systems such as an elevator, washing machine, etc. The challenge remains here to handle large-scale systems through fuzzy control. Handling a large-scale fuzzy control system could potentially require an infinite number of fuzzy rules.

The notion of hierarchy, as discussed in this chapter, covers a lot of ground. On one hand, hierarchical ordering of fuzzy rules is used to reduce the size of the inference engine. On the other hand, real-time implementation of fuzzy controllers can help reduce the burden of large-sized rule sets by *fusing* sensory data before inputing the system's output to the inference engine. Thus, a concerted effort has been made, in Section 8.3, to formally reduce the size of the fuzzy rule base to make fuzzy control attractive for large-scale systems. The notions of dynamic hier-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

archy (Chapter 4) and decentralization (Chapter 5) can also be integrated with fuzzy logic and some insights and open problems on "fuzzy control via interaction balance" and "decentralized fuzzy control" are presented in Section 8.3.3, under the title of "hybrid" techniques, i.e., incorporating fuzzy control with one or more other schemes. Section 8.3 and this final chapter will end by briefly treating three hybrid or "intelligent control" problems. These are the hybridization of fuzzy logic (IF-THEN rules) and evolutionary algorithms and neural networks. Schemes considered are fuzzy-genetic algorithms and fuzzy-genetic programming in addition to the fuzzy neural network approaches to control.

8.2 Adaptive Fuzzy Control Systems

Two of the difficulties with the design of any fuzzy control system are the shape of the membership functions and the choice of the fuzzy rules. In fact, one may call these needs a "black art." One of the approaches that a great many researchers advocate is to use equally spaced triangular membership functions, "common sense" rules, and then find a way to "modify," "tune," or "adapt" them to the plant variations, unmodeled dynamics, or external disturbances. The initial definition of an *adaptive fuzzy control system* in this book is a fairly general one. Later on a few more technically motivated classifications of adaptive fuzzy control systems will also be given.

A control system is said to be an adaptive fuzzy control system if either a set of fuzzy rules are used to modify or tune an existing crisp controller (see Figure 7.4) or some algorithm is used to modify or change an existing fuzzy controller's architecture, i.e., membership functions and/or rules. Excluded from these general definitions are those fuzzy control systems whose architecture is being altered through neural networks and/or evolutionary algorithms, e.g., genetic algorithms. These types of integrations are referred to as "hybrid intelligent systems" which are not treated here.

In the current section, three ways of designing adaptive fuzzy control systems will be discussed. These are *parameter estimation*, *multiterm controllers*, and *indirect*. The body of literature on adaptive fuzzy control systems is already so immense that there is at least one book on the subject by Wang (1994a). Several other authors (Sugeno and Murakami, 1985; Yin and Lee, 1995; Yamakawa and Furukawa, 1992; Nomura *et al.*, 1992; Jang, 1993; Wang and Vachtsenavos, 1992; Fei and Isik, 1992; Bonarini *et al.*, 1994; Zhang and Edmunds, 1992; Alang-Rashid, 1992; Alang-Rashid and Heger, 1992a,b) have also taken up the subject, some of which will be reviewed in this section.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

8.2.1 *Adaptation by Parameter Estimation*

In Section 7.2 and Figure 7.1, one definition of fuzzy control systems was given. This type of fuzzy controller is what some authors call a *TYPE-1 Fuzzy Controller*, i.e., those which try to mimic a human expert operator. In such a case, the rule base requires little modification, and the adaptation will be concentrated around the parameters associated with the membership functions. Let a fuzzy control system consist of M inputs and one output. Assume that the ith input has n_i linguistic labels (fuzzy sets) and the output has N labels. If the membership function of the primary fuzzy sets are specified by m parameters, then the total number of adjustable parameters for both inputs, output, and the overall system are

$$P_i = m \cdot \sum_{i=1}^{M} n_i \text{ , } P_o = mN \text{ , and } P_s = P_i + P_o = m\left(N + \sum_{i=1}^{M} n_i\right)$$

respectively (Alang-Rashid, 1992). As an example, for the simple second-order inverted pendulum problem and the use of a triangular membership function, using both inputs and output with $m = 3$ adjustable parameters each with $n_i = N = 5$ labels, we would have $P_s = 3\ (5 + 5) + 3(5) = 45$ parameters to be adjusted. To avoid excessive numbers of adjustable parameters, Alang-Rashid (1992) has proposed modifying the parameters of the output membership functions, thereby reducing the total number of adjustable parameters to only N.

Let the class of membership functions considered here be Gaussian as shown in Figure 8.1. The centroid of each primary fuzzy set, except endpoint sets (see Figure 8.1), is equal to the value corresponding to the maximum membership value $\mu(c) = 1$. The centroid of the endpoint membership functions is the value of a variable such that the area under the membership function curve to its left equals its right, as shown in Figure 8.1, i.e., find c such that

$$\int_{x_L}^{c} \mu(x)dx = \int_{c}^{x_R} \mu(x)dx \tag{8.2.1}$$

where $x \in [x_L\ ,\ x_R]$.

Let us consider the centroid defuzzification equation (7.2.16), rewritten below as,

$$u_c(k) = \left(\frac{\sum_{i=1}^{N} \mu_{R_i} c_i}{\sum_{i=1}^{N} \mu_{R_i}} \right)_k \tag{8.2.2}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

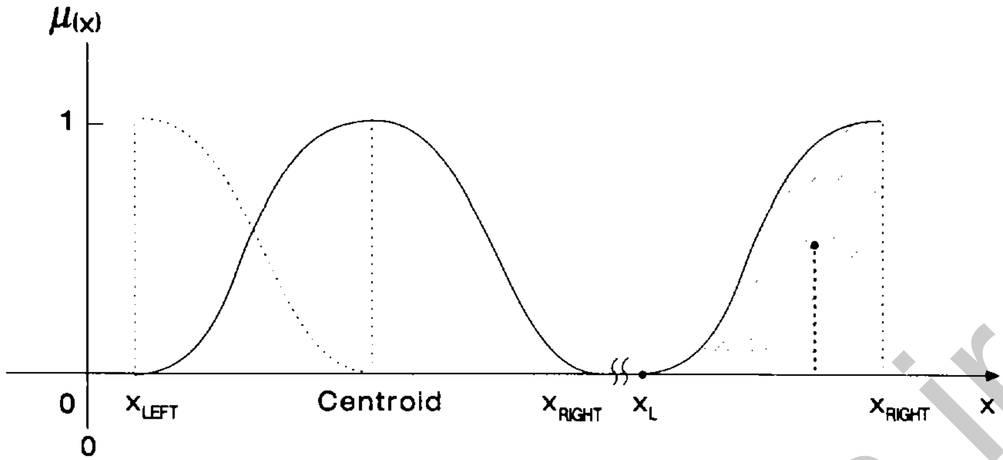


Figure 8.1 Gaussian membership functions.

where $u_c(k)$ is the crisp (defuzzified) output value at time k, μ_{R_i} is the membership function resulting from the ith rule strength.

In the parameter estimation approach considered here, c_i, $i = 1, 2, ..., N$ are the unknown variables, where N is the number of output variables' primary fuzzy sets. One may view Equation (8.2.2) as a mathematical model of a multiinput, single output system depicted in Figure 8.2. The output of the system is u_c, and its inputs are $\mu_{R_i} \Big/ \sum_{i=1}^{N} \mu_{R_i}$, where N is the number of inputs. The principal aim of the tuning process is to determine the "system states" c_i by making use of the system input-output data.

By setting up the fuzzy controller output in this way, it is clear that the adaptive control or tuning problem can also be posed as a system identification problem. System identification consists of model structure, and model parameter estimation. We have the first part, which is in the form of Equation (8.2.2). We devote this section to the second part, for which three methods have been proposed by Alang-Rashid (1992).

The fuzzy controller output equation of Method 1 is cast as a set of algebraic equations that are solved twice (Alang-Rashid and Heger, 1992a): first, to capture the numerical interpretation of the rule base, and second, to modify the output variable centroids. We call this method a substitution method because in finding the new centroids, the FLC output is replaced with the desired output. Method 2 uses a recursive least-squares technique to change the centroids' values at every sampling instant (Alang-Rashid and Heger, 1992b). Method 3 updates the centroid values

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

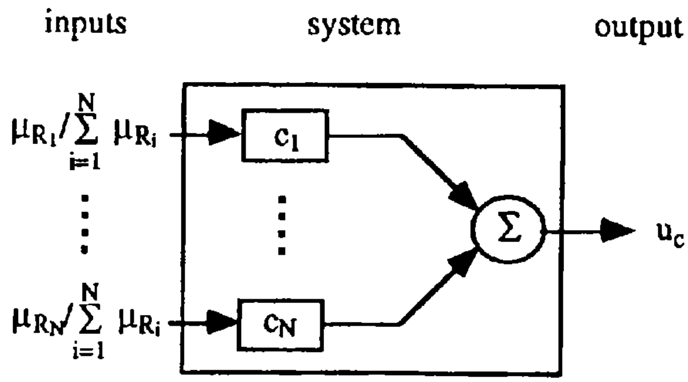


Figure 8.2 "System" representation of the FLC output.

by an amount proportional to the gradient of the error between the desired and the actual outputs. Each of these methods is described in what follows. Applications of the method, using control problems of known solution, are also given. The aim is to demonstrate the methods and to check their applicability. The next three sections describe each of these methods.

Method 1—Substitution Approach For a series of data points sampled at time intervals $k = 1, 2, \ldots, p$, the fuzzy controller's output, Equation (8.2.2), can be written as

$$u_c(k) = a_1(k)c_1 + a_2(k)c_2 + \ldots + a_N(k)c_N \tag{8.2.3}$$

where

$$a_i(k) = \left(\frac{\mu_{R_i}}{\sum_{i=1}^{N} \mu_{R_i}} \right)_k \tag{8.2.4}$$

and $i = 1, 2, \ldots, N$. Equation (8.2.3) gives a set of p simultaneous equations that can be written in matrix form as

$$\mathbf{u_c} = \mathbf{A}\,\mathbf{c} \tag{8.2.5}$$

where,

$$\mathbf{u}_\mathbf{c}^T = \left[u_c(k)\ u_c(k+1) \ldots u_c(p)\right]$$

$$\mathbf{c}^T = [c_1\ \ c_2 \ldots c_N]$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and $\mathbf{A}$ is a $(p \times N)$ matrix. The superscript T denotes matrix transpositions. The elements of the matrix $\mathbf{A}$ are $\mathbf{a}_{ik} = a_i(k)$, $i = 1, 2, \ldots, N$, and $k = 1, 2, \ldots, p$. For simplicity, for now, let $p = N$; then, $\mathbf{A}$ is a square matrix.

For the same operating conditions, and at the same sampling points, $k = 1, 2, \ldots, p$, let the desired fuzzy controller outputs be $u_d(k)$, i.e.,

$$\mathbf{u}_d^T = \left[u_d(k)\; u_d(k+1) \ldots u_d(p)\right]$$

For the fuzzy controller to produce these desired outputs, without changing the elements of the matrix $\mathbf{A}$ (the rule base), the centroid vector, $\mathbf{c}$ in Equation (8.2.5), must be adjusted. Let $\mathbf{c}^*$ be the new centroid vector such that $u_c(k) \rightarrow u_d(k)$ at every $k = 1, 2, \ldots, p$, i.e.,

$$\mathbf{c}^{*T} = \left[c_1^*\; c_2^* \ldots c_N^*\right]$$

Then, from Equation (8.2.5)

$$\mathbf{u}_d = \mathbf{A}\;\; \mathbf{c}^* \tag{8.2.6}$$

Provided that the matrix $\mathbf{A}$ is nonsingular, the new centroid vector can be obtained by solving

$$\mathbf{c}^* = \mathbf{A}^{-1}\, \mathbf{u}_d \tag{8.2.7}$$

For the cases of $p > N$, i.e., an overdetermined system, the pseudo inverse (see Section 2.2.1) of the matrix $\mathbf{A}$ can be used in Equation (8.2.7), thus,

$$\mathbf{c}^* = \mathbf{A}^{+}\, \mathbf{u}_d \tag{8.2.8}$$

where $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

Example 8.2.1 Consider a simple single-input, single-output system described by $y = 10 \sin(10\pi x)$, in which y is the output and x is the input. Let us limit the range of the output to the positive half period, i.e., [0, 10], only. Therefore, the possible values of the input are [0, 0.1]. It should be noted that this example is more of a fuzzy relation (see Appendix A) than a fuzzy controller with adaptive parameters. It is used here as an identification problem (see also Example 2.6.1).

SOLUTION: Let the input and output primary fuzzy sets be {L, ML, M, MH, H} and {L, M, H}, respectively. The primary fuzzy set L denotes Low, ML = Medium Low, M = Medium, MH = Medium High, and H =

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 8.1 Membership Function Parameters of x and y

variable	$\tilde{x}$	$\bar{x}$	σ
input, x	L	0	0.005
	ML	0.02	0.005
	M	0.05	0.005
	MH	0.08	0.005
	H	0.10	0.005
output, y	L	0	10
	M	0.5	10
	H	1	10

High. Let the membership functions be Gaussian-like. The general form of this membership function is $\mu_x = \exp\left(-2(x-\bar{x})^2\right)/\sigma$ where $\bar{x}$ is the median value of x. The membership function parameters are shown in Table 8.1. From this table, the output variable centroid vector is $\mathbf{c}^T = [1.1 \;\; 5.0 \;\; 8.9]$.

Let the rules relating x and y be R1: (L or H $\Rightarrow$ L); R2: (ML or MH $\Rightarrow$ M): and R3: (M $\Rightarrow$ H). We use this notation for clarity. For example, the rule (ML $\Rightarrow$ M) means: IF x is ML, THEN y is M, etc.

From Equation 8.2.3, an input value of $x = 0.01$ gives an output of u_c = ((0.39 x 1.1) + (0.39 x 5) + (0.21 x 8.9))/0.99 = 4.29. The desired value of the output, y_d, as given by $y = 10 \sin(10\pi x)$, is 3.09.

If we take a series of inputs, x = (0.01, 0.02, 0.03, 0.04, 0.05), a matrix **A** of

$$\mathbf{A} = \begin{bmatrix} .39 & .39 & .21 \\ .33 & .39 & .28 \\ .28 & .38 & .33 \\ .23 & .36 & .41 \\ .18 & .34 & .48 \end{bmatrix}$$

is obtained. Further, application of Equation (8.2.5),

$$\mathbf{y} = \begin{bmatrix} .39 & .39 & .21 \\ .33 & .39 & .28 \\ .28 & .38 & .33 \\ .23 & .36 & .41 \\ .18 & .34 & .48 \end{bmatrix} \begin{bmatrix} 1.1 \\ 5.0 \\ 8.9 \end{bmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

gives the FLC output of $\mathbf{y} = (4.25, 4.72, 5.15, 5.70, 6.17)$. The actual, or desired, output is given by solving $y = 10 \sin(10\pi x)$; thus, $y_d = (3.09, 5.88, 8.09, 9.51, 10.0)$. Therefore, the centroid vector must be corrected. By application of Equation (8.2.8), we obtain the corrected centroids to be $c_1^* = -53.01$, $c_2^* = 62.78$, and $c_3^* = -3.64$.

The actual and fuzzy (with original and corrected centroids) solutions of $y = 10 \sin(10\pi x)$ are plotted in Figure 8.3. We observe that the corrected centroids work very well in this case.

The main uneasiness that arises in this method, however, is that the corrected centroids, $\mathbf{c}^*$, are shifted very far from their original locations. The new centroids also were placed outside the domain of the output variable. According to the inference rules, c_1 represents the centroid of the primary fuzzy set Low, c_2 Medium, and c_3 High. The new centroid vector, however, implies that Low < Medium which is acceptable, and High < Medium which is counter to logic.

Method 2—Recursive Least-Squares Approach This method uses a recursive least-squares technique to minimize the sum of the squares of the error between the actual and the desired fuzzy controller outputs. The minimization is done in the least-squares sense by recursively updating the estimates of the values of the centroids.

Let $e(k)$ be the error between the actual and the desired fuzzy controller outputs at any time step, $k = 1, 2, \ldots, p$, i.e.,

$$e(k) = u_c(k) - u_d(k) \tag{8.2.9}$$

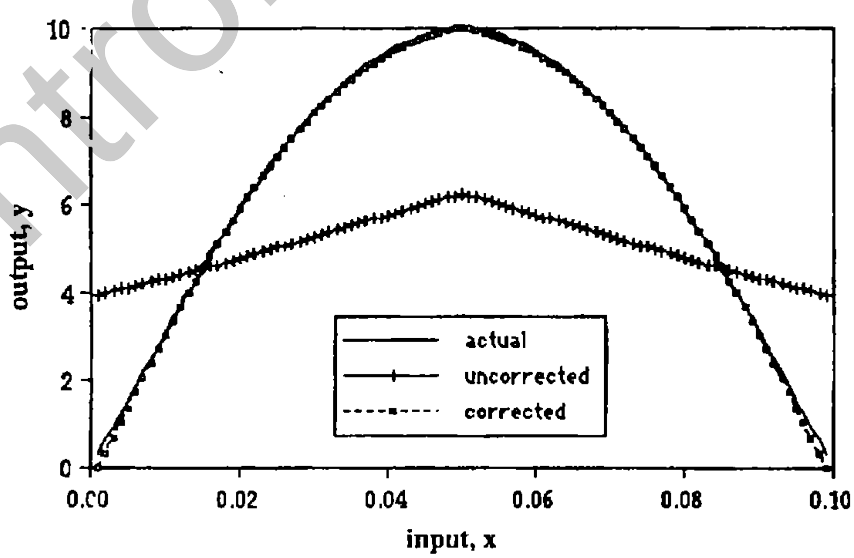


Figure 8.3 Actual, fuzzy uncorrected, and fuzzy corrected solutions.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Instead of vector-matrix form, as was done in Method 1, one can rewrite Equation (8.2.3) in vector notation: $u_c(k) = \mathbf{a}^T(k)\mathbf{c}$, in which $\mathbf{a}^T(k) = [a_1(k)\ a_2(k) \ldots a_n(k)]$, $k = 1, 2, \ldots, p$, and the vector $\mathbf{c}$ is defined through Equation (8.2.5). In this case, each vector represents the fuzzy controller output at time step k. By similar notation, the desired controller output can be written as $u_d(k) = \mathbf{a}^T(k)\,\mathbf{c}^*$, where the vector $\mathbf{c}^*$ represents the adjusted centroid vector as defined by Equation (8.2.6). Therefore, from Equation (8.2.9),

$$u_c(k) = \mathbf{a}^T(k)\mathbf{c} + e(k)$$

Let the error to be minimized, i.e., the objective function, be given by

$$\begin{aligned} E &= \sum_{k=1}^{p}\big(u_c(k) - u_d(k)\big)^2 \\ &= \sum_{k=1}^{p}\big(\mathbf{a}^T(k)\mathbf{c} - u_d(k)\big)^2 \end{aligned} \tag{8.2.10}$$

Let $\mathbf{a} \in \boldsymbol{R}^{N\times 1}$ and $\mathbf{b} \in \boldsymbol{R}^{N\times 1}$. Then $\mathbf{a}^T\mathbf{b} = a_1b_1 + a_1b_1 + \ldots + a_Nb_N$, and $\partial(\mathbf{a}^T\mathbf{b})/\partial b_k = a_k$ for every $k = 1, 2, \ldots, p$. Vectorially, then, $\partial(\mathbf{a}^T\mathbf{b})/\partial\mathbf{b} = \mathbf{a}$; and equivalently, $\partial(\mathbf{a}^T\mathbf{b})/\partial\mathbf{b}^T = \mathbf{a}^T$. Expansion of Equation (8.2.10) gives

$$E = \sum_{k=1}^{p}\Big(\big(\mathbf{a}^T(k)\mathbf{c}\big)^2 - \big(2\mathbf{a}^T(k)\mathbf{c}u_d(k)\big) + \big(u_d(k)\big)^2\Big)$$

Applying the result of vector differentiation, the above relation gives

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{c}} &= 2\sum_{k=1}^{p}\big(\mathbf{a}^T(k)\mathbf{c}\mathbf{a}(k) - \mathbf{a}(k)u_d(k)\big) \\ &= 2\sum_{k=1}^{p}\big(\mathbf{a}^T(k)\mathbf{a}(k)\mathbf{c} - \mathbf{a}(k)u_d(k)\big) \end{aligned}$$

By letting $\partial E/\partial c_i = 0$, i = 1, 2, …, N, we have

$$\left(\sum_{k=1}^{p}\big(\mathbf{a}(k)\mathbf{a}^T(k)\big)\right)\mathbf{c} - \sum_{k=1}^{p}\big(\mathbf{a}(k)u_d(k)\big) = 0$$

controlengineers.ir

controlengineers.ir
@controlengineers

Solving for the optimum centroid vector, **c**, we obtain

$$\mathbf{c}^* = \left(\sum_{k=1}^{p} \left(\mathbf{a}(k)\mathbf{a}^T(k) \right) \right)^{-1} \left(\sum_{k=1}^{p} \left(\mathbf{a}(k)u_d(k) \right) \right) \tag{8.2.11}$$

Equation (8.2.11) gives the centroid vector obtained at the pth sampling time. To recursively update the centroid vector, let us simplify Equation (8.2.11) by introducing the notations,

$$\mathbf{H}(p) = \left(\sum_{k=1}^{p} \left(\mathbf{a}(k)\mathbf{a}^T(k) \right) \right)^{-1} \tag{8.2.12}$$

and

$$\mathbf{h}(p) = \sum_{k=1}^{p} \left(\mathbf{a}(k)u_d(k) \right) \tag{8.2.13}$$

The matrix $\mathbf{H}(p)$ is of dimension $(N \times N)$, and $\mathbf{h}(p)$ is a $(N \times 1)$ vector. Equations (8.2.12) and (8.2.13) can be written as

$$\begin{aligned} \mathbf{H}^{-1}(p) &= \mathbf{a}(1)\mathbf{a}^T(1) + \cdots + \mathbf{a}(p-1)\mathbf{a}^T(p-1) + \mathbf{a}(p)\mathbf{a}^T(p) \\ &= \sum_{k=1}^{p-1} \left(\mathbf{a}(k)\mathbf{a}^T(k) \right) + \mathbf{a}(p)\mathbf{a}^T(p) \end{aligned}$$

Therefore, the recursive form of Equation (8.2.12) is $\mathbf{H}^{-1}(p) = \mathbf{H}(p-1) + \mathbf{a}(p)\mathbf{a}^T(p)$. Similarly, Equation (8.2.13) can be written as

$$\begin{aligned} \mathrm{h}(p) &= \mathrm{a}(1)u_d(1) + . \ . \ . + \mathrm{a}(p-1)u_d(p-1) + \mathrm{a}(p)u_d(p) \\ &= \sum_{k=1}^{p-1} \left(\mathrm{a}(k)u_d(k) \right) + a(p)u_d(p) \end{aligned}$$

Therefore, the recursive form of Equation (8.2.13) is given by

$$\mathbf{H}^{-1}(p) = \mathbf{H}^{-1}(p-1) + \mathbf{a}(p)\ \mathbf{a}^T(p) \tag{8.2.14}$$

$$\mathbf{h}(p) = \mathbf{h}(p-1) + \mathbf{a}(p)\ u_d(p) \tag{8.2.15}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Pre-multiply Equation (8.2.14) by $\mathbf{H}(p)$, and post-multiply the resulting equation by $\mathbf{H}(p-1)$ to obtain

$$\mathbf{H}(p-1) = \mathbf{H}(p) + \mathbf{H}(p)\mathbf{a}(p)\mathbf{a}^T(p)\mathbf{H}(p-1) \tag{8.2.16}$$

Post-multiplying Equation (8.2.16) by $\mathbf{a}(p)$ gives,

$$\mathbf{H}(p-1)\mathbf{a}(p) = \mathbf{H}(p)\mathbf{a}(p)\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)$$

Post-multiplying the above equation by $\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)^{-1}\mathbf{a}^T(p)\cdot$ $\mathbf{H}(p-1)$ gives,

$$\begin{aligned}&\mathbf{H}(p-1)\mathbf{a}(p)\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)^{-1}\mathbf{a}^T(p)\mathbf{H}(p-1)\\ &\quad = \mathbf{H}(p)\mathbf{a}(p)\mathbf{a}^T(p)\mathbf{H}(p-1)\end{aligned}$$

By making use of Equation (8.2.16) in the right-hand side of the above equation, we obtain

$$\begin{aligned}&\mathbf{H}(p-1)\mathbf{a}(p)\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)^{-1}\mathbf{a}^T(p)\mathbf{H}(p-1)\\ &\quad = \mathbf{H}(p-1) - \mathbf{H}(p)\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{H}(p) = \mathbf{H}(p-1) - \mathbf{H}(p-1)\mathbf{a}(p)\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)^{-1}\cdot\\ \mathbf{a}^T(p)\mathbf{H}(p-1)\end{aligned} \tag{8.2.17}$$

From Equation (8.2.11) through Equation (8.2.13), we obtain $\mathbf{c}^*(p) = \mathbf{H}(p)\mathbf{h}(p)$. By making substitutions of Equation (8.2.15) and Equation (8.2.17), $\mathbf{c}^*(p)$ can be written as

$$\begin{aligned}\mathbf{c}^*(p) = \Big(&\mathbf{H}(p-1) - \mathbf{H}(p-1)\mathbf{a}(p)\left(1+\mathbf{a}^T(p)\mathbf{H}(p-1)\mathbf{a}(p)\right)^{-1}\cdot\\ &\mathbf{a}^T(p)\ \mathbf{H}(p-1)\Big)\left(\mathbf{h}(p-1)+\mathbf{a}(p)u_d(p)\right)\end{aligned}$$

By generalizing this result to any time step k, we have the following equation:

$$\mathbf{c}^*(k) = \mathbf{c}^*(k-1) - \mathbf{H}(k)\mathbf{a}(k)\left(\mathbf{a}^T(k)\mathbf{c}^*(k-1) - u_d(k)\right) \tag{8.2.18}$$

This equation shows that the centroid vector is updated following the time

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

variation of the plant dynamic, while keeping the centroid vector optimum in the sense of the objective function shown in Equation (8.2.10). A starting value for the centroid vector can be set to correspond to the designed value of the centroid vector.

Example 8.2.2 To illustrate the second method, we use a control problem in which its solution is known. This problem is the control of fluid level in a holding tank (Slotine and Li, 1991). The control objective is to maintain the level of fluid in the tank, $h(t)$, so that it is close to a desired level $h_d(t)$. It is desired to apply Method 2 of the parameter estimation method.

SOLUTION: The system dynamic is given by Equation (8.2.19), where g = 9.81 m/s^2, S is the cross-sectional area of the tank (m^2), s is the cross-sectional area of the outlet pipe (m^2), and $u(t)$ is the volume flow rate into the tank (m^3/s). The volume flow rate out of the tank is assumed constant.

$$S\frac{dh(t)}{dt} = u(t) - s\sqrt{2gh(t)} \tag{8.2.19}$$

The output of a controller that will track the desired fluid level is given by

$$u(t) = s\sqrt{2gh(t)} - \alpha S\big(h(t) - h_d(t)\big) \tag{8.2.20}$$

The parameter α is strictly positive, dimensionless, and constant. In this, we set α to 1. Substitution of Equation (8.2.20) into (8.2.19) gives the closed-loop system dynamic

$$\frac{dh(t)}{dt} + \alpha\big(h(t) - h_d(t)\big) = 0 \tag{8.2.21}$$

We use Equation (8.2.21) as our reference system for which a fuzzy controller is designed. From a simulation of Equation (8.2.21), a single-input, single-output fuzzy controller is developed. The fuzzy controller input is ERROR = $h(t) - h_d(t)$, and its output is FLOWRATE, $u(t)$. By setting the tank's maximum height to be 30 units, the simulation model gives ERROR to be in the range of [−30, 30], and FLOWRATE in the range of [−8, 10]. Both fuzzy variables are partitioned into five primary fuzzy sets: NB (Negative Big); NS (Negative Small); ZE (Zero); PS (Posi-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 8.2 Membership Function Parameters of ERROR and FLOWRATE

variable	$\mu(x)$	$\bar{x}$	σ
ERROR	NB	-30	200
	NS	-15	200
	ZE	0	200
	PS	15	200·
	PB	30	200
FLOWRATE	NB	-8	10
	NS	-5	10
	ZE	0	10
	PS	5	10
	PB	30	10

tive Small); and PB (Positive Big). The membership functions are Gaussian-like with parameters shown in Table 8.2.

Rules relating the ERROR to the FLOWRATE are: (NB $\Rightarrow$ PB), (NS $\Rightarrow$ PS), (ZE $\Rightarrow$ ZE), (PS $\Rightarrow$ NS), and (PB $\Rightarrow$ NB).

The response of the fuzzy controller, both before and after tuning, in tracing a desired fluid level are shown in Figure 8.4. The desired level is described by $h_d(t) = 10 + 10\,(1 - \exp(-t/20))$. It shows that the tuned fuzzy

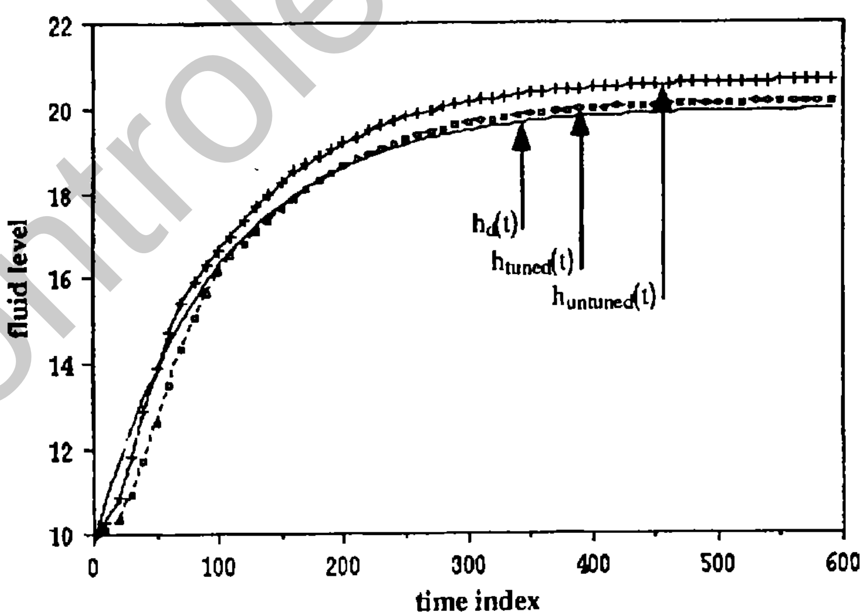


Figure 8.4 Performance of the fuzzy controller, before and after tuning using Method 2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controller performance is improved, especially in reducing the offset error between the desired and the actual fluid level.

Method 3—Gradient Approach The gradient method is formulated as follows. Find the centroid vector **c**, such that an objective function, defined by $e^2 = (u - u_d)^2 = f(\mathbf{c})$, is minimized.

Let the first guess of the centroid vector be $\mathbf{c}_0$. Update the centroid vector to $(\mathbf{c}_0 + d\mathbf{c})$ such that $f(\mathbf{c}_0 + d\mathbf{c}) < f(\mathbf{c}_0)$, or equivalently, maximize $(f(\mathbf{c}_0) - f(\mathbf{c}_0 + d\mathbf{c}))$. The distance traveled as a result of the incremental vector $d\mathbf{c}$ is given by

$$(ds)^2 = \sum_{i=1}^{N} \left(d\mathbf{c}_i\right)^2 \tag{8.2.22}$$

where $i = 1, 2, \ldots, N$, and N is the number of fuzzy logic controller output primary fuzzy sets. Equation (8.2.22) can be written as

$$1 - \sum_{i=1}^{N} \left(\frac{d\mathbf{c}_i}{ds}\right)^2 = 0 \tag{8.2.23}$$

In this formulation, the maximization problem is equivalent to finding dc_i/ds such that df/ds is maximized, where, by chain rule

$$\frac{df}{ds} = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial c_i} \frac{dc_i}{ds}\right) \tag{8.2.24}$$

Equivalently, maximize df/ds such that Equation 8.2.23) is satisfied. This problem can be solved by using Lagrange multipliers (Strang, 1986), thus

$$L\left(\frac{df}{ds}, k\right) = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial c_i} \frac{dc_i}{ds}\right) + k\left[1 - \sum_{i=1}^{N} \left(\frac{dc_i}{ds}\right)^2\right] \tag{8.2.25}$$

Therefore, the maximization of df/ds is obtained by satisfying

$$\frac{\partial L}{\partial\left(\frac{dc_i}{ds}\right)} = \frac{\partial f}{\partial c_i} - 2k\frac{dc_i}{ds} = 0 \tag{8.2.26}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and

$$\frac{\partial L}{\partial k} = 1 - \sum_{i=1}^{N} \left(\frac{dc_i}{ds} \right)^2 = 0 \tag{8.2.27}$$

Solving Equations (8.2.26) and (8.2.27) gives

$$\frac{dc_i}{ds} = \frac{1}{2k} \frac{\partial f}{\partial c_i} \tag{8.2.28}$$

where

$$k = \pm \tfrac{1}{2} \left[\sum_{i=1}^{N} \left(\frac{\partial f}{\partial c_i} \right)^2 \right]^{\frac{1}{2}} \tag{8.2.29}$$

Therefore,

$$\frac{dc_i}{ds} = \pm \frac{\dfrac{\partial f}{\partial c_i}}{\left[\displaystyle\sum_{i=1}^{N} \left(\frac{\partial f}{\partial c_i} \right)^2 \right]^{\frac{1}{2}}} \tag{8.2.30}$$

The denominator in Equation (8.2.30) is a normalizing factor. Therefore, we can rewrite this equation as

$$\frac{dc_i}{ds} = \pm \frac{\partial \bar{f}}{\partial c_i} \tag{8.2.31}$$

where $\bar{f}$ denotes normalization. From Equation (8.2.31), the amount by which the centroid is to be modified is given by

$$dc_i = \pm K \frac{\partial \bar{f}}{\partial c_i} \tag{8.2.32}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where ds is substituted by K, denoting step size for discrete implementation of Equation (8.2.31). To choose the sign that is appropriate for our purpose, substitute Equation (8.2.30) into Equation (8.2.24) and obtain

$$\begin{aligned}\frac{df}{ds} &= \pm \frac{\sum_{i=1}^{N}\left(\frac{\partial f}{\partial c_i}\right)^2}{\left[\sum_{i=1}^{N}\left(\frac{\partial f}{\partial c_i}\right)^2\right]^{\frac{1}{2}}} \\ &= \pm\left[\sum_{i=1}^{N}\left(\frac{\partial f}{\partial c_i}\right)^2\right]^{\frac{1}{2}}\end{aligned} \tag{8.2.33}$$

This means that the plus sign is taken for maximization, and the minus sign for minimization of the objective function $f(\mathbf{c})$. Since we are minimizing the square of the error, the centroid update equation is given by

$$c_i(k+1) = c_i(k) - K\frac{\partial (e(k))^2}{\partial c_i} \tag{8.2.34}$$

where k is time step index. The gradient of the square of the error term is given by

$$\frac{\partial (e(k))^2}{\partial c_i} = \frac{\partial}{\partial c_i}\left(u_c(k) - u_d(k)\right)^2 \tag{8.2.35}$$

Substituting for $u_c(k)$ from Equation (8.2.2) gives

$$\frac{\partial (e(k))^2}{\partial c_i} = \frac{\partial}{\partial c_i}\left(\left(\frac{\sum_{i=1}^{N}\mu_{R_i}c_i}{\sum_{i=1}^{N}\mu_{R_i}}\right)_k - u_d(k)\right)^2 \tag{8.2.36}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Equation (8.2.36) is equivalent to

$$\begin{aligned}\frac{\partial(e(k))^2}{\partial c_i} &= \frac{\partial}{\partial c_i}\left(\sum_{i=1}^{N} a_i(k)c_i(k)-u_d(k)\right)^2 \\ &= 2e(k)a_i(k)\end{aligned} \tag{8.2.37}$$

Therefore, the centroid update equation is given by

$$c_i(k+1)=c_i(k)-2K\,e(k)a_i(k) \tag{8.2.38}$$

This method has a simpler form than the least-squares approach of Method 2. It requires much less computation, and therefore, is more suitable for on-line estimation. Basically, this method is a predictor estimator, i.e., the next centroid values to be used are determined using data at the previous time step.

Example 8.2.3 The same fuzzy controller that was designed in Example 8.2.2 is used again. The initially designed centroid vector is used as the starting values of the centroids, $c_i(0)$, for the recursion in Equation (8.2.38). We used the same value, 0.02, for the estimator gains of all primary fuzzy sets of the output variable.

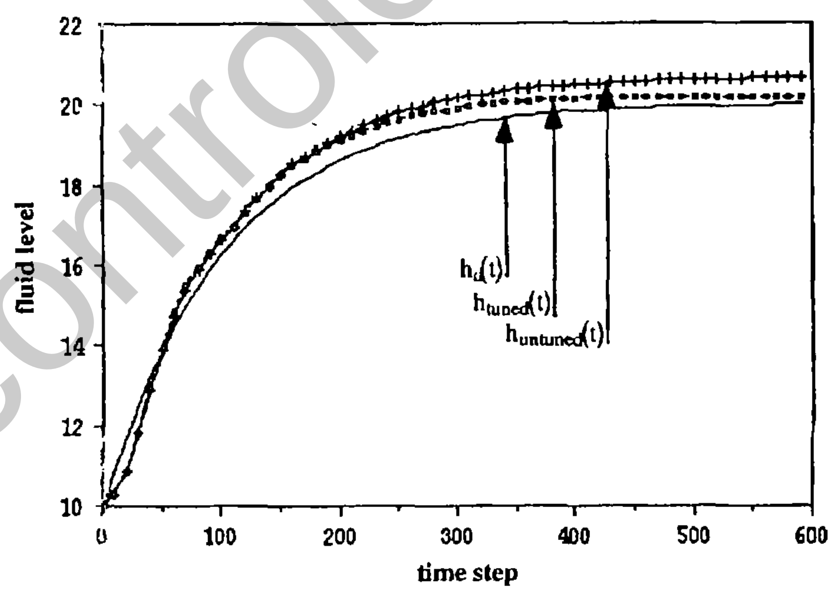


Figure 8.5 Performance of the fuzzy controller, before and after tuning using the gradient approach.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Figure 8.5 compares the performance of the "raw" fuzzy controller with the fuzzy controller that is tuned using this method. Figure 8.6 shows the variations of each of the five centroid values as the tuning progresses. It should be noted that this tuning procedure does not strive to fix the centroids at any values; rather the centroid values are adapted to the changing plant condition (Alang-Rashid, 1992).

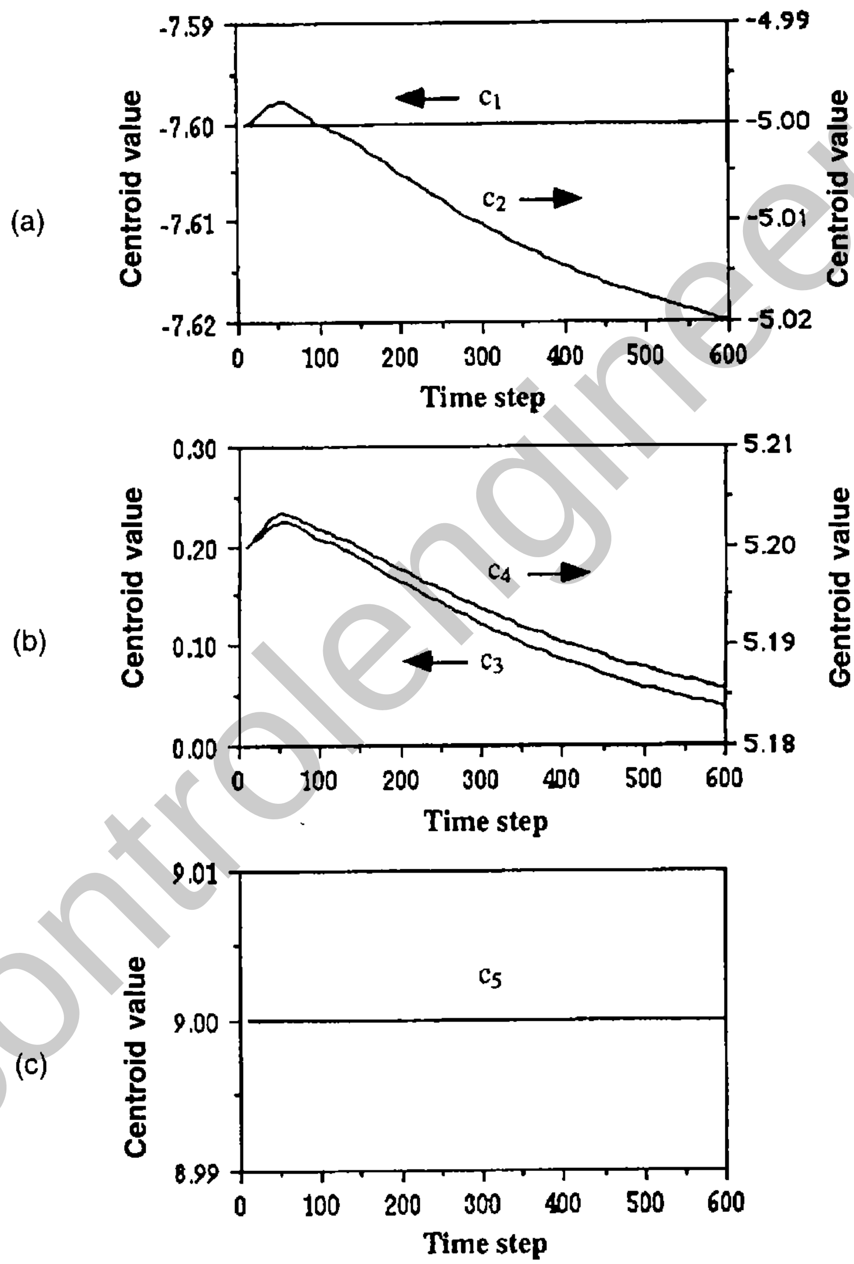


Figure 8.6. Variations of the centroid values at each time step: (a) NB and NS, (b) ZE and PS, and (c) PB.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

8.2.2 Adaptive Fuzzy Multiterm Controllers

Thus far, the schemes discussed are based on either estimation of a fuzzy controller's parameters or through a supervisory rule to adapt the low-level controller or not. In this section, fuzzy logic will be used to "tune" or "adapt" a multiterm controller such as a PD, PI, or a PID. The discussions here are based, in part, on the works of Zhao *et al.*(1993) and Barak (1993).

Consider, with no loss of generality, a fuzzy-tuned PID controller shown in Figure 8.7. The PID controller can be represented by either one of the following two forms:

Continuous:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \dot{e}(t) \tag{8.2.39}$$

or

Discrete:

$$u(k) = K_p e(k) + K_i T_s \sum_{i=1}^{n} e(i) + \frac{K_d}{T_s} \Delta e(k) \tag{8.2.40}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively, $T_i = K_p / K_i$ and $T_d = K_d / K_p$ are known as the integral and derivative time constants, $\Delta e(k) \triangleq e(k) - e(k-1)$, T_s is the sampling period, and n is the number of samples.

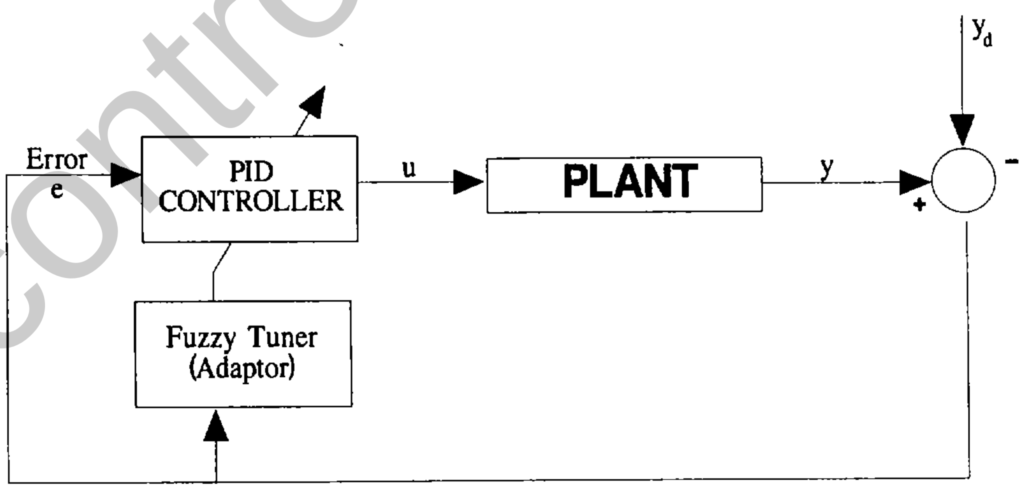


Figure 8.7 A fuzzy tuner (adaptive) PID control system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The basic approach in fuzzy tuning of a multiterm controller such as a PID is to devise fuzzy supervisory rules of one of the following types of rules:

i) IF $e(k)$ is A_i and $\Delta e(k)$ is B_i,
THEN K_p is C_i, K_i is D_i, and K_d is E_i (8.2.41)

ii) IF $e(k)$ is A_i and $\Delta e(k)$ is B_i,
THEN K_p is C_i, T_i is D_i, and T_d is E_i (8.2.42)

iii) IF $e(k)$ is A_i and $\Delta e(k)$ is B_i,

$$\text{THEN } u(k) = K^i_{p_o} e(k) + \left(K^i_{i_o} T_s\right)\sum_j e(j) + \left(K^i_{d_o} / T_s\right)\Delta e(k) \qquad (8.2.43)$$

iv) IF $e(k)$ is A_i and $\Delta e(k)$ is B_i,
THEN K_p is C_i, K_d is D_i, and $\alpha = \alpha_i$ (8.2.44)

In the above equations, A_i, B_i, C_i, D_i, and E_i are linguistic variables of the ith rule ($i = 1, 2, \ldots, m$) which can be represented by appropriate functions. T_i and T_d in Equation (8.2.42) have been defined before; $K^i_{p_o}$, $K^i_{i_o}$, and $K^i_{d_o}$ in Equation (8.2.41) are constants, and α_i in Equation (8.2.44) is a constant, while α is defined by

$$T_i = \alpha T_d \qquad (8.2.45)$$

and

$$K_i = K_p/(\alpha T_d) = K_p^2/(\alpha K_d) \qquad (8.2.46)$$

The above two equations eliminate K_i as an independent parameter and introduce α as a third tuning parameter (Zhao *et al.*, 1993). The rule listed in Equation (8.2.43) is similar to the Takagi-Sugeno type which was first introduced by Equation (7.2.17a). The membership functions for A_i and B_i can be a set of triangular forms such as those in Figure 8.8 without any loss of generality. The fuzzy sets C_i, D_i, and E_i for K_p, K_i, and K_d can either be of the same form as those in Figure 8.8 or as a variety of the "large" or "small" variables shown in Figure 8.9.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Zhao *et al.* (1993) have presented a number of simulated examples for a fuzzy-tuned PID control system with rules like Equation (8.2.44) and an exponential variation for the membership function of Figure 8.9. They have made a comparison with the standard PID controller of Ziegler and Nichols (1942) and have outperformed it through an objective criterion. The authors have chosen two performance indices—integral square error, $\mathrm{ISE} = \int e^2(t)dt$, and integral absolute error, $\mathrm{IAE} = \int |e(t)|dt$—which are standard in optimal parameter design problems (Jamshidi and Malek-Zavarei, 1986).

Example 8.2.4 Let us reconsider the inverted pendulum problem of Figure 7.8 and Equation (7.2.24). It is desired to design an adaptive PD controller for it.

SOLUTION: Since the inverted pendulum's states are θ and $\dot{\theta}$, then a full-state feedback in this formulation is equivalent to a PD controller. This controller would stabilize the system. However, the performance of the controller can be influenced by the position of the closed-loop poles (CLP), i.e., closed-loop poles placed far to the left of the imaginary axis will create a system with a fast response; conversely, if the CLP are placed closer to the imaginary axis, a system with a slow response will result. Fuzzy logic inference tries to take advantage of this situation by acting in a supervisory mode over the state-feedback controller and adjusting its coefficients so that improved performance is achieved. Figure 8.10 gives a block diagram of this scenario.

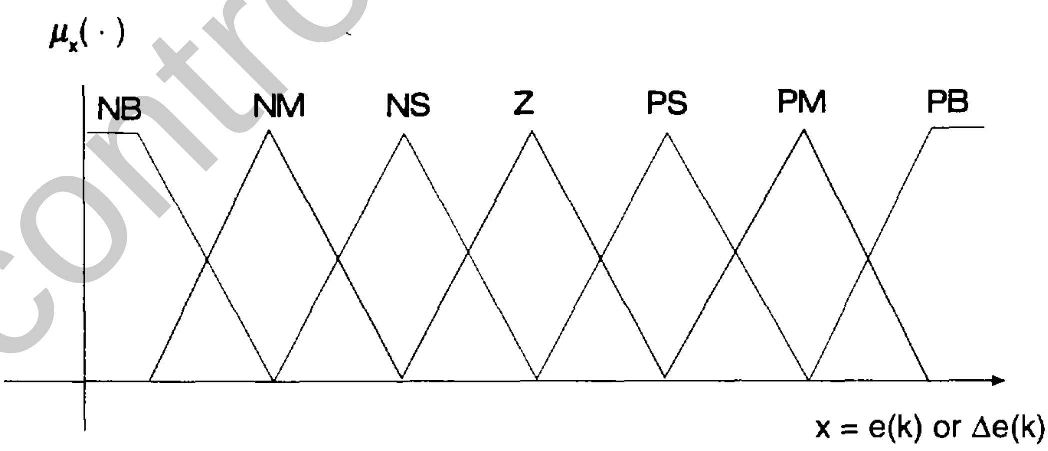


Figure 8.8 Typical triangular membership functions for error $e(k)$ and its variation $\Delta e(k)$.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

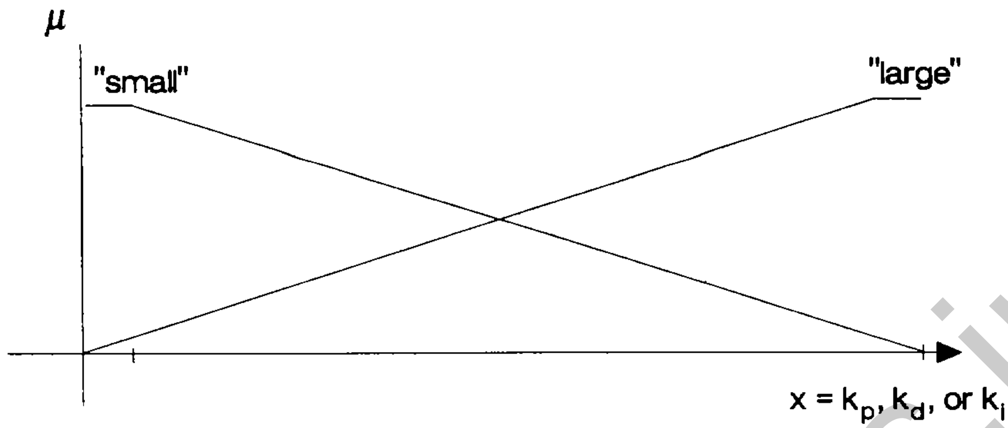


Figure 8.9 Membership functions for parameters of a multiterm controller.

From Figure 8.10 it is clear that the input to the fuzzy controller is the measured angular position (state variable x_3), and the outputs are the state-feedback coefficients that form the control input u. In order to define realistic limits for k_i coefficients, carriage track length was limited to 1 meter and carriage velocity to ±1 meter/second, as in Jamshidi *et al.* (1992). Keeping these realistic design criteria in mind and using arbitrarily chosen pole sets P_{min} and P_{max} in the Left-Hand Plane (LHP), a standard MATLAB (Jamshidi *et al.* 1992) pole placement routine was used to determine the

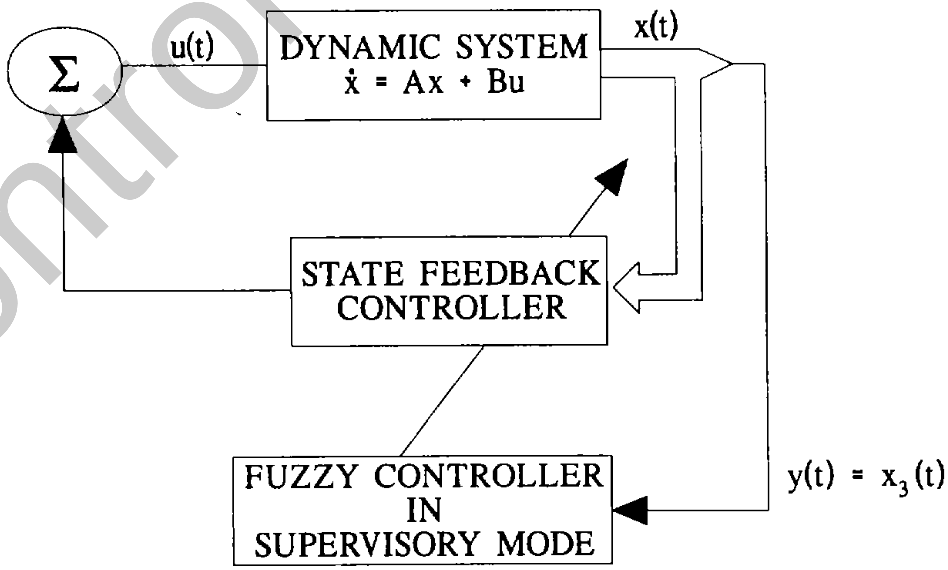


Figure 8.10 Block diagram of FLC in adaptive mode.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

limits of k_i coefficients in the following way:

for the minimum: place (A, b, P_{min})

$$P_{min} = [\,-0.50 \quad -1.00 \quad -1.50 \quad -2.00\,]$$

$$k_{min} = [\,-0.39 \quad -1.61 \quad -34.33 \quad -16.42\,],$$

for the maximum: place (A, b, P_{max})

$$P_{max} = [\,-2.0 \quad -4.0 \quad -6.0 \quad -8.0\,]$$

$$k_{max} = [\,-99.2 \quad -103.3 \quad -596.9 \quad -291.5\,]$$

Membership functions were designed as a uniform distribution of triangles between the design limitations, i.e., for measured x_3 between $\pm 5°$; for the coefficients k_i, a multiplication coefficient *K-FACTOR* was designed such that the CLP lie on the real axis between the previously determined values of P_{min} and P_{max} (see Figures 8.11 and 8.12).

Rules for the input/output mapping were designed intuitively, i.e., if the measured angular position is large, the CLP should lie as far as possible in the LHP; conversely, if the measured angular position for fast initial re-

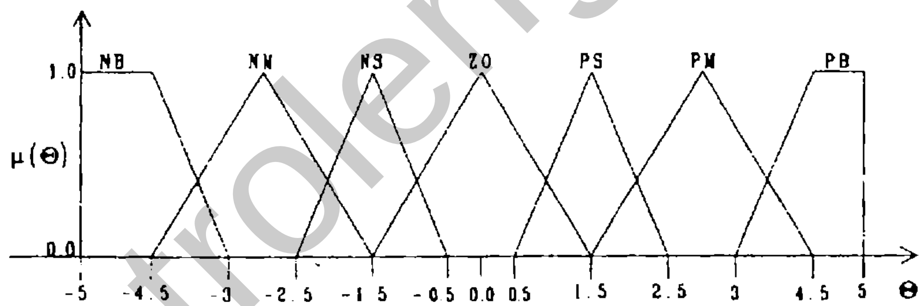


Figure 8.11 Theta membership function.

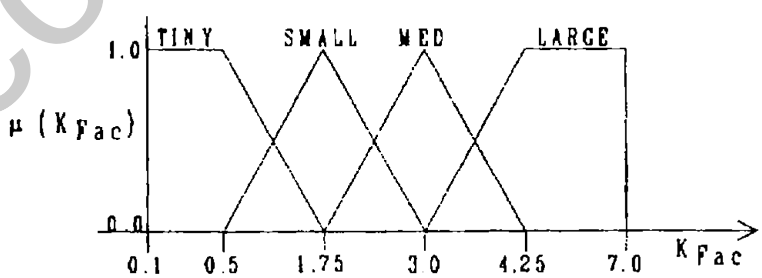


Figure 8.12 K-FACTOR membership function.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

sponse is relatively small, the CLP should lie relatively close to the y-axis so that settling time is improved. In this fashion, seven rules were designed as follows:

Rule 1: *If THETA is PB then K-FACTOR is LARGE.*

Rule 2: *If THETA is PM then K-FACTOR is MEDIUM.*

Rule 3: *If THETA is PS then K-FACTOR is SMALL*

Rule 4: *If THETA is ZO then K-FACTOR is ZERO.*

Rule 5: *If THETA is NS then K-FACTOR is SMALL.*

Rule 6: *If THETA is NM then K-FACTOR is MEDIUM.*

Rule 7: *If THETA is NB then K-FACTOR is LARGE.*

A plot of feedback gain modifier, *K-FACTOR*, in response to the initial conditions of $\theta = -5°$ is presented in Figure 8.13.

Simulation results were achieved using a Microsoft-C Optimizing Compiler (Microsoft, Inc., 1990), a Togai Fuzzy-C Compiler (Togai InfraLogic, Inc., 1991), and numerical routines to solve the differential equations. The results are shown in Figures 8.14 and 8.15.

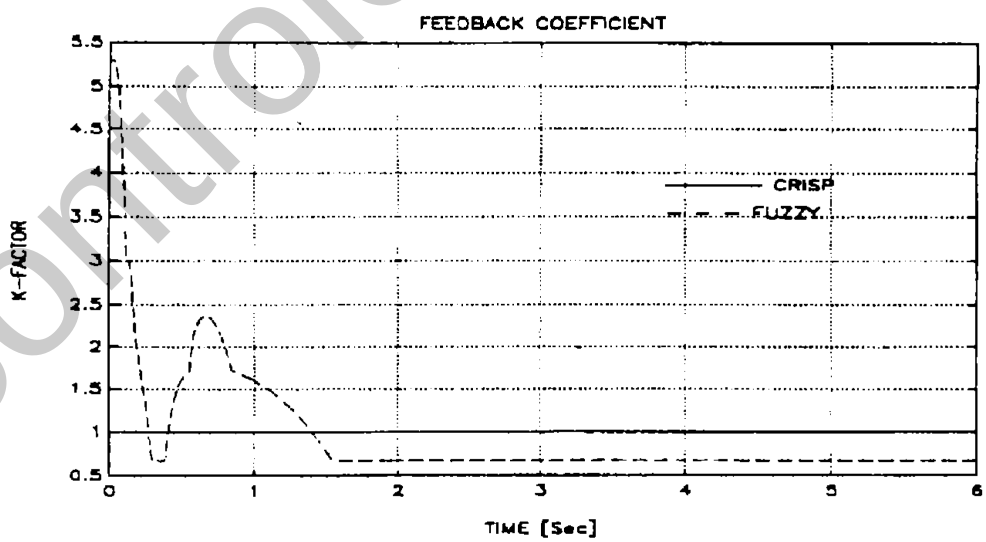


Figure 8.13 FLC vs. state-feedback controller (feedback gain coffiecient).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Figures 8.14 and 8.16 show that, because of the adjustment of the state-feedback coefficients, the FLC had a faster initial response and a smaller overshoot/undershoot than its state-feedback counterpart. This is related to FLC state-feedback coefficients' adjustment capability. At first, a large feedback gain was used to achieve fast response; later, the coefficients were decreased in order to achieve a small overshoot.

Figures 8.14 through 8.17 also compare the performances of the conventional state-feedback controllers on all four feedback states, i.e., cart position, cart velocity, angular position, and angular velocity, respectively.

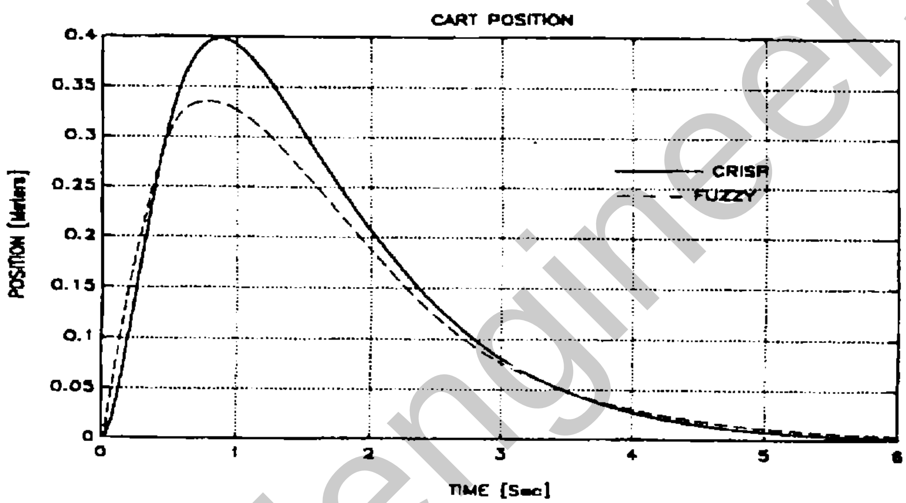


Figure 8.14 FLC vs. state-feedback controller (carriage position).

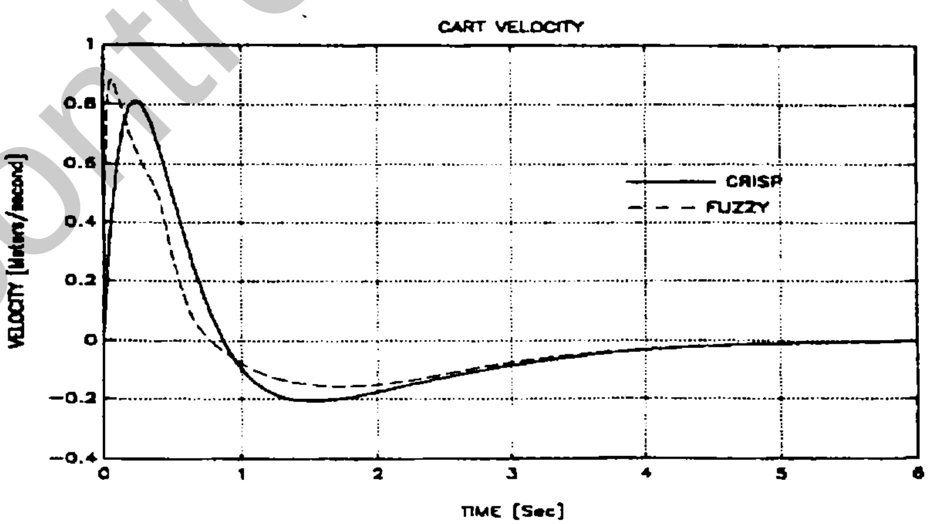


Figure 8.15 FLC vs. state-feedback controller (carriage velocity).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

8.2.3 *Indirect Adaptive Fuzzy Control*

Wang (1994a, 1994b) has proposed a group of adaptive fuzzy control design techniques which combine three important issues of this chapter into a unified approach, i.e., adaptation, hierarchy (supervision), and stability. Due to the space limitations, we can not cover various approaches that Wang (1994a) has proposed. Here, we make a brief presentation of only one of the approaches—indirect adaptive fuzzy control systems.

Two adaptive control approaches have been well documented in control literature (Åström and Wittenmark, 1991). The MRAC-model

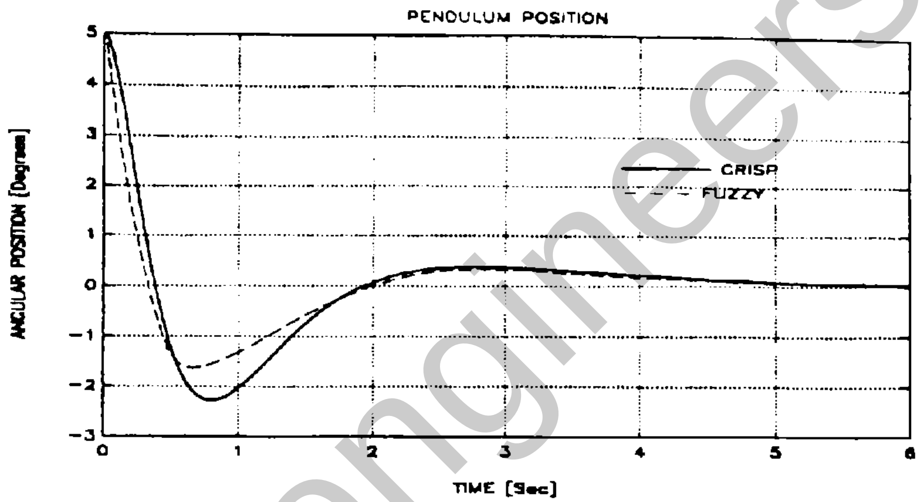


Figure 8.16 FLC vs. state-feedback controller (angular position).

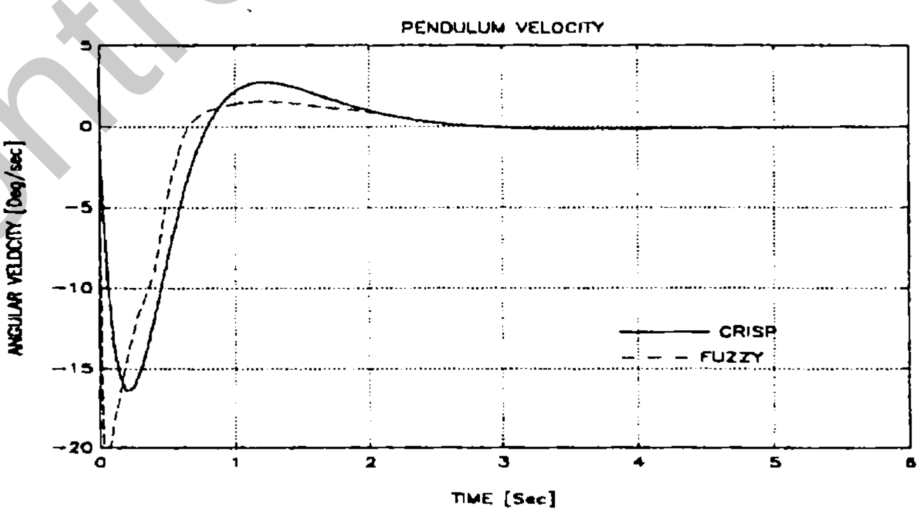


Figure 8.17 FLC vs. state-feedback controller (angular velocity).

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

reference adaptive control (see Chapter 5) is also sometimes called a *direct* adaptive control. On the other hand, an self-tuning regulator STR) is called an *indirect* adaptive control. Unlike MRAC, in STR the system's model is identified first and the controller is redesigned periodically, leading to an adaptation of control to plant variations. Now, consider the following definitions.

Definition 8.1 An adaptive fuzzy control is said to be direct if the corresponding fuzzy rules define the control actions to be taken under various conditions of the system's state. For example, in a direct adaptive fuzzy control system, one may have a rule like: "IF speed is high, THEN apply more force to the brake." Hence, direct adaptive fuzzy control may be said to be the fuzzy counterpart to MRAC. In other words, in a direct adaptive controller, the rules are incorporated into the controller description.

Definition 8.2 An adaptive fuzzy control is said to be indirect if the corresponding fuzzy rules describe the behavior of the unknown "plant" rather than the controller which is to control it. For example, the rule used in Definition 8.1 would be restated as: "IF you apply more force to the brake, THEN the speed of the car will decrease." Here again, an indirect adaptive control system can be thought of as the fuzzy counterpart of an STR adaptive control system. In other words, in an indirect adaptive controller the rules incorporate fuzzy descriptions of the plant.

In this section an indirect adaptive fuzzy controller will be designed for the following special class of nonlinear systems,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_n &= f(\mathbf{x}) + g(\mathbf{x})u \end{aligned} \tag{8.2.47}$$

$$y = (1\ \ 0\ \ 0\ \ 0\,)\,\mathbf{x} \tag{8.2.48}$$

where $\mathbf{x} = (x_1\ x_2\ \dots\ x_n)^T$ is the nth order state vector, u is the single input to the system (assumed be separable from state vector $\mathbf{x}$), and y is the single output variable. Functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are unknown except that $g(x) > 0$ for $\mathbf{x} \in X_c \subset R^n$, where X_c is certain controllability space. System (8.2.47)–(8.2.48) is called a nonlinear system of *normal* form with relative degree n (Slotine and Li, 1991). The control problem can now be stated as follows: Find a feedback fuzzy control $u = u\ (\mathbf{x} \mid \theta)$ with an adaptation law for

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

adjusting the parameter vector θ so that (Wang, 1994a)

(i) the closed-loop system is globally stable, i.e., $\mathbf{x}(t)$, $\theta(t)$, and $u(\cdot)$ are uniformly bounded, or $|\mathbf{x}(t)| < k_x < \infty$, $|\theta(t)| \le K_\theta < \infty$, and $|u(\cdot)| \le K_u < \infty$ for all $t \ge 0$. Here, K_x, K_θ, and K_u are designer-specific parameters, and

(ii) the $\lim_{t\to\infty} e(t) = 0$, where $e(t) \stackrel{\Delta}{=} y_d(t) - y(t)$ is the tracking error, and $y_d(t)$ is a given bounded desired reference signal for the overall system.

Controller Design

Let a parameter vector $\mathbf{k} = (k_1 \ \dots \ k_n)^T \in R^n$ exist such that the polynomial $s^n + k_n s^{n-1} + \dots + k_1$ is Hurwitz. Should the vectors $f(\mathbf{x})$ and $g(\mathbf{x})$ in Equation (8.2.47) be known, as in model-based fuzzy control situations, the desired control law would be

$$u(t) = \frac{1}{g(\mathbf{x})}\left[y_d^{(n)} + \mathbf{k}^T \mathbf{e}(t) - f(\mathbf{x})\right] \tag{8.2.49}$$

where $\mathbf{e} = \left(e \ \dot{e} \ \dots \ e^{(n-1)}\right)^T \in R^n$ is the error vector and g($\mathbf{x}$) is the nonzero coefficient of nth order differential equation in (8.2.47). Vector $\mathbf{k}$ was defined earlier and $\mathbf{x}$ represents a vector quantity. Upon the application of Equation (8.2.49) to (8.2.47), one gets

$$e^{(n)} + k_n e^{(n-1)} + \dots + k_1 e = 0 \tag{8.2.50}$$

for which it follows that $\lim_{t\to\infty} e(t) = 0$, one of the main goals of the controller design. If $f(\mathbf{x})$ and $g(\mathbf{x})$ are not known, then one needs to find an estimate of these functions. As a sequel, fuzzy logic will be used to complete the design process in the absence of exact knowledge of the plant dynamics.

The functions f and g in Equation (8.2.49) can be replaced by their respective fuzzy systems $\hat{f}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$, where either one of these functions

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

is defined by (Wang, 1994a):

$$\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^{M} \bar{y}^i \left\{ \prod_{j=1}^{n} \exp\left[-\left(\left(x_j - \bar{x}_j^i \right) / \sigma_j^i \right)^2 \right] \right\}}{\sum_{i=1}^{M} \left\{ \prod_{j=1}^{n} \exp\left[-\left(\left(x_j - \bar{x}_j^i \right) / \sigma_j^i \right)^2 \right] \right\}} \tag{8.2.51}$$

where $\bar{y}^i$, $\bar{x}_j^i$, and σ_j^i are adjustable parameters. This fuzzy system representation assumes a singleton fuzzifier, Gaussian membership functions, center of gravity defuzzifier, and the max-product inference rule (see Appendix A). The controller Equation (8.2.49) can now be rewritten by

$$u_c = \frac{1}{\hat{g}(\mathbf{x})} \left[y_d^{(n)} + k^T e(t) - \hat{f}(\mathbf{x}) \right] \tag{8.2.52}$$

which represents the so-called *certainty equivalent controller* (Sastry and Bodson, 1989). If Equation (8.2.52) is now applied to Equation (8.2.47) with some straightforward manipulation and definition of the error function $e(t)$, we get

$$e^{(n)} = -\mathbf{k}^T \mathbf{e} + \left[\hat{f}(\mathbf{x}) - f(\mathbf{x}) \right] + \left[\hat{g}(\mathbf{x}) - g(\mathbf{x}) \right] u_c \tag{8.2.53}$$

where n-dimensional vectors $\mathbf{e}$ and $\mathbf{k}$ were defined in Equation (8.2.49). Equation (8.2.53) can be equivalently rewritten in its state space form as

$$\dot{\mathbf{e}} = \mathbf{A}_c \mathbf{e} + \mathbf{b}_c \left\{ \left[\hat{f}(\mathbf{x}) - f(\mathbf{x}) \right] + \left[\hat{g}(\mathbf{x}) - g(\mathbf{x}) \right] u_c \right\} \tag{8.2.54}$$

where $\mathbf{A}_c$ and $\mathbf{b}_c$ are of standard controllable cannonical forms,

$$\mathbf{A}_c = \left[\begin{array}{c|ccc} 0 & & & \\ 0 & & I_{n-1} & \\ \vdots & & & \\ 0 & & & \\ \hline -k_1 & -k_2 & \cdots & -k_n \end{array} \right], \; \mathbf{b}_c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{8.2.55}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and I_{n-1} is an $(n-1) \times (n-1)$ identity matrix. Since k_i's represent coefficients of a Hurwitz polynomial $k(s)$, $\mathbf{A}_c$ is a stable matrix with $k(s) = \det(sI - A_c) = s^n + k_n s^{n-1} + \ldots + k_1$ and it is well known (Jamshidi *et al.*, 1992) that there exists a unique positive definite symmetric $n \times n$ matrix P satisfying the following vector Lyapunov equation;

$$\mathbf{A}_c^T \mathbf{P} + \mathbf{P}\mathbf{A}_c + \mathbf{Q} = 0 \tag{8.2.56}$$

where $\mathbf{Q}$ is an arbitrary $n \times n$ symmetric positive definite matrix. Now, let us choose a Lyapunov function $\upsilon = \frac{1}{2}\mathbf{e}^T P\mathbf{e}$. By virtue of Equations (8.2.54) and (8.2.56), one gets

$$\begin{aligned} \dot{\upsilon} &= \tfrac{1}{2}\dot{\mathbf{e}}^T P\mathbf{e} + \tfrac{1}{2}\mathbf{e}^T P\dot{\mathbf{e}} \\ &= -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} + \mathbf{e}^T P\mathbf{b}_c\left\{\left[\hat{f}(\mathbf{x}) - f(\mathbf{x})\right] + \left[\hat{g}(\mathbf{x}) - g(\mathbf{x})\right]u_c\right\} \end{aligned} \tag{8.2.57}$$

It is clear now that if we require $\dot{\upsilon} \le 0$ with $\upsilon > \overline{V}$, a large constant, then by virtue of Lyapunov's second method, the error $e(t) = y_d(t) - y(t) = y_d(t) - x_1(t)$ (see Equation (8.2.48)) would be bounded as t approaches infinity. The difficulty at hand is that it is not too obvious how the last term in Equation (8.2.57) would become negative.

Stabilizing Supervisory Controller

To overcome the difficulty in proving that $\dot{\upsilon} \le 0$ in Equation (8.2.57), Wang (1994b) proposes a supervisory controller u_s to append the certainty equivalent controller u_c, i.e., let

$$u = u_c + u_s \tag{8.2.58}$$

Now, using Equation (8.2.58) in Equation (8.2.47) and conducting the same mathematical manipulations, we get a new error equation

$$\mathbf{e} = \mathbf{A}_c\mathbf{e} + \mathbf{b}_c\left\{\left[\hat{f}(\mathbf{x}) - f(\mathbf{x})\right] + \left[\hat{g}(\mathbf{x}) - g(\mathbf{x})\right]u_c - g(\mathbf{x})u_s\right\} \tag{8.2.59}$$

Using Equations (8.2.59) and (8.2.56), the time derivative of υ is given by

$$\begin{aligned} \dot{\upsilon} &= -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} + \mathbf{e}^T P\mathbf{b}_c\left\{\left[\hat{f}(\mathbf{x}) - f(\mathbf{x})\right] + \left[\hat{g}(\mathbf{x}) - g(\mathbf{x})\right]u_c - g(\mathbf{x})u_s\right\} \\ &\le -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} + \left|\mathbf{e}^T P\mathbf{b}_c\right|\left[\left|\hat{f}(\mathbf{x})\right| + \left|f(\mathbf{x})\right| + \left|\hat{g}(\mathbf{x})u_c\right| + \left|g(\mathbf{x})u_c\right|\right] \\ &\quad - \mathbf{e}^T P\mathbf{b}_c g(\mathbf{x})u_s \end{aligned} \tag{8.2.60}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

If we assume that $\hat{f}$ and $\hat{g}$ satisfy the following bounds,

$$
\begin{aligned}
&|f(x)| \le f^U \\
&0 < g_L \le g(x) \le g^U
\end{aligned}
\tag{8.2.61}
$$

then our plant Equation (8.2.47) is not totally unknown, i.e, some bounds on f and g are known. Based on conditions in Equation (8.2.61) and our ultimate goal in Equation (8.2.60), we choose u_s as

$$
u_s = I^* \,\mathrm{SGN}\left(\mathbf{e}^T \mathbf{P}\mathbf{b}_c\right)\frac{1}{g_L}\left[\left|\hat{f}(\mathbf{x})\right| + f^U + \left|\hat{g}(\mathbf{x})u_c\right| + \left|g^U u_c\right|\right]
\tag{8.2.62}
$$

where $I^* = 1$(or 0) if $\upsilon > \bar{V}$ (or $\upsilon > \bar{V}$) and the sign function $\mathrm{SGN}(a) = 1$(or -1) if $a > 0$ (or $a < 0$). Now, substituting Equations (8.2.62) into (8.2.60) and considering the case when $\upsilon > \bar{V}$, one has

$$
\begin{aligned}
\dot{\upsilon} \le {} & -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} + \left|e^T \mathbf{P}\mathbf{b}_c\right|\left[\left|\hat{f}\right| + |f| + \left|\hat{g}u_c\right| + \left|gu_c\right|\right] \\
& -\frac{g}{g_L}\left(\left|\hat{f}\right| + f^U + \left|\hat{g}u_c\right| + \left|g^U u_c\right|\right) \le -\tfrac{1}{2}\mathbf{e}^T \mathbf{Q}\mathbf{e} \le 0
\end{aligned}
\tag{8.2.63}
$$

This now implies that a combination of two controllers u_c in (8.2.52) and u_s in (8.2.62) would indeed stabilize the system (8.2.47), i.e., making sure that error function $e(t)$ and vector $\mathbf{e}$ remain bounded. It is noted that all the terms in the two controllers u_c and u_s in (8.2.52) and (8.2.62), respectively, are either known or measurable. Moreover in lieu of the switching function I^* in (8.2.62), the supervisory controller would not need to do anything (*supervise*) if the fuzzy controller u_c in (8.2.52) would be sufficient in controlling the system. In effect, u_s *supervises* over the stabilizing capability of the fuzzy controller u_c.

What is left now is to find expressions for $\hat{f}$ and $\hat{g}$ which would adaptively force tracking error e to converge to zero.

Adaptation Law

We begin by noting that the fuzzy logic systems (or a set of IF-THEN rules) described by $\hat{f}$ and $\hat{g}$ described by (8.2.51) can be alternatively represented by (Wang, 1994a).

$$
\hat{f}(\mathbf{x}) \triangleq \hat{f}(\mathbf{x}|\theta) = \sum_{i=1}^{M} \theta_i h_i(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{h}(\mathbf{x})
\tag{8.2.64}
$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where $\boldsymbol{\theta} = (\theta_1, \ldots \theta_M)^T$, $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_M(\mathbf{x}))^T$ and $h(\mathbf{x})$ is called *fuzzy basis function* and is defined by

$$h_i(\mathbf{x}) = \frac{\prod_{j=1}^{n} \mu_{F_j^i}(x_j)}{\sum_{i=1}^{M} \prod_{j=1}^{n} \mu_{F_j^i}(x_j)} \tag{8.2.65}$$

and θ_i are unknown adjustable parameters and $\mu_{F_j^i}$ are membership functions.

Let us define

$$\boldsymbol{\theta}_f^* = \arg\min_{\boldsymbol{\theta}_f \in \Omega_f} \left[\sup_{\mathbf{x} \in U_c} \left| \hat{f}(\mathbf{x}|\boldsymbol{\theta}_f) - f(\mathbf{x}) \right| \right] \tag{8.2.66}$$

$$\boldsymbol{\theta}_g^* = \arg\min_{\boldsymbol{\theta}_g \in \Omega_g} \left[\sup_{\mathbf{x} \in U_c} \left| \hat{g}(\mathbf{x}|\boldsymbol{\theta}_g) - g(\mathbf{x}) \right| \right] \tag{8.2.67}$$

where Ω_f and Ω_g are designer-specified constrained sets for θ_f and θ_g, respectively and U_c is the Universe of discourse for x. For Ω_f, we require θ_f to be bounded, and for fuzzy logic system (8.2.51), $\sigma_j^i > 0$, i.e.,

$$\Omega_f = \left\{ \boldsymbol{\theta}_f : \left| \theta_f \right| \le M_f \, , \sigma_j^i \ge \sigma \right\} \tag{8.2.68}$$

where M_f and σ are designer-specified positive constants. Note, that the designer can equally choose (8.2.64) in which case the assumption on σ_j^i would no longer be relevant. In a similar fashion, we assume

$$\Omega_g = \left\{ \boldsymbol{\theta}_g : \left| \boldsymbol{\theta}_g \right| \le M_g \, , \bar{y}^i \ge \varepsilon \, , \sigma_j^i \ge \sigma \right\} \tag{8.2.69}$$

when M_g, ε, and σ are once again designer-specified positive constants. Now, define a so-called *minimum approximation error*

$$w = \left(\hat{f}(\mathbf{x}|\boldsymbol{\theta}_f^*) - f(\mathbf{x}) \right) + \left(\hat{g}(\mathbf{x}|\theta_f^*) - g(\mathbf{x}) \right) u_c \tag{8.2.70}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

and rewrite the error equation (8.2.59) as

$$\dot{\mathbf{e}} = \mathbf{A}_c\mathbf{e} + \mathbf{b}_c g(x)u_s + \mathbf{b}_c\Big\{\Big[\hat{f}\big(\mathbf{x}|\theta_f\big) - \hat{f}\big(\mathbf{x}|\theta_f^*\big)\Big] + \Big[\hat{g}\big(\mathbf{x}|\theta_g\big) - \hat{g}\big(\mathbf{x}|\theta_g^*\big)\Big]u_c + w\Big\} \tag{8.2.71}$$

Assuming $\hat{f}(\cdot)$ *and* $\hat{g}(\cdot)$ to be fuzzy logic systems of the form in Equation (8.2.64), then (8.2.71) can be rewritten as

$$\dot{\mathbf{e}} = \mathbf{A}_c\mathbf{e} - \mathbf{b}_c g(\mathbf{x})u_s + \mathbf{b}_c w + \mathbf{b}_c\Big[\phi_f^T h(\mathbf{x}) + \phi_g^T h(\mathbf{x})u_c\Big] \tag{8.2.72}$$

where $\phi_f = \theta_f - \theta_f^*$, $\phi_g = \theta_g - \theta_g^*$, $h(\mathbf{x})$ is the fuzzy basis function, defined by Equation (7.3.65). Consider, next, a Lyapunov function

$$v_a = \tfrac{1}{2}\mathbf{e}^T\mathbf{P}\mathbf{e} + \frac{1}{2d_f}\phi_f^T\phi_f + \frac{1}{2d_g}\phi_g^T\phi_g \tag{8.2.73}$$

where d_f and d_g are positive constants. Then the time derivative of v_a is given by

$$\dot{v}_a = -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} - g(\mathbf{x})\mathbf{e}^T\mathbf{P}\mathbf{b}_c u_s + \mathbf{e}^T\mathbf{P}\mathbf{b}_c w + \frac{1}{d_1}\phi_f^T\Big[\dot{\theta}_f + d_1\mathbf{e}^T P\mathbf{b}_c\mathbf{h}(x) + \frac{1}{d_2}\phi_g^T\big[\dot{\theta}_g + d_2 e^T Pb_c h(x)u_c\big]\Big] \tag{8.2.74}$$

in lieu of (8.2.56) and the fact that θ_f^* and θ_g^* are time invariant. Noting Equation (8.2.62) and the fact that $g(\mathbf{x}) > 0$, it follows that the term $g(\mathbf{x})$ $\mathbf{e}^T P\mathbf{b}_c u_s$ in Equation (8.2.74) is nonnegative. Now, if we choose the following adaptation laws:

$$\dot{\theta}_f = -d_f\mathbf{e}^T\mathbf{P}\mathbf{b}_c\mathbf{h}(\mathbf{x}) \tag{8.2.75}$$

$$\dot{\theta}_g = -d_g\mathbf{e}^T\mathbf{P}\mathbf{b}_c\mathbf{h}(\mathbf{x})u_c \tag{8.2.76}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

then the relation (8.2.74) is reduced to

$$\dot{\upsilon}_e \leq -\tfrac{1}{2}\mathbf{e}^T Q\mathbf{e} + \mathbf{e}^T P\mathbf{b}_c w \tag{8.2.77}$$

This expression is still not in the final form due to the second term $\mathbf{e}^T P\mathbf{b}_c w$. However, we do know that because of w this term is of the order of the minimum approximation error. If f and g are within the search spaces for $\hat{f}$ and $\hat{g}$, then $w = 0$ and Equation (8.2.77) is reduced to $\dot{\upsilon}_e = -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} \leq 0$. Wang has presented a theorem which shows that $\lim_{t\to\infty}|e(t)| = 0$ (Wang, 1994a).

The above development assumes that $\hat{f}$ and $\hat{g}$ are in the forms similar to Equation (8.2.64). If they are in the form of (8.2.51), then one may use a Taylor's series expansion of $\hat{f}$ and $\hat{g}$ about θ_f^* and θ_g^*, respectively and follow the same procedure.

The final problem is to guarantee that the constraints $\theta_f \in \Omega_f$ and $\theta_g \in \Omega_g$. To achieve this one can use the parameter projection algorithm (Luenberger, 1984), i.e., if the adaptation parameter vectors θ_f and θ_g are within the sets Ω_f and Ω_g, respectively and moving inward, then use Equations (8.2.75) and (8.2.76) as the adaptation laws. On the other hand, if the vectors are on the boundaries and moving outward, use the parameter projection algorithm to modify the laws (8.2.75) and (8.2.76) such that the parameter vectors will remain inside the constraint sets. Details of this can be seen in Wang (1994a). The indirect adaptive fuzzy control system is summarized in Figure 8.18. Shown here are all the three basic elements of the design:

(i) low-level fuzzy controller, u_c,
(ii) adaptation laws, and
(iii) stabilizing supervisory controller, u_s.

Wang (1994a) has proposed a very similar development as above for *direct adaptive fuzzy* control systems. The main difference here is that in the direct approach a set of rules relating output variables x, y, or e and the control variables $u(x)$, $u(y)$, or $u(e)$ are available instead of those for the plant itself.

controlengineers.ir
@controlengineers

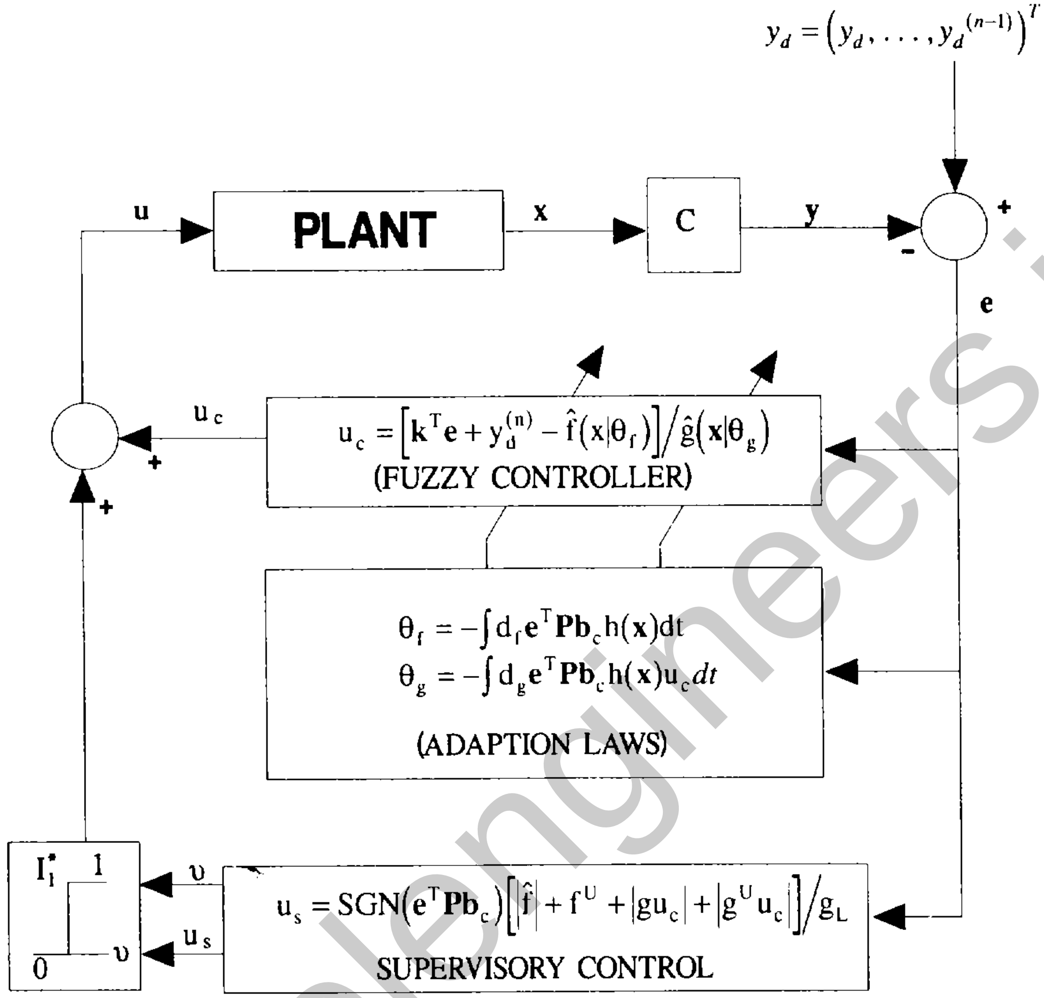


Figure 8.18 The indirect adaptive fuzzy control system.

Example 8.2.5 Consider the two-dimensional nonlinear model of the inverted pendulum, described by Equation (7.2.19) and use $m = \cancel{100}$ 150 grams, $M = \cancel{1000}$ 1500 grams, $l = 1/2$ meter, $g = 9.8$ m/sec and a desired angular position $\theta_d(t) = \frac{\pi}{30}$ ~~cos t~~ sin t. It is further assumed that no linguistic information (IF-THEN rules) is available about functions $f(\cdot)$ and $g(\cdot)$ in Equations (7.2.19). Let $|x_i| < \pi/6$ for $i = 1, 2$. Assume that each state is divided into five membership functions as follows:

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\mu_{F_j^1}(x_j)=\exp\left[-\left(\frac{x_j+\pi/6}{\pi/24}\right)^2\right],$$

$$\mu_{F_j^2}(x_j)=\exp\left[-\left(\frac{x_j+\pi/12}{\pi/24}\right)^2\right],\quad \mu_{F_j^3}(x_j)=\exp\left[-\left(\frac{x_j}{\pi/24}\right)^2\right],$$

$$\mu_{F_j^4}(x_j)=\exp\left[-\left(\frac{x_j-\pi/12}{\pi/24}\right)^2\right],\ \text{and}$$

$$\mu_{F_j^5}(x_j)=\exp\left[-\left(\frac{x_j-\pi/6}{\pi/24}\right)^2\right]$$

over the interval $[-\pi/6, \pi/6]$. It is desired to find a stabilizing indirect adaptive fuzzy controller for this system.

SOLUTION: To apply the indirect adaptive control scheme, let us first find the bounds on f and g as described in Equation (7.2.19).

$$|f(\mathbf{x})|=\left|\frac{g\sin x_1-\dfrac{m\ell x_2^2\cos x_1\sin x_1}{m+M}}{\ell\left(\dfrac{4}{3}-\dfrac{m\cos^2 x_1}{m+M}\right)}\right|\le$$

$$\frac{9.8+0.023x_2^2}{0.62}=15.8+0.037x_2^2\triangleq f^U(\mathbf{x}) \tag{8.2.78}$$

$$|g(\mathbf{x})|=\left|\frac{\cos x_1\dfrac{1}{m+M}}{\ell\left(\dfrac{4}{3}-\dfrac{m\cos^2 x_1}{m+M}\right)}\right|\le\frac{1}{1.1\left(\dfrac{2}{3}-\dfrac{0.05}{1.1}\right)}=1.46\triangleq g^U(\mathbf{x}) \tag{8.2.79}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Since $|x_1| \le \pi/6$, then

$$|g(\mathbf{x})| \ge \frac{\cos\pi/6}{1.1\left(\frac{2}{3} + \frac{0.05}{1.1}\cos^2\pi/6\right)} = 1.12 \triangleq g_L(\mathbf{x}) \tag{8.2.80}$$

Assume the constraints

$$|x_1| < \frac{\pi}{6} \quad \text{and} \quad |u| \le 180 \tag{8.2.81}$$

need to be satisfied. By Theorem 8.1 of Wang (1994a), we have

$$|\theta_f(t)| \le M_f, \quad |\theta_g(t)| \le M_g$$

all elements of $\theta_g \ge \varepsilon$,

$$|\mathbf{x}(t)| \le |\mathbf{y}_d| + \left(\frac{2\bar{V}}{\lambda_{\min}}\right)^{\frac{1}{2}} \tag{8.2.82}$$

and

$$|u(t)| \le \frac{1}{\varepsilon}\left(M_f + \left|y_d^{(n)}\right| + |\mathbf{k}|\left(\frac{2\bar{V}}{\lambda_{\min}}\right)^{\frac{1}{2}}\right)$$
$$+ \frac{1}{g_L(\mathbf{x})}\left[M_f + \left|f^U(\mathbf{x})\right| + \frac{1}{\varepsilon}\left(Mg + g^U\right)\left(M_f + \left|y_d^{(n)}\right| + |\mathbf{k}|\left(\frac{2\bar{V}}{\lambda_{\min}}\right)^{\frac{1}{2}}\right)\right] \tag{8.2.83}$$

for all $t \ge 0$, where $\lambda_{\min}$ is the minimum eigenvalue of Matrix $\mathbf{P}$ in Equation (8.2.56) and $\mathbf{y}_d = \left(y_d \; \dot{y}_d \; . \; . \; . \; y_d^{(n-1)}\right)^T$. We will need to use the above equations to evaluate the parameters $\bar{V}$ and $\lambda_{\min}$ such that design constraints (8.2.81) are satisfied. Since $|y_d| \le \pi/30$, if one determines $\bar{V}$ and $\lambda_{\min}$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

such that $\left(\frac{2\bar{V}}{\lambda_{\min}}\right)^{\frac{1}{2}} \leq 2\pi/15$, then by Equation (8.2.82) we have $|\mathbf{x}| \leq \pi/30 + 2\pi/15 = \pi/6$. Since we have two constraints in Equation (8.2.81), but six design parameters in $\bar{V}$, k_1, k_2, ε, M_f, and M_g, then we have freedom to specify four of them. Let us begin by choosing $k_1 = 1$ and $k_2 = 2$ such that $s^2 + 2s + 1$ is Hurwitz. Let $Q = 10I_2$, where I_2 is a 2×2 identity matrix, then the solution of the Lyapunov equation in (8.2.56) is given by $\mathbf{P} = \begin{pmatrix} 15 & 5 \\ 5 & 5 \end{pmatrix} > 0$. Thus, $\lambda_{\min} = 2.93$. If we choose $\bar{V} = \frac{\lambda_{\min}}{2}\left(\frac{2\pi}{15}\right)^2 = 0.267$, then the constraint on $|\mathbf{x}|$ can be satisfied by virtue of Equation (8.2.82). Using Equation (8.2.83) and constraint on $|u|$ from Equation (8.2.81), we can choose one of the parameters M_f, ε, and M_g. Let $M_f = 16$, $M_g = 1.6$, then by straight calculation $\varepsilon = 0.7$. With the following choices:

$$M_f = 16,\ M_g = 1.6,\ \varepsilon = 0.7$$
$$\bar{V} = 0.267,\ \mathbf{k} = (2 \quad 1)^T \ \textit{and}\ Q = 10I_2$$

the two constraints of Equation (8.2.81) are satisfied.

Next, since the relative bounds of f and g in Equations (8.2.78) and (8.2.79) are such that $|f| >> |g|$, then a set of parameters $d_f = 50$ and $d_g = 1$ are chosen. Figures 8.19 and 8.20 show the results of the simulation for the indirect adaptive fuzzy control of the inverted pendulum. These results differ from those of Wang (1994a) since we chose a slightly different $y_d(t)$. The initial state $\mathbf{x}(0)$ was chosen to be $\mathbf{x}(0) = (\pi/60 \quad 0)^T$. The results look fairly reasonable and in accord with those obtained by Wang (1994a).

8.3 Large-Scale Fuzzy Control Systems

Thus far in this chapter, three important aspects of fuzzy control systems have been considered. These were fuzzy control architectures, adaptive fuzzy control systems, and stability of fuzzy control systems. In this section, the notion of hierarchy is studied for fuzzy control systems. An attempt is made to reduce the size of the inference engine for a large-scale system.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

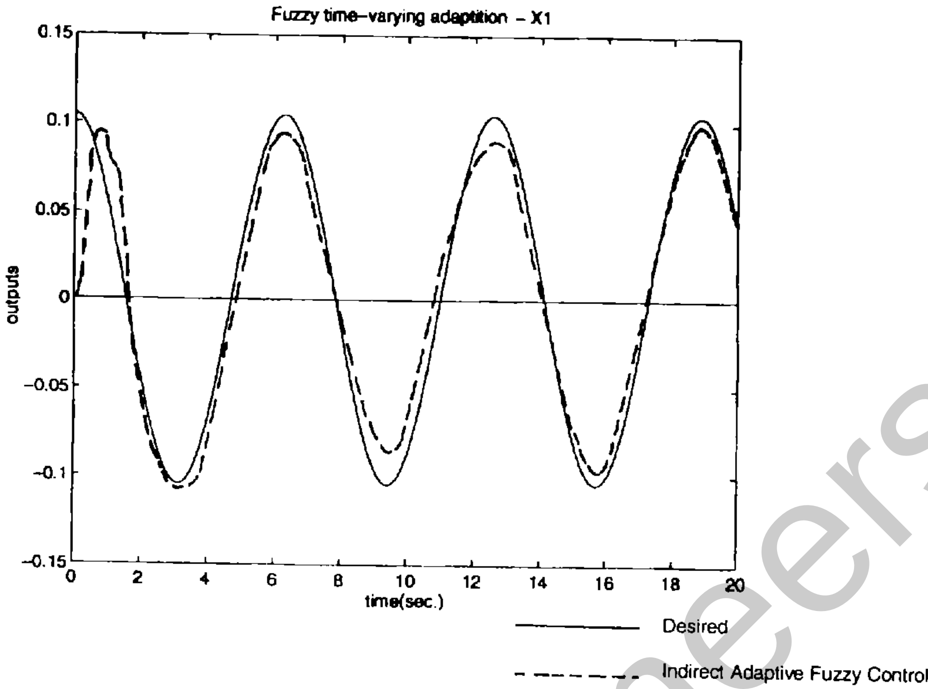


Figure 8.19 X1 results of the simulation for the indirect adaptive fuzzy control of the inverted pendulum.

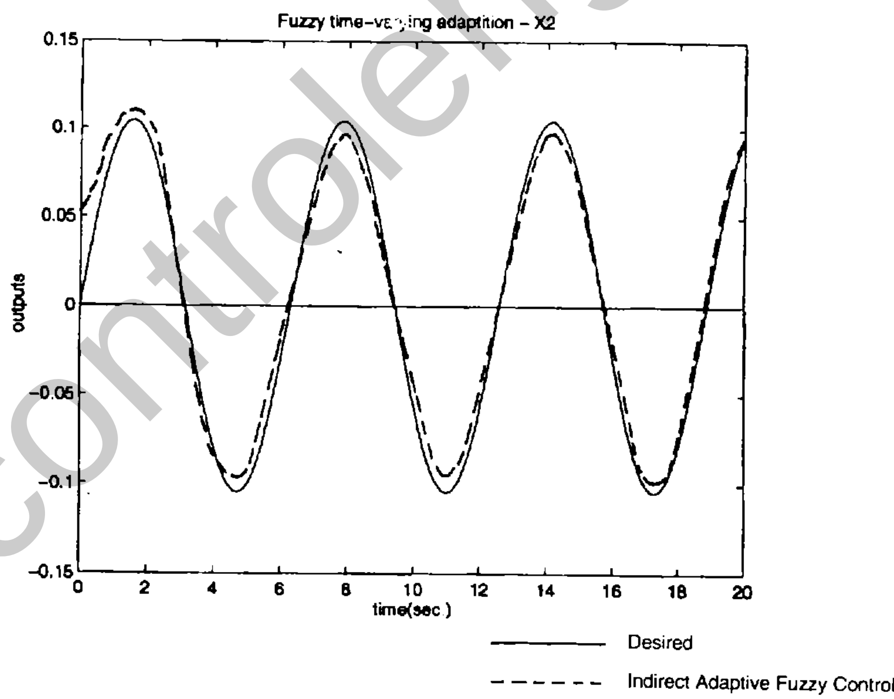


Figure 8.20 X2 results of the simulation for the indirect adaptive fuzzy control of the inverted pendulum.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

8.3.1 Hierarchical Fuzzy Control

When a fuzzy controller is designed for a large-scale system, often several measurable output and actuating input variables are involved. In addition, each variable is represented by a finite number l of linguistic labels which would indicate that the total number of rules is equal to l^n, where n is the number of system variables. Clearly, even for a very low-order system with as few as five labels per variable, this exponential expression would become large. In this section, a hierarchical scheme by Raju *et al.* (1991) is presented.

Consider a fuzzy controller with n rules of the following type:

$$\text{IF } y_1 \text{ is } A_{1i} \text{ and } y_2 \text{ is } A_{2i} \text{ and } \ldots \text{ and } y_n \text{ is } A_{ni} \text{ THEN } u_1 \text{ is } B_1$$
$$\text{for } i = 1, 2, \ldots, n$$

where y_i, $i = 1, \ldots, n$ are the system's output variables (a subset of all variables), u_i, $i = 1, \ldots, n$ are the system's control variables, A_{ij} and B_j, $i, j = 1, \ldots, n$ are fuzzy sets similar to those discussed in Chapter 7 thus far and this chapter, i.e., *NB, NM, NS, ZO, PS, PM*, and *PB* to stand for negative big, negative medium, ... , and positive big. Now, consider the following theorem.

Theorem 8.1 *For a fuzzy system with n variables and m fuzzy sets (labels) per variable, the total number of rules is given by* $k = m^n$.

PROOF: For $n = 1$, the number of rules is obviously m. Suppose that the theorem is true for $n = n_1$ and let $n = n_1 + 1$, then for every rule with n_1 variables, there are m different rules if one adds one variable. In other words, every rule (of m^{n_1} such rules) with n_1 variables will change into m rules with $n_1 + 1$ variables. Thus, the total number of rules k is given by

$$k = m \times m^{n_1} = m^{(n_1+1)} = m^n \tag{8.3.1}$$

This proves the theorem. Q.E.D. ■

Example 8.3.1 Consider a fourth-order inverted model of Example 7.2.3, where $n = 4$ and $m = 5$. It is used to find the total number of fuzzy rules.

SOLUTION: According to Theorem 8.1 and Equation (8.3.1), we need a total of $k = m^n = 5^4 = 625$. If there were five variables, then $k = 3125$.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

From the above simple example, it is very clear that the application of fuzzy control to any system of significant size would result in a "curse of dimensionability," much like standard dynamic programming. This exponential explosion of the size of the rule base can be handled in a variety of ways:

1) Fuse sensory variables before feeding them to the inference engine, thereby reducing the size of the inference engine.

2) Group the rules in prioritized levels to design a hierarchical fuzzy controller.

3) Reduce the size of the inference engine directly using notions of passive decomposition of fuzzy relations.

4) Decompose a large-scale system into a finite number of reduced-order subsystems, thereby eliminating the need for a large-sized inference engine.

The first scheme is a hardware or implementation issue and will be discussed in Section 8.3.2. In this section, the second option is first considered and then other schemes will be given later on.

In the proposed hierarchical fuzzy control structure, the first-level rules are those related to the most important variables and are gathered to form the first-level hierarchy. The second most important variables, along with output of the first-level, are chosen as inputs to the second-level hierarchy, and so on. The first and the ith rule of the hierarchically categorized sets are given by

IF y_1 is A_{11} and … and y_{n_1} is A_{1n_1} THEN u_1 is B_1

$\vdots$

IF y_{N_i+1} is A_{N_i1} and … and $y_{N_i+n_i}$ is $A_{N_in_i}$ and u_{i-1} is B_{i-1} THEN u_i is B_i

where $N_i = \sum_{j=1}^{i-1} n_j \le n$ and n_j is the number of jth level system variables used as inputs. The following theorems, due to Raju *et al.* (1991), evaluate the number of overall rules for a hierarchical fuzzy controller and the minimum number of rules.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Theorem 8.2 *Consider a hierarchical fuzzy controller with L levels of rules, n system variables, and n_i variables contained in the ith level including the output of the $(i-1)$th level for $i > 1$. Then the total number of rules is given by*

$$k = \sum_{i=1}^{L} m^{n_i} \tag{8.3.2}$$

where m is the number of fuzzy sets and

$$n_1 + \sum_{i=2}^{L} (n_i - 1) = n \tag{8.3.3}$$

PROOF: For the ith level, we have n_i variables. Thus, by Theorem 8.1, the total number of rules for all levels is m^{n_i}. The overall number of rules is given as a sum of the rules at all levels, i.e.,

$$k = \sum_{i=1}^{L} m^{n_i} \tag{8.3.4}$$

which proves the theorem. Q.E.D. ■

If there is a constant number of rules at each level, i.e., $n_i = a$ = a constant for $i = 1, \ldots, L$, then by (8.3.4), one has

$$a + \sum_{i=2}^{L} (a-1) = a + (a-1)(L-1) = n$$

Solving for L, one has

$$L = 1 + (n-a)/(a-1) \tag{8.3.5}$$

Hence, the total number of fuzzy rules, using Equations (8.3.2) and (8.3.5), will be

$$k = \sum_{i=1}^{L} m^a = Lm^a = [1 + (n-a)/(a-1)]m^a \tag{8.3.6}$$

which indicates that through this hierarchical structure, the total number of fuzzy rules is now a linear function of the number of system variables and not exponential. The next theorem determines the minimum number of rules.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Theorem 8.3 *For a hierarchical fuzzy control structure with n variables, if m and n_i satisfy conditions $m \geq 2$ and $n_i \geq 2$, the total number of rules in the rule set will take on its minimum value when $n_i = a = 2$ and on its maximum when $n_i = n_1 = n$.*

PROOF: Let n_i variables be involved at the ith level and assume that $n_i \geq 3$. By virtue of Theorem 8.1, the number of rules at ith level is m^{n_i}. Now, assume that one splits this level into two sublevels containing 2 and $n_i - 1$ variables, respectively. Note that the output from the previous level is also counted here. Now, the total number of rules for both sublevels combined is

$$k_{i2} = m^2 + m^{n_i - 1} \tag{8.3.7}$$

Since $n_i \geq 3$ or $n_i - 1 \geq 2$, it follows that $m^2 \leq m^{n_i - 1}$ and hence

$$m^2 + m^{n_i - 1} \leq 2m^{n_i - 1} \tag{8.3.8}$$

Since it is assumed that $m \geq 2$, then

$$2m^{n_i - 1} \leq m^{n_i} \tag{8.3.9}$$

Now, considering this, we have

$$m^2 + m^{n_i - 1} \leq m^{n_i}$$

Thus, the total number of rules in the hierarchical structure would be reduced if one would split any level with three or more variables into two sublevels, one of which has two variables. If this process is repeated for all levels, it is asserted that the total number of rules will reach its minimum if every hierarchical level would contain only two variables (see Figure 8.21).

Now let the number of variables be n_i and n_j at their respective levels with the assumption that $n_i \geq n_j$, then these two levels would have

$$k_{ij} = m^{n_i + n_j - 1} \tag{8.3.10}$$

rules. The following inequality would then follow:

$$m^{n_i} + m^{n_j} \leq m^{n_i} + m^{n_i} = 2m^{n_i} \tag{8.3.11}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

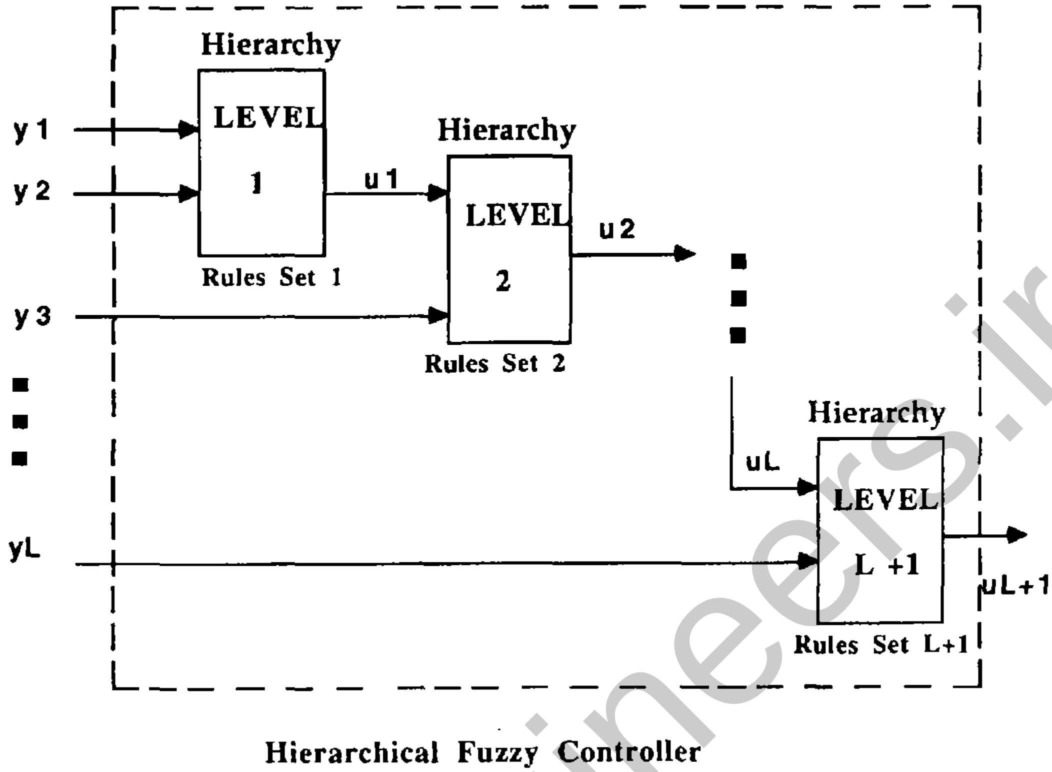


Figure 8.21 A schematic representation of a hierarchical fuzzy controller.

In a similar fashion one infers that $n_j \geq 2$, $n_j - 1 \geq 1$ and $m \geq 2$, hence

$$2m^{n_i} \leq m \cdot m^{n_i} \leq m^{n_i + n_j - 1} \tag{8.3.12}$$

Utilizing (8.3.10)–(8.3.12), it follows that

$$m^{n_i} + m^{n_j} \leq m^{n_i + n_j - 1} = k_{ij}$$

Thus, it is concluded that the total number of rules would decrease if one begins combining two levels into one. Repetition of this process would eventually lead to the maximum number of rules when $n_i = n_1 = n$. This proves the theorem. Q.E.D. ■

Example 8.3.2 In Example 7.2.5, the fuzzy control of the inverted pendulum was simulated on FULDEK. It is desired to make a comparison of the results and assess the fuzzy rules reduction.

SOLUTION: The resulting states (x_1 and x_2) and control (u) trajectories

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

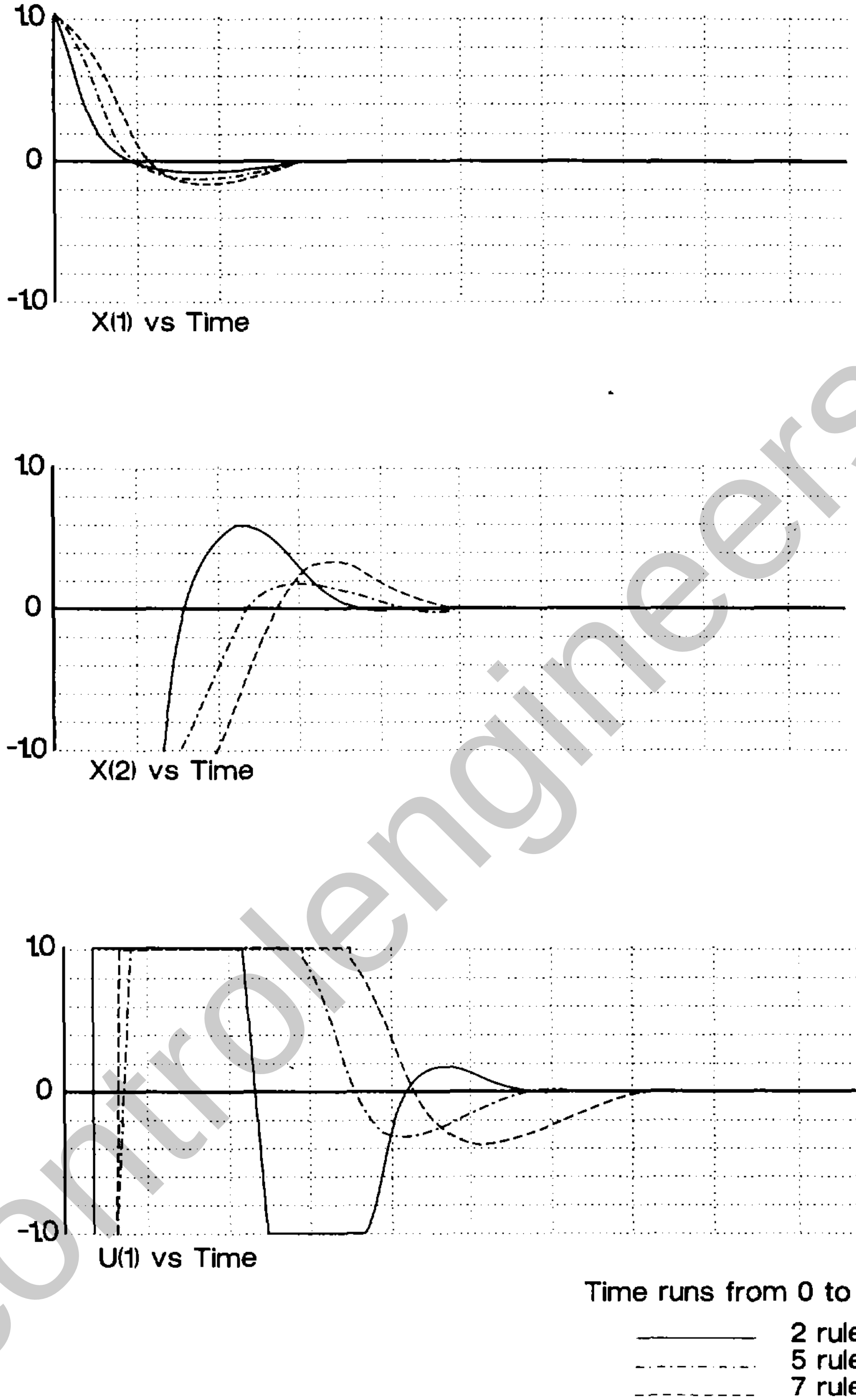


Figure 8.22 The states and control trajectories for three sets of fuzzy rules from a FULDEK simulation.

controlengineers.ir

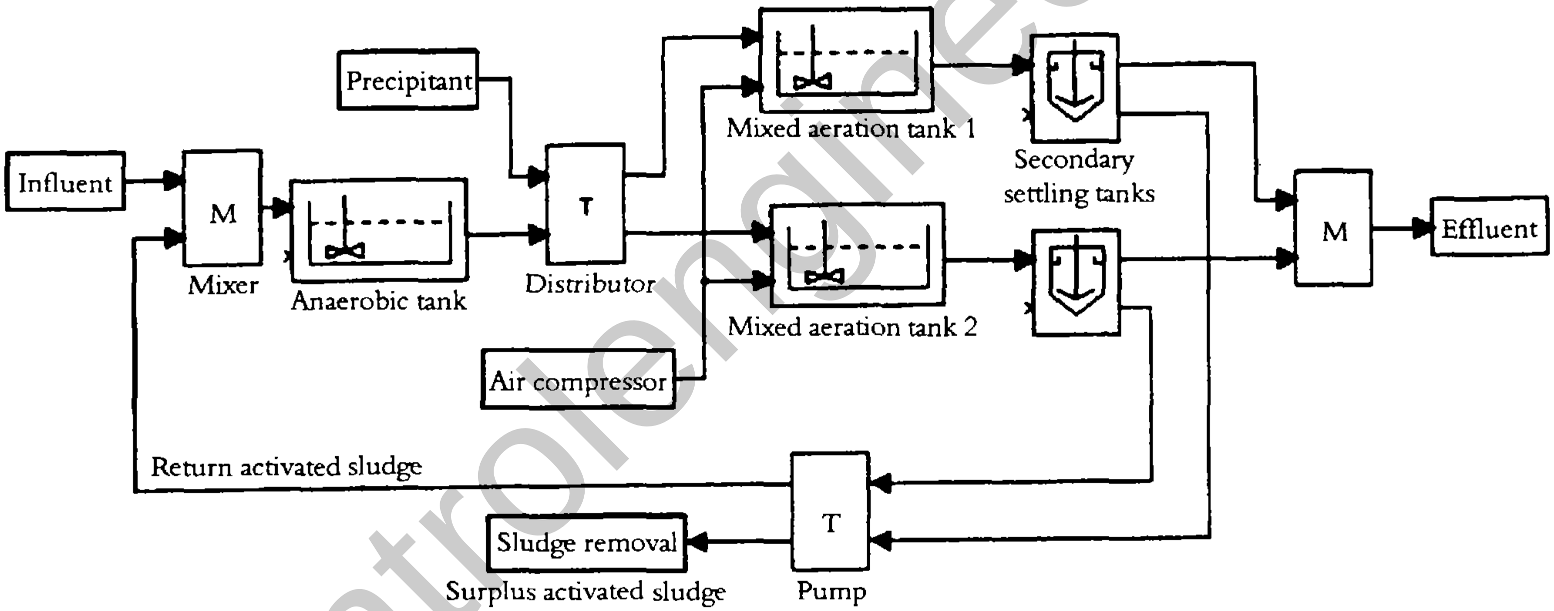


Figure 8.23 A waste water treatment plant.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

for three sets of fuzzy rules directly from a FULDEK simulation are shown in Figure 8.22. The state trajectories for $x_1 = \theta(t)$ show that the three cases with two, five, and seven rules are not a great deal apart. The addition of more rules, the plots indicate, has resulted in slower responses. This is even more evident from the responses of the angular velocity $x_2 = \dot{\theta}(t)$ as shown in the same figure. In fact, a close inspection of three x_2 plots indicates that the case of five rules has less overshoot than both two and seven rule cases. This simple numerical experiment seems to confirm that more rules are not necessarily going to improve the quality of the system's response. What is critical, is to have rules which involve important and dominant variables and their proper combinations involved in the fuzzy rules. On the other hand, reducing the size of the inference engine should not be at the cost of drastic performance degradation. The three plots of $u(t)$ versus time indicates that the level of control effort would increase with more rules at the same time that the performance is degraded.

Example 8.3.3 Consider a waste water treatment plant shown in Figure 8.23 whose dynamic plant model is shown below:

$$\dot{x} = \begin{bmatrix} -1.82 & 0 & 0 & 0 \\ 0 & -40 & 0 & 0 \\ 0 & 0 & -1.7 & 0 \\ 0 & 0 & 0 & -46 \end{bmatrix} x + \begin{bmatrix} 10 \\ 10 \\ 01 \\ 01 \end{bmatrix} u$$

$$y = \begin{bmatrix} 17.5 & -17.5 & 0 & 0 \\ 0 & 0 & 13.4 & -13.4 \end{bmatrix} x$$

where y_1 is nitrogen concentration, y_2 is ammonia concentration, u is the effluent oxygen concentration, and vector y and vector x represent a linear combination of nitrogen and ammonia concentrations and their derivative, chosen arbitrarily as state variables. It is desired to design a fuzzy controller for this system which minimizes the untreated overflows, stabilizes the treatment process and effluent quality, and minimizes the total pollution.

SOLUTION: The objective of this work has been to develop a control strategy to run a biological waste water plant so that it would guarantee good cleaning performance in the face of flow and load pattern variations. A block diagram of the fuzzy control system is shown in Figure 8.24. Assuming that the desired state is a constant vector x_d and defining

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

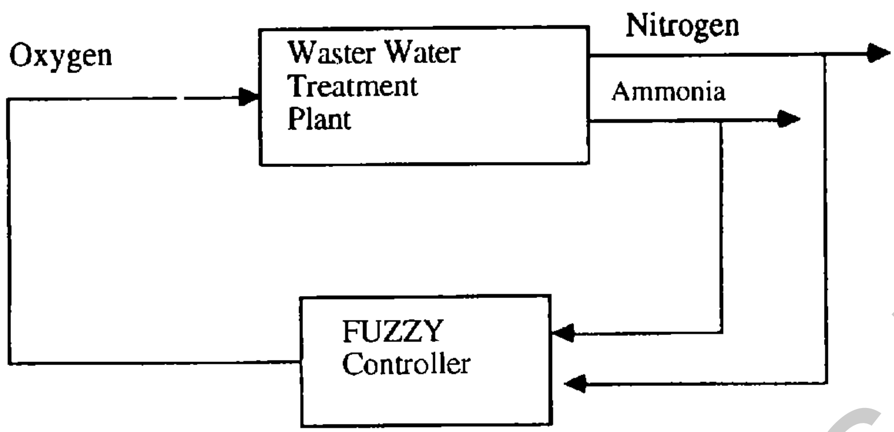


Figure 8.24 A fuzzy controller for a waste water treatment plant.

error vector $e = x - x_d$, one has the following error equation

$$\dot{e} = Ae + Bu + Ax_d$$

The membership functions for nitrogen (e_N) and ammonia (e_A) concentration errors were chosen as triangular functions with negative, zero, and positive linguistic labels. For oxygen (e_O), only zero and positive labels were chosen. The following rules were chosen for the fuzzy inference engine:

R1: IF e_N is N THEN e_O is P
R2: IF e_N is Z THEN e_O is Z
R3: IF e_N is P THEN e_O is Z
R4: IF e_A is N THEN e_O is P
R5: IF e_A is Z THEN e_O is Z
R6: IF e_A is P THEN e_O is Z

It is noted that the above six rules can be implemented separately as two, three-rule sets, or they can be combined (R1 + R4, R2 + R5, and R3 + R6) to reduce them to three rules such as:

IF e_N is N or e_A is N THEN e_O is P
IF e_N is Z or e_A is Z THEN e_O is Z
IF e_N is P or e_A is P THEN e_O is N

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

It is usually a good practice to reexamine the rule set before composing them on a software environment. For this case, FULDEK was used with two separate sets of rules for nitrogen and ammonia. Figure 8.25 shows the simulation results. The concentrations of both chemical concentration errors reduce to zero in 0.8 of a second.

Example 8.3.4 This example is concerned with the control of a steam drum by Raju *et al.* (1991), considered in Figure 8.26. The control problem is to maintain the water level in the steam drum at a desired level. It is desired to design a hierarchical fuzzy controller for the system.

SOLUTION: The closed-loop hierarchical fuzzy control system is shown in Figure 8.27. The dynamic model of the system is given by

$$\dot{x} = Ax + B_o u_o + B_d u_d$$

where x is an 18th-order state vector, u_o and u_d are controller action and disturbance input, respectively. A, B_o, and B_d are matrices of appropriate dimensions. The detailed model of this system can be obtained in Berkan and Upadhyaya (1988). All the system variables and output are normalized between [-1,1] as a uniform universe of discourse. The membership functions were chosen to be triangular type from large negative (LN), medium negative (MN), SN, ZN, Z, ZP (zero positive), SP, medium positive (MP), and large positive (LP).

The hierarchical fuzzy rule set has two important classes of rules: *proportional* type and *integral* type. Examples of these rules are given

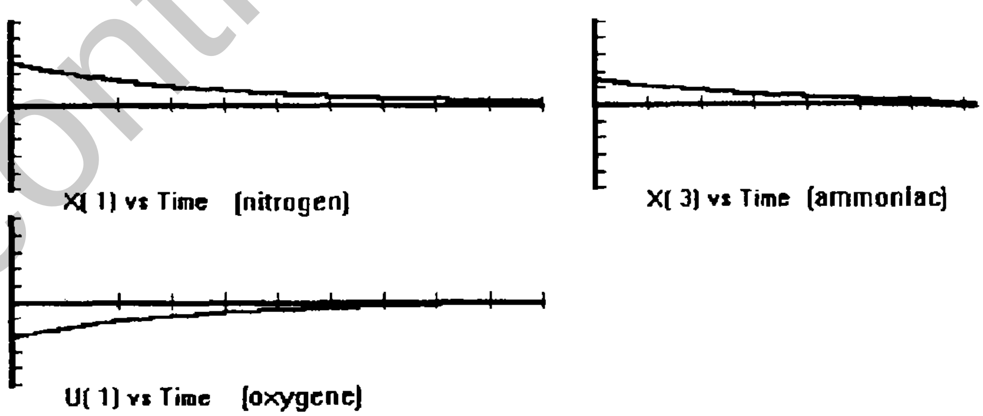


Figure 8.25 FULDEK simulation results of the waste water treatment system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

below:

IF x_1 is a_{1k} and x_2 is a_{2k} THEN the output is b_k

IF x_1 is a_{1j} and x_2 is a_{2j} THEN increase the output by b_j

Raju *et al.* (1991) categorize the rules into several levels involving the basic structure of Figure 8.21. Figure 8.28 shows the results of simulation of the hierarchical fuzzy control of the steam drum system. The simulated variables are all expressed in the form of an error (deviation) form from

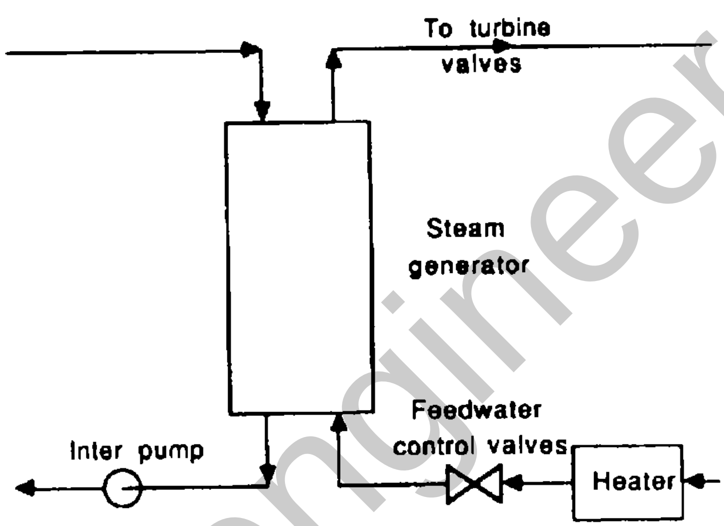


Figure 8.26 A steam drum system.

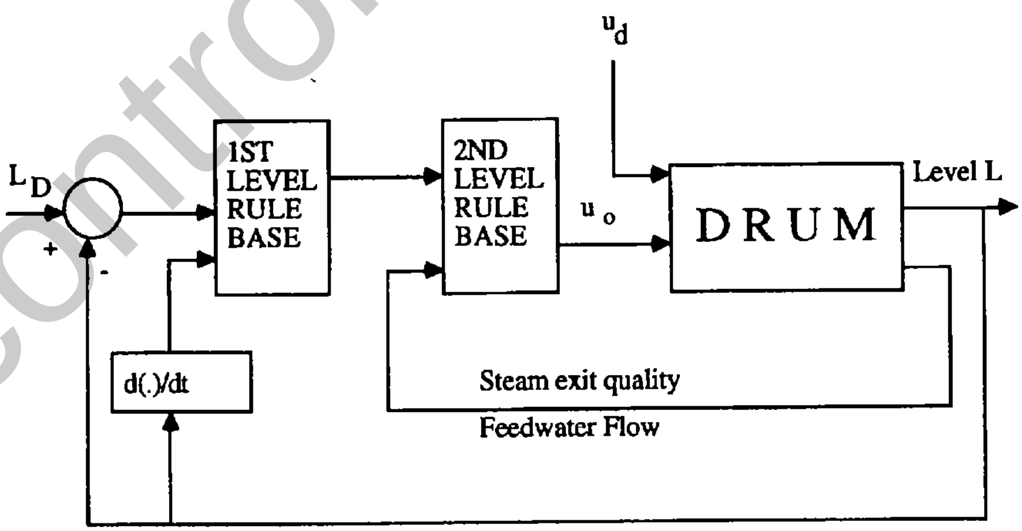
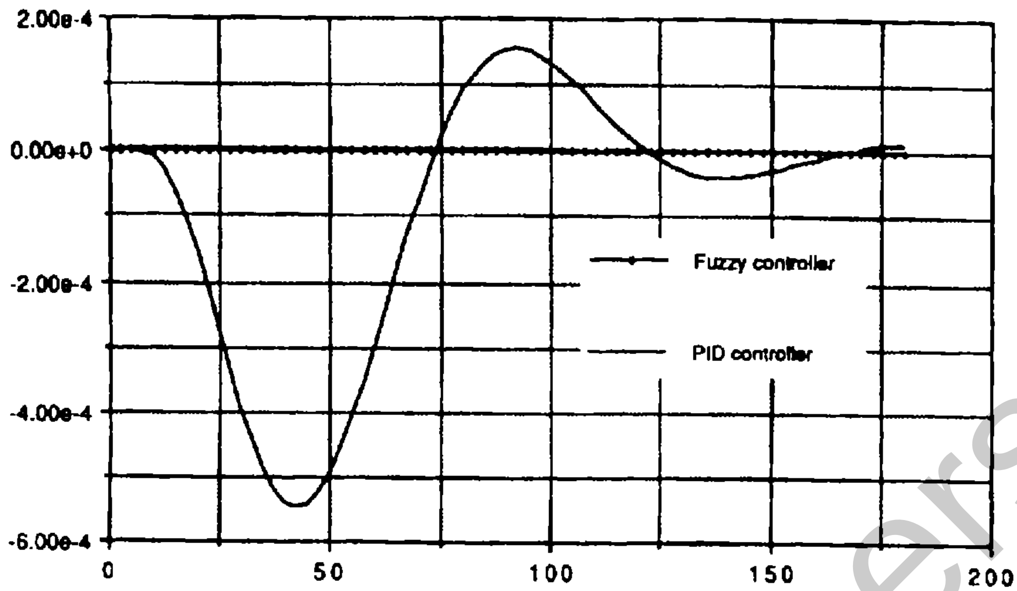


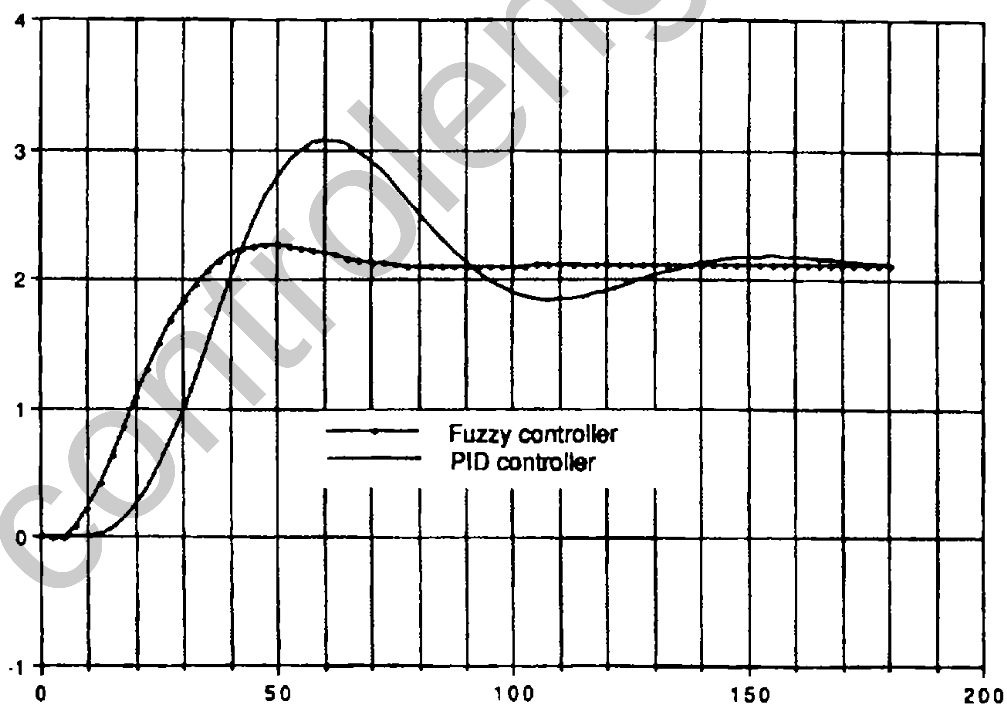
Figure 8.27 The hierarchical fuzzy control architecture for a steam drum system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



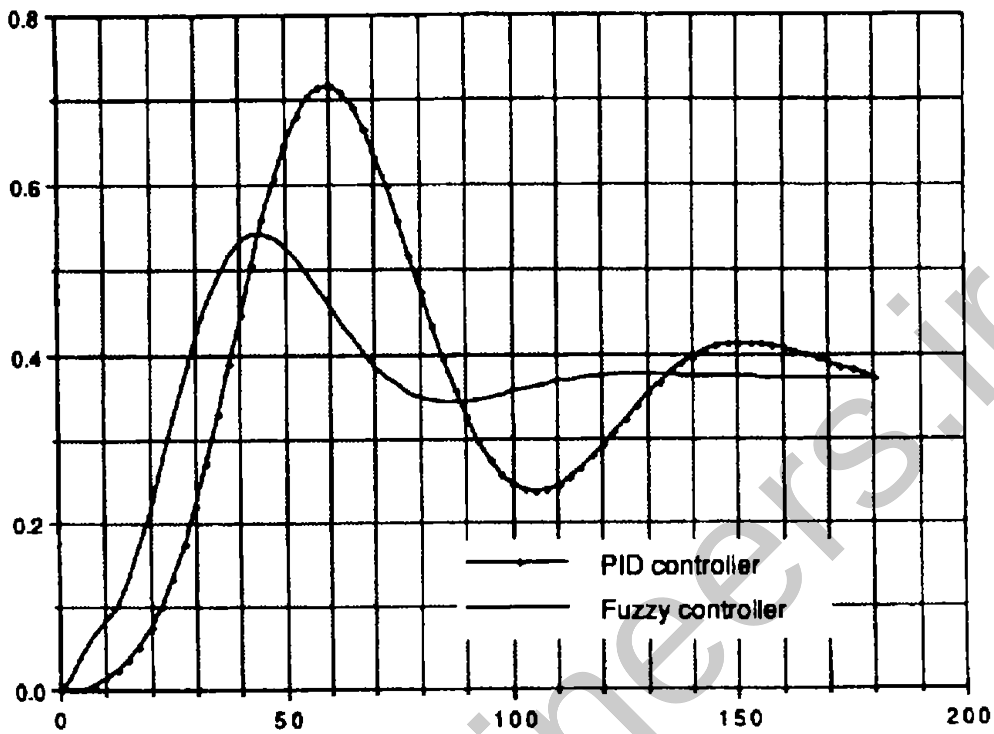
(a) Drum level for a 10% steam valve opening perturbation.



(b) Feedwater flow for a 10% steam valve opening perturbation.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(c) Feedwater flow for a 10% feedwater temperature perturbation.

Figure 8.28 Simulation results of steam drum system (Raju *et al.*, 1991).

the desired values. The resulting fuzzy control has also been compared with a PID controller. The former controller has displayed a better performance, i.e., lower rise time, less overshoot, and much reduced error variation.

Raju and Zhou (1993) have also applied the notion of a hierarchical fuzzy inference engine to an adaptive formulation in which the rules are updated by a supervisory rule set. This approach has been applied to the steam drum problem of Example 8.3.4.

8.3.2 Rule-Base Reduction

As it was discussed in the previous section, the number of rules grows exponentially with respect to the number of the system's sensory feedback variables. Although the hierarchical structure suggested by Raju *et al.* (1991) does reduce the number of rules considerably, it is still not computationally effective. Consider, as an example, the all too familiar inverted pendulum problem of Section 7.2.5. For the fourth-order nonlinear (or linear) model, the number of rules reduced from $5^4 = 625$ to $3(5)^2 =$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

75. Clearly, this number of rules is still too many and unacceptable. Moreover, this approach is not of any use for two-variable systems, on one hand, and the intermediate variables $u_1, u_2, \ldots, u_L$ (see Figure 8.21) are not fuzzy sets of the controller input sensory variables. They represent an aggregate of fuzzy decisions whose contributions into higher-level rules cannot be obtained readily.

Here, the hierarchical structure of the previous section is combined as one layer of a systematic procedure to drastically reduce the number of rules, i.e., the size of the overall rule base or the inference engine.

Sensory Fusion

An alternative approach is to investigate the physical possibility and feasibility of combining or fusing sensory signals (variables) *before* being fed to the fuzzy controller (inference engine). Assume that a fuzzy controller has three inputs (y_i, $i = 1, 2, 3$) and one output (u). Furthermore, let each variable (input or output) be represented by five linguistic variables:

R1: IF y_1 is A_1 and y_2 is B_1 and y_3 is C_1 THEN u is D_1
R2: IF y_1 is A_2 and y_2 is B_2 and y_3 is C_2 THEN u is D_2
⋮
R125: IF y_1 is A_5 and y_2 is B_5 and y_3 is C_5 THEN u is D_5

Now, one would look into combining the sensory data (variables y_i, $i = 1, 2, 3$) in one of these four possible ways:

(i) All three variables fused

$$Y = ay_1 + by_2 + cy_3$$

where a, b, and c are positive parameters dictated by physical consideration and designer's experience.

(ii) Variables 1 and 2 are fused

$$Y_1 = ay_1 + by_2$$
$$Y_2 = y_3$$

(iii) Variables 1 and 3 are fused

$$Y_1 = ay_1 + by_3$$
$$Y_2 = y_2$$

(iv) Variables 2 and 3 are fused

$$Y_1 = ay_2 + by_3$$
$$Y_2 = y_1$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

In this manner the number of fuzzy rules would be reduced from 125 in the unfused (general) case to $k_1 = 5$ or $k_i = 25$, $i = 2, 3$, and 4 depicting the above four cases. In this way a remarkable reduction in the size of the rule base would result. In fact, the reduction has a lower bound if *all* variables could, somehow, be fused. The rules for the first case are shown below:

R1: IF Y is A_1 THEN u is D_1

⋮

R5: IF Y is A_5 THEN u is D_5

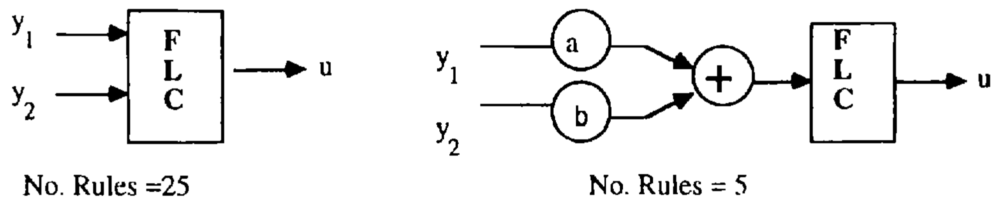
If every two variables could be combined, then for an even number of variables, the reduction is even more pronounced. For example, if $n = 4$, then the rules would reduce from $5^4 = 625$ to $5^2 = 25$, a 96% reduction versus an 80% reduction for $n = 3$. Figure 8.29 illustrates this simple idea for $n = 2$, 3, and 4.

Example 8.3.5 In this example, the well-known wine glass balancing problem of Yamakawa (1991) is considered. It is desired here to design an experimentally feasible fuzzy controller which would balance an inverted pendulum with a flat surface and a wine glass on top of it.

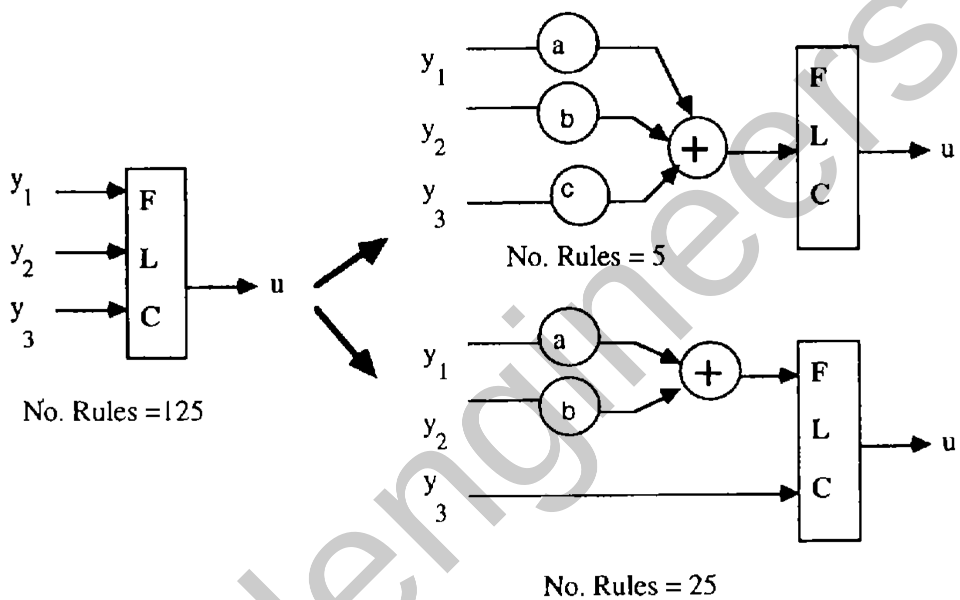
SOLUTION: This system was already shown in Figure 7.14 with both a wine glass and a chick being balanced. A schematic of the system is shown in Figure 8.30. As seen, the system now has six state variables $\theta_1, \dot{\theta}_1, x, \dot{x}, \theta_2$, and $\dot{\theta}_2$. In an attempt to develop a mathematical model for this system, one will be confronted with sixth-order, highly nonlinear and interconnected differential equations involving such nonlinear terms as $\cos(\theta_1 - \theta_2)$, $\sin(\theta_1 - \theta_2)$, $\ddot{x}\cos\theta_1$, $\ddot{\theta}_2 \sin\theta_2$, etc. In short, a very complicated nonlinear model would result. Designing a conventional controller would be possible if the model were first linearized around a set of nominal values and then classical techniques, such as PD or PID, were used to find a controller. This controller is most accurate when the linearized model is a good approximation and plant parameters and variables do not change a lot. In this case, for every new drop of wine poured into the glass, the center of gravity of the system has moved, the nonlinear model has changed, and the linearized model is not as accurate as it was before. It is clear that for this conventional controller to succeed, one needs to redesign or at least keep tuning the linear controller throughout the experiment. Below, a fuzzy controller using the sensory fusion idea is used to implement a robust stabilizing controller for the wine glass balancing problem.

controlengineers.ir

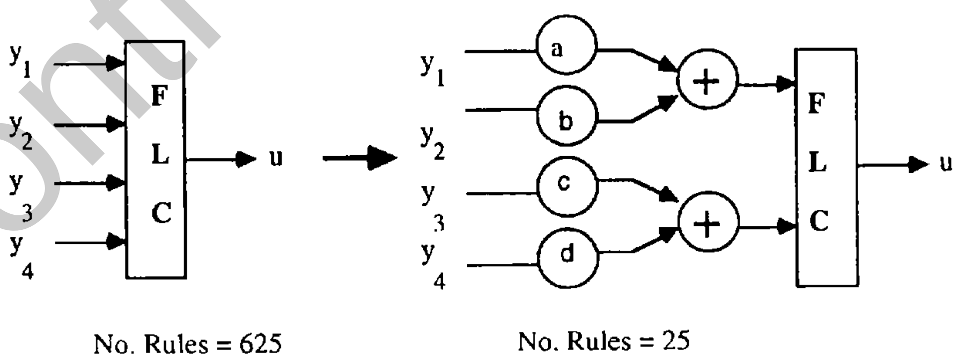
اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) Two variables.



(b) Three variables.



(c) Four variables.

Figure 8.29 Fuzzy logic controller's rule-base reduction for three cases.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

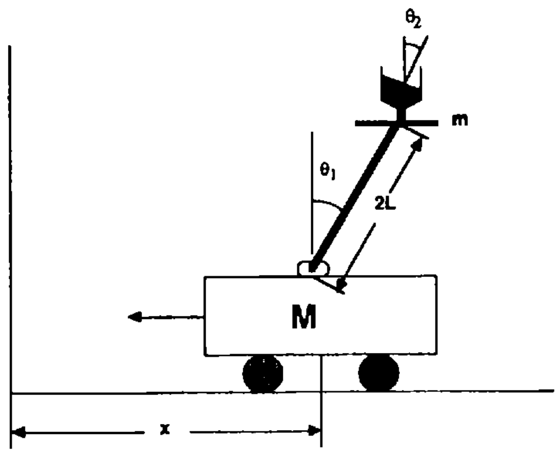


Figure 8.30 Schematic of a wine glass balancing experiment on an inverted pendulum.

The system is assumed to have only four variables instead of six variables, i.e., a reduced model is assumed here. The remaining four variables $\left(\theta_1, \dot{\theta}_1, x, \textit{and } \dot{x}\right)$ constitute a set of 625 rules which may be even worse than dealing with a complicated nonlinear model. The fusion of some of the four variables can now be investigated. One possibility is to combine variable pairs (θ, x) and $\left(\dot{\theta}, \dot{x}\right)$. These combinations, $Y_1 = a_1\theta_1 + b_1 x$ and $Y_2 = a_2\dot{\theta}_1 + b_2\dot{x}$ would provide some measures of the tip of the pole's position and velocity. The sensory data is not appropriate because the dynamics of the cart is not fully incorporated and the system would become uncontrollable. The next choice is to combine pairs $(\theta, \dot{x})$ and $\left(\dot{\theta}, x\right)$. This combination is not realizable and no further steps can be taken. The only other alternative is to combine the variables as

$$
\begin{aligned}
Y_1 &= a_1\theta_1 + b_1\dot{\theta}_1 \\
Y_2 &= a_2 x + b_2\dot{x}
\end{aligned}
\tag{8.3.13}
$$

The new variables Y_1 and Y_2 can be thought of as measures of "emergency" in the variations of pole angle and cart position, respectively. Nominally at this point, a further insight into the physical system would indicate that this combination is realizable. The maximum number of rules at this point is 25. Figure 8.31 shows the fuzzy associate map or "*qualitative phase plane*" of the system. The diagram indicates that

IF Y_1 is NS and Y_2 is NS THEN u is NM

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

ANGLE

POSITION: Y2 \ Y1	NM	NS	AZ	PS	PM
PM	NS		PS		
PS		NS		PM	
AZ	NM		AZ		PM
NS		NM		PS	
NM			NS		PS

Figure 8.31 Qualitative phase plane (FAM bank) for wine glass balancing system.

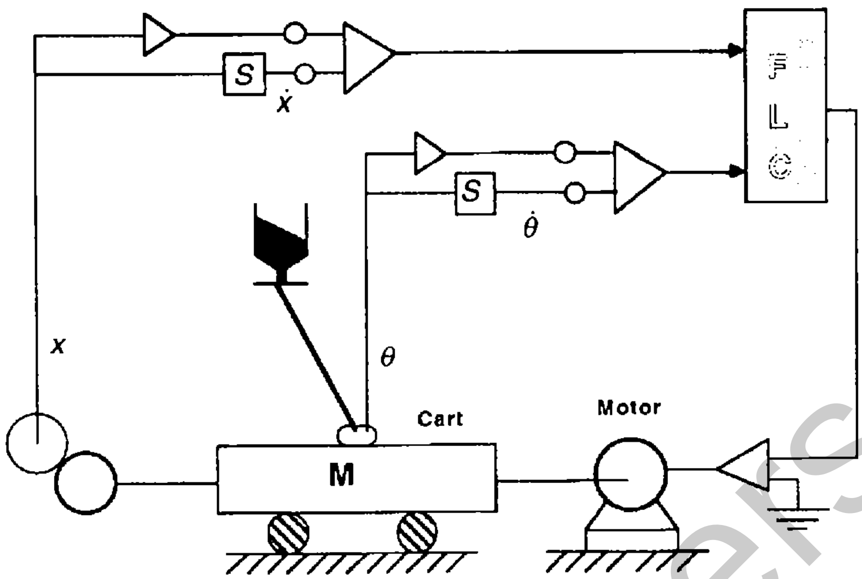
where $u = \dot{x}$, the cart's position is proportional to the armature voltage, exciting the motor which would move the cart. At this point, the pole is leaning on the left and a NM velocity will attempt to correct the perturbation and bring it back. Next, the "trajectory" on the qualitative phase plane moves to the middle bottom square (see Figure 8.31). The angle is AZ now and, hence, a less negative velocity is needed to correct the situation, hence u is NS. Eventually, at the "origin" of this phase plane, one has a rule:

IF Y_1 is AZ and Y_2 is AZ THEN u is AZ

The experimental setup of the system is shown in Figure 8.32. Shown here are the physical implementations of the fused input variables to the fuzzy logic controller. The controller was implemented using four analog fuzzy chips developed by Yamakawa (1991). The number of rules have been further reduced from 25 to just 11 since the remaining scenarios can be taken care of by those rules shown in Figure 8.31 and hence a pruned set of rules have resulted. Note that going from 625 rules to just 11 rules is a huge, i.e., over 98% reduction in the size of the rule base. The analytical phase plane of the variations of cart position based on Yamakawa's experimental work is shown in Figure 8.33. The diagram indicates that the negative disturbance in the cart's position is compensated by a positive control action (velocity) applied to the cart. Yamakawa (1991) has developed this experiment as a testbed for the demonstration of robustness and yet simplicity of a fuzzy controller. A videotape of the experiment is available either through the author or Yamakawa (1991).

controlengineers.ir

controlengineers.ir
@controlengineers
controlengineers.ir



Experimental Set-Up of Wine-Glass Balancing System

(Courtesy: T. Yamakawa, Iizuka, Japan)

Figure 8.32 Experimental setup of wine glass balancing system .

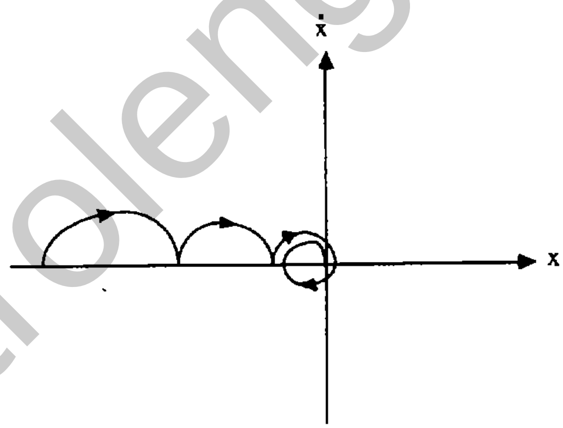
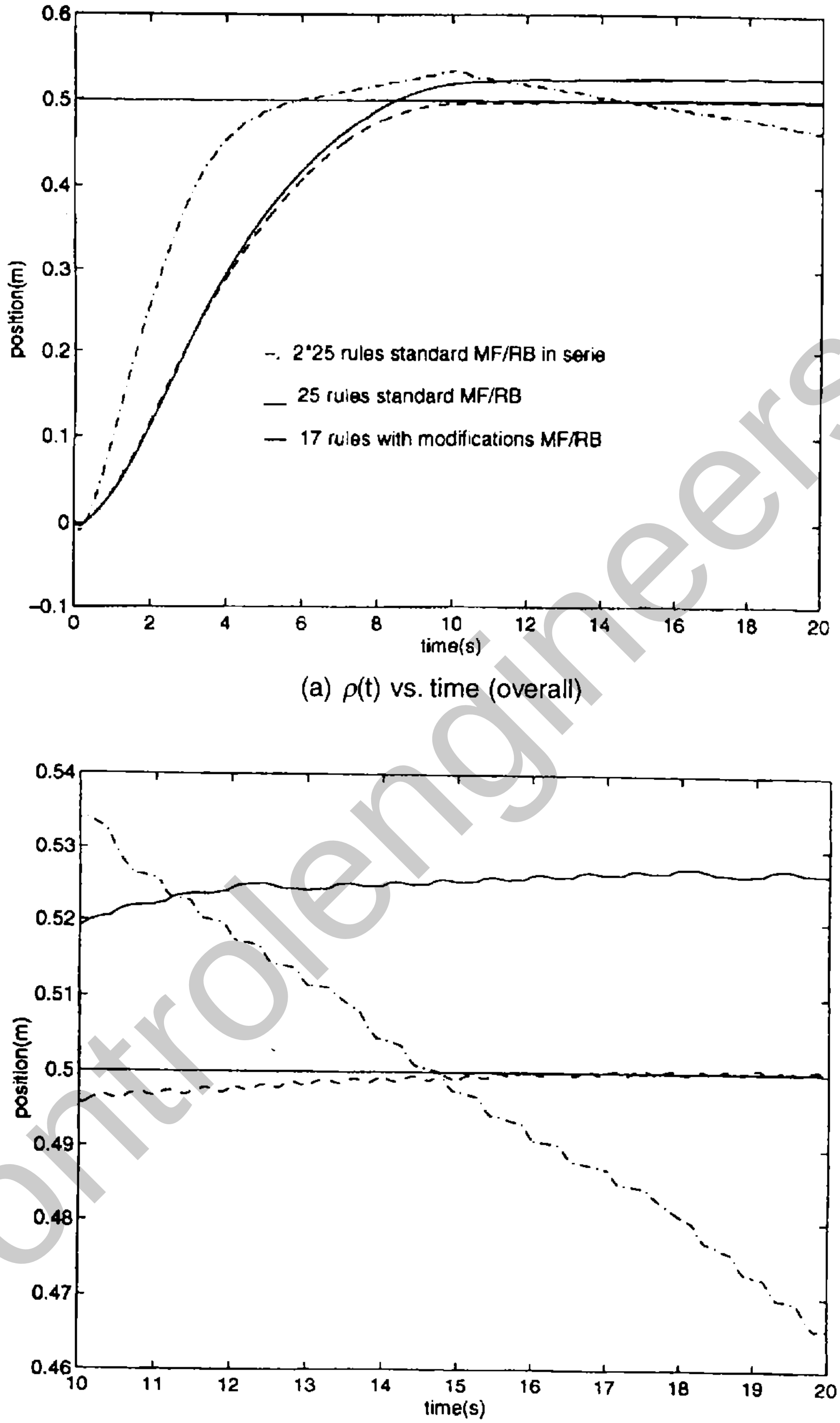


Figure 8.33 Experimental phase-plane trajectory for wine glass stability system.

This example has demonstrated that linguistic and physical knowledge about a system can help solve a much more challenging problem. The controller is easy to debug, easy to understand, and cheap to implement since the sensors' costs are very low. Fuzzy control has demonstrated that it is very suitable for nonlinear complex systems, and through sensory fusion, a great deal of software and hardware development cost, including time, is saved.

controlengineers.ir
@controlengineers



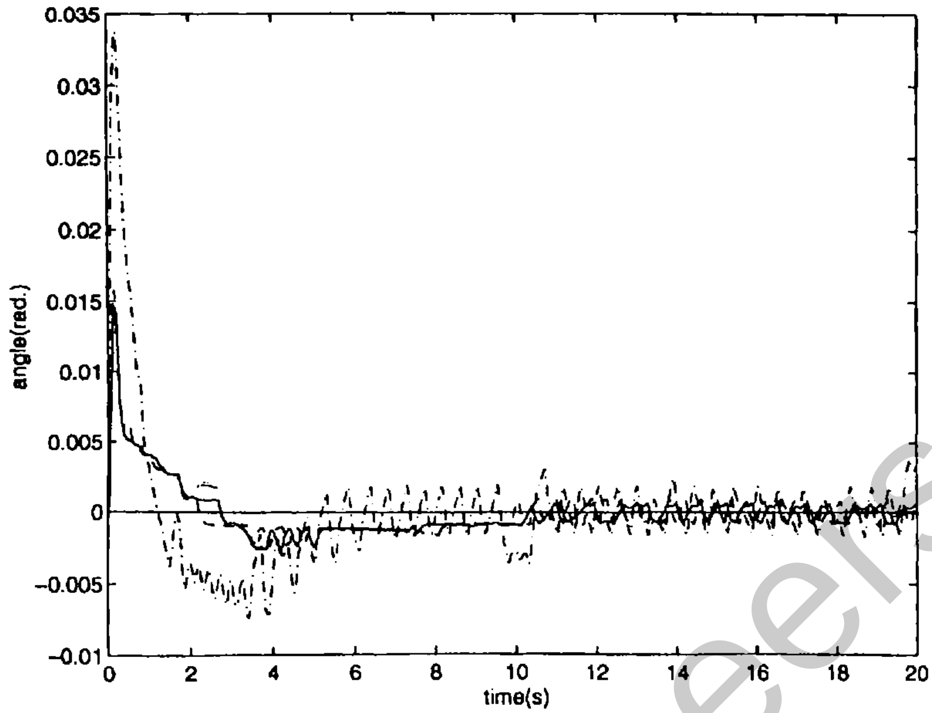
(a) ρ(t) vs. time (overall)

(b) x(t) vs. time (steady-state)

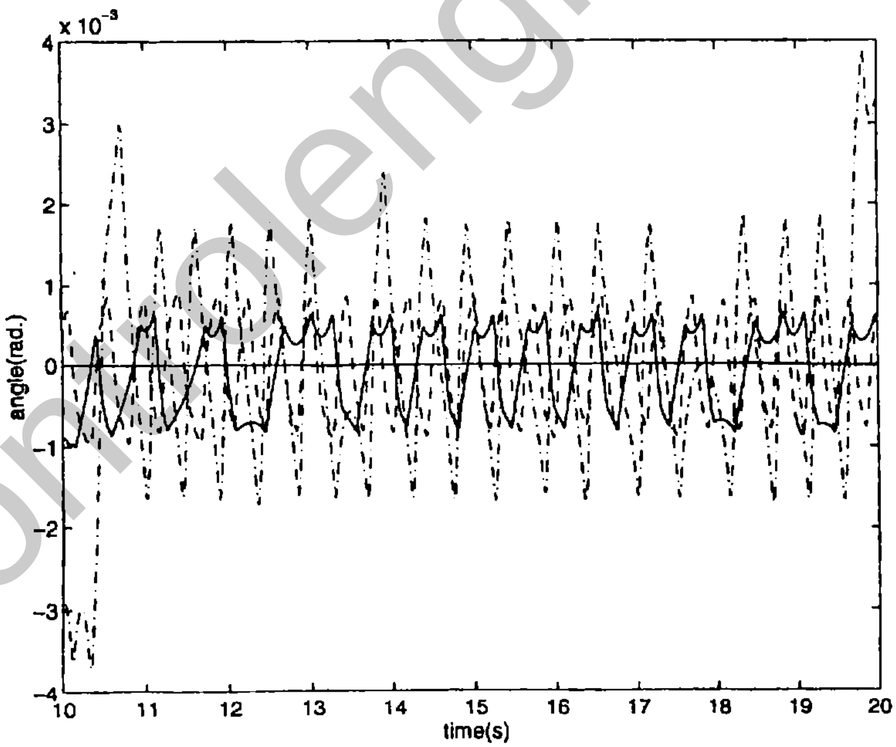
Figure 8.34 Cart position responses vs. time.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) $\theta(t)$ vs. time (overall).

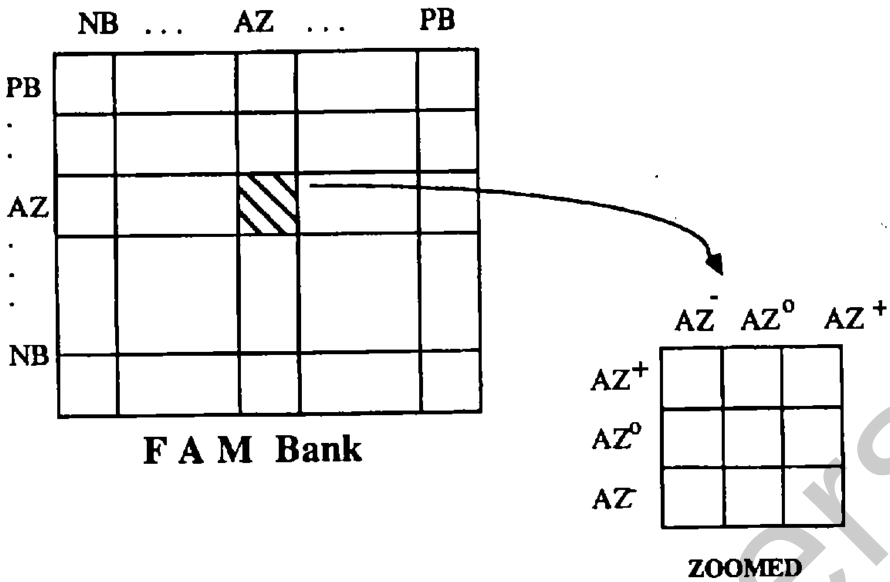


(b) $\theta(t)$ vs. time (steady-state).

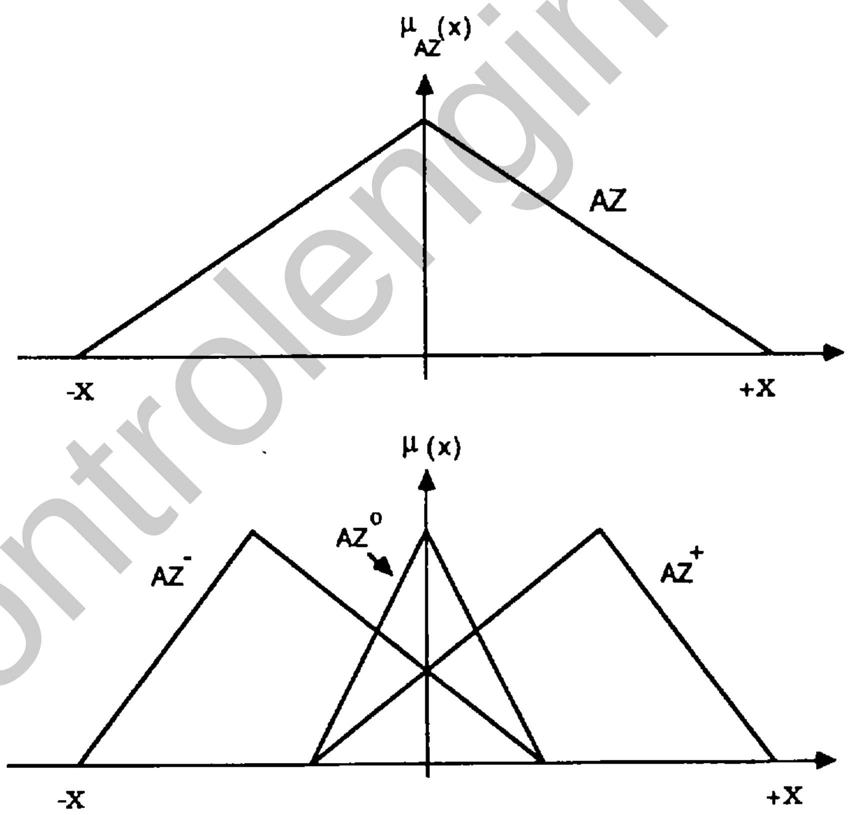
Figure 8.35 Pole angle responses vs. time.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a) zooming of fuzzy rules.



(b) expansion of membership functions.

Figure 8.36 Zooming effect in fuzzy associated memory banks.

controlengineers.ir

controlengineers.ir
@controlengineers
اختصاصی مهندسی کنترل

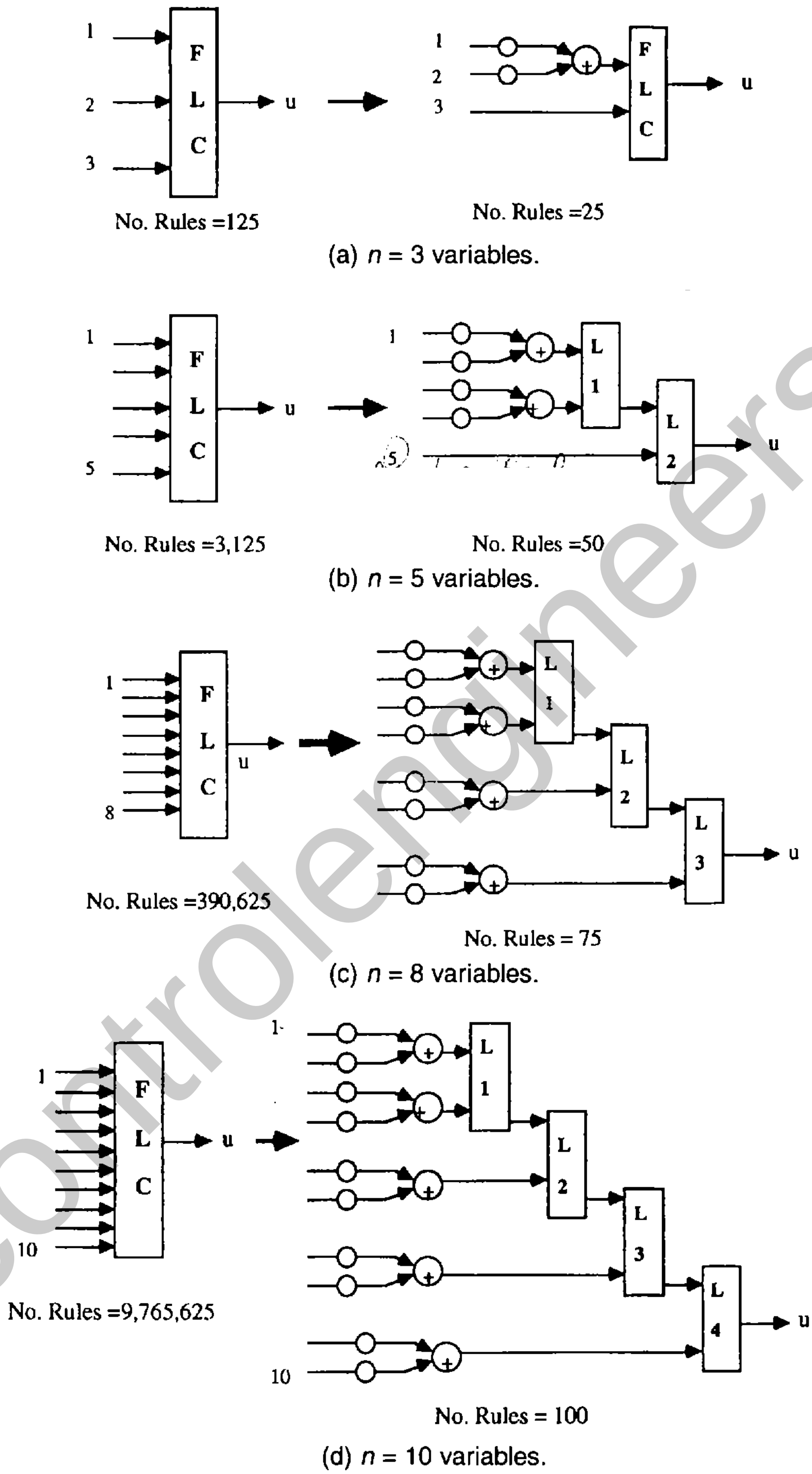


Figure 8.37 Rule-base reduction using hierarchy and sensor fusion.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

In spite of the remarkable reduction of the size of the rule base for lower values of n, if n is large, say $n = 10$, there are $5^{10} = 9{,}765{,}625$ rules in the original base which would be reduced to $5^5 = 3{,}125$ rules if every two sensory variables can be fused. Although 3,125 is the upper bound by way of sensory fusion, the hierarchical structure of Section 8.3.1 would result in only $9(5)^2 = 225$ rules, i.e., a linear increase as a function of n.

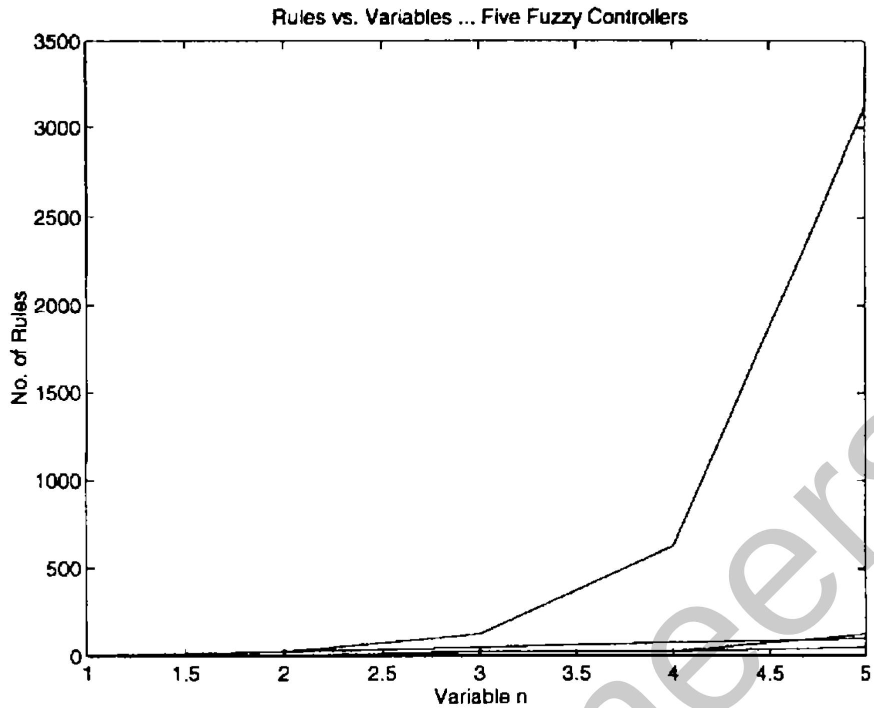
Now, as $n > 10$, the number of rules gets larger and larger regardless of what approach one might choose.

The nonlinear model of the pendulum was simulated using 17 rules (Bruinzeel *et al.*, 1995). Figures 8.34–8.35 show the result of the simulation. The trajectories in these figures represent a 50-rule standard (no fusion), a 25-rule standard membership function/rule base with fusion, and a 17-rule fusion including those of the zooming effect (Pedrycz, 1993) at zero, as shown in Figure 8.36. The responses of the lowest number of rules shown in Figures 8.34–8.35 exhibit the best steady-state performance and fairly comparable transient behavior.

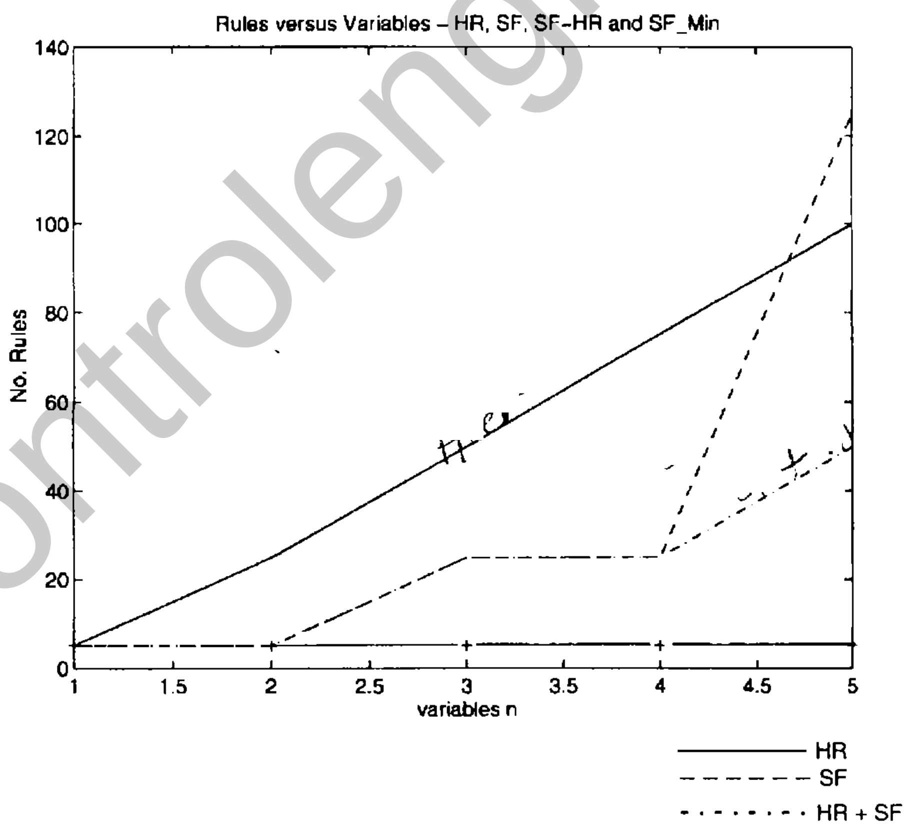
Next, a combination of sensory fusion and hierarchy of rules will be introduced which will further improve the picture.

Hierarchy and Sensory Fusion Approach Here, the variables are simply combined first, as in Figure 8.29, and then are organized into a hierarchical structure similar to that of Figure 8.21. For one or two input variables there is no problem in reducing the size of the rule base. Figure 8.37 shows four possible fuzzy control structures corresponding to 3, 4, 8, and 10 input variables (sensory values). The reductions in the size of the rule base is quite remarkable. For the four cases shown, the size of the rule base reduces from 80% for $n = 3$, 98.4% for $n = 5$, 99.98% for $n = 8$ to nearly 100% (99.998976%) for $n = 10$. Figure 8.38 shows five charts for the size of the rule base. The first one is the single-level standard fuzzy controller rule base which is growing exponentially. The second chart depicts the hierarchical structure of Section 8.3.1, i.e., a linear increase in the number of rules before and beyond $n = 2$. The third chart shows the progression of the number of rules as a result of sensory fusion. The fourth graph represents the progression of rules as a result of combining sensory fusion (two variables at a time) and hierarchy of the rules. Finally, the fifth chart shows *very ideal* (impossible) or absolute minimum rules which could be obtained by fusing all variables and using a single level of hierarchy. Clearly, depending on how many variables can be fused and in what order, when they are put into a hierarchical structure, the size of the rule base would be reduced differently. At this point, questions such as which variables to fuse with which ones, and what are the most suitable values for a_i and b_i coefficients in (8.3.13), are open. The only certain

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



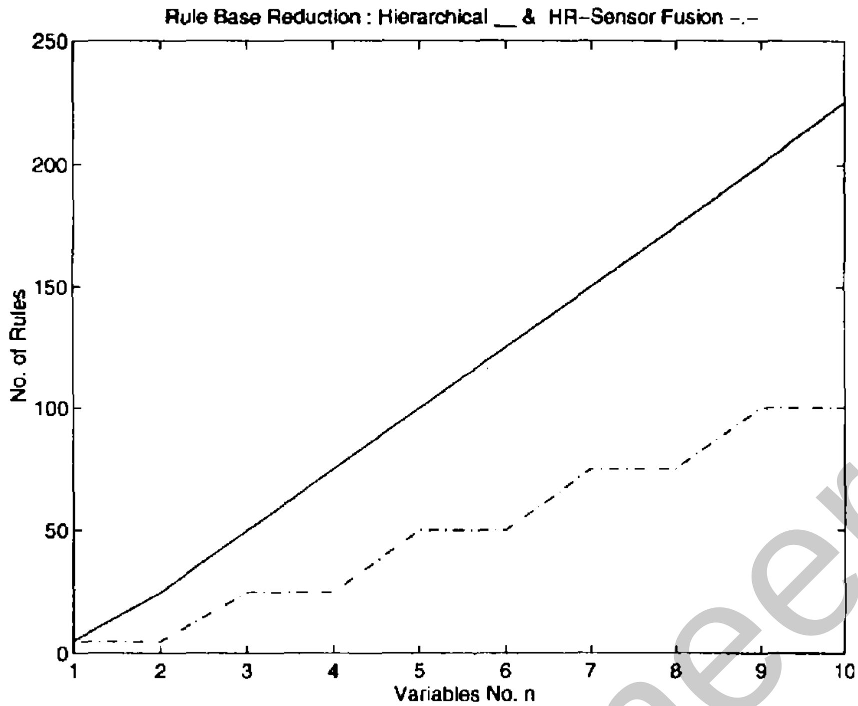
(a) rule-base progression for five methods



(b) rule-base progression for three methods

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(c) rule-base progression for two methods.

Figure ~~**8.39**~~ 8.38 (above and at left) Rule-base size for five possible large-scale fuzzy control structures.

issue here is that the decisions on which variables to fuse and which ones to group at which level of hierarchy depend mostly on one fact or *knowledge* about the system. For the time being, the fusion can be done through the following rule,

$$E = \alpha e + \beta \Delta e \tag{8.3.14}$$

where e and Δe are error and its rate of change, E is the fused variable, and parameters α and β are, at present time, chosen arbitrarily.

Going back to the proposed scheme of fuzzy rule-base reduction, as the number of variables increase, the percent reduction of rules quickly increases to *nearly* 100%, as shown in Figure ~~8.39~~ 8.43.

Example 8.3.6 In this final example, the previous rule-base reduction schemes, including the zooming membership (ZMMF) scheme of Figure 8.36, will be used for real-time fuzzy control of the balancing experiments on the inverted pendulum.

SOLUTION: The bulk of this solution is based on the experiments per-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

formed at LAAS-CNRS in Toulouse, France. The report by Bruinzeel *et al.* (1995) has presented the complete picture.

The real-time experimental setup of the inverted pendulum is shown in Figure 8.39. This system implements both the sensory fusion and hierarchical approach into one. This implementation utilizes the SGS Thompson's ADB board with the WARP chip. The real-time implementation, which would include a PC, is shown in Figure 8.40.

The detailed hardware issues are described in the report by Bruinzeel *et al.* (1995). The angular velocity of the pendulum was calculated via software since no sensor was available to measure it.

In this example, first the real-time results are given and discussed. Next, a comparison is made with the simulation results. Finally, the execution time of the WARP chip is measured and compared with FLCs implemented on a PC.

The experiments that have been done with the different FLCs are the following:

- Observation of the size of the oscillations in both the position and the angle in the steady state;
- Evaluation of a step response, with the same step of -30 cm applied as in the simulations, in order to facilitate comparison;
- Observation of the steady-state behavior with the special setup as described in Figure 8.29;
- Evaluation of a step of -30 cm with the special setup.

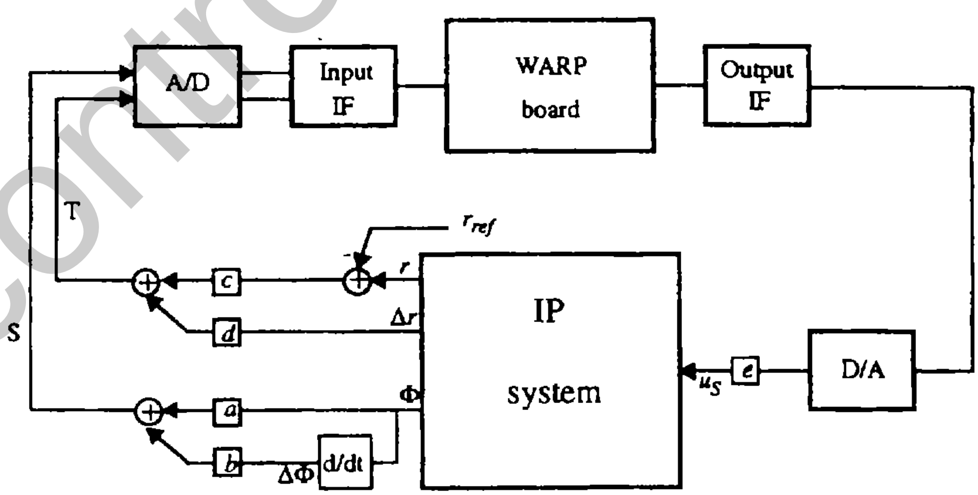


Figure 8.39 Block diagram of sensory fusion hierarchy FL control approach.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

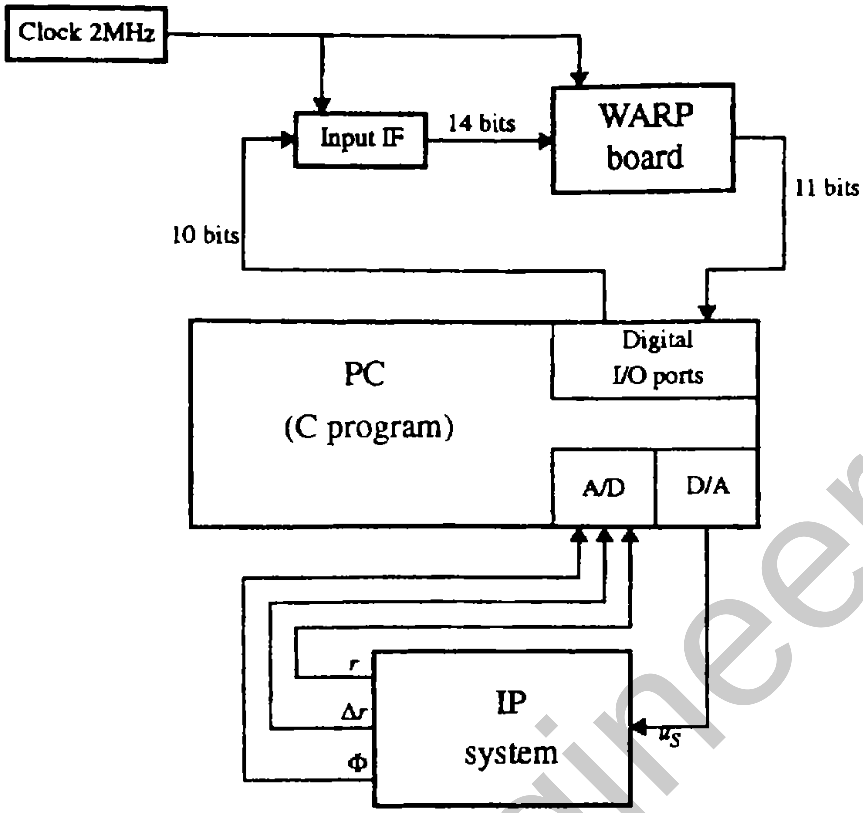


Figure 8.40 Block diagram of the real-time implementation via PC.

The different FLCs whose results will be dicussed are

S25(Sensory Fusion with 25 rules): This is the FLC based on sensory fusion and with application of ZMMF (see Figure 8.36) as was designed and simulated. However, the FLC has been modified slightly. In order to force the cart to stay on the reference position, the membership fuctions NS and PS of the position emergency had to be more extremely modified, according the ZMMF approach.

S25 (without ZMMF): This FLC is similar to *S25*, only without ZMMF. That means that this FLC has only triagular membership functions.

S9 This FLC is resulted from the rule-base reduction straight from the control surface. The data from the 25-rule control surface was fed into the FULDEK software environment (see Appendix B) and asked to create nine rules. This approach

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

approximates a certain FLC by another FLC that has only triangular membership functions. One can conclude that with nine rules the small changes that had been made to the membership functions if *S25* cannot be approximated properly. Therefore, ZMMF had to be applied to *S9* as well in order to obtain the desired precision in the position of the cart. The rule base of *S9* is given below:

P \ T	N	Z	P
P	S7	S8	S9
Z	S4	S5	S6
N	S1	S2	S3

The SGT's WARP board used here can work "stand alone," without the assistance of the PC, after rules have been downloaded onto the WARP chip. Therefore, to control the Inverted Pendulum (IP), the WARP board can be connected "directly" to the system to be controlled. However, as shown in Figure 8.40, one can use a PC and the WARP board together as well.

The PC used for the real-time experiments had a 486-type processor and a clock frequency of 66 MHz. On the PC were installed DOS 6.00 and Windows 3.1 (necessary to run Fuzzy Studio and FULDEK). The PC was equipped with two I/O boards, the RTI-817 (Real Time Devices, Inc., (State College, Pennsylvania, USA), and the PCL-812 (Advantech Co., Ltd., Taiwan).

The RTI-817 allows 24 bits digital input or output, configurable per 8 bits. This board is used for digital output (10 bits) to the Input IF of the WARP board.

The PLC-812 Enhanced Multi-Lab Card offers A/D (16 channels) and D/A converters (2 channels with a 12-bit resolution), 16 bits digital input, 16 bits digital output, and a programmable timer/counter. One channel generates a clock signal of 2 MHz that is used as a clock for both the WARP board and the Input IF. Of the digital input, 11 bits are used to read the output of the WARP board and "NP."

Amira (Duisburg, Germany), the manufacturer of the inverted pendu-

controlengineers.ir
@controlengineers

lum (type LIP 100), offers C language functions to facilitate the use of the inverted pendulum.

The experiments were performed for the "wine-glass balancing problem" as described in Section 8.3 and Figures 8.30 and 8.32.

In the regular inverted pendulum problem, a solid weight is placed at the tip of the pole and therefore the center of gravity is fixed. In this case, the weight at the tip of the pole is partly liquid, so the center of gravity is continuously changing. A mathematical model for such a system is a set of complicated 6th-order nonlinear coupled differential equations.

The result of sensory fusion $\Theta = a\theta + b\Delta\theta$ and $X = cx + d\Delta x$ reduces the number of rules from 625 (5^4) to only 25 (5^2). The value of parameters a to d were chosen through trial and error. The data from the control surface of 25-rule design (one ASCII file of $11 \times 11 = 121$ look-up table) was used on FULDEK to generate only nine rules. The number nine was reached through experimental experience. The control surfaces for both the 25- and 9-rule cases are shown in in Figure 8.41. As one can see, the two cases are fairly close.

In real time, one deals with noise on the sensor signals, with friction on the cart, and with nonlinearities of the DC motor. One cannot expect the real-time results to be as good as the results of the performed simulations. The sizes of the the position and angle oscillations are about five to ten times larger in the real-time experiments. Especially in the steady state, where small and precise control signals are given, the rather rough constant friction compensation results in larger oscillations. Furthermore, both *S25* and *H34* have slightly shorter rise times (for the same values of parameters a, c, and e). This might be due to the constant friction compensation or again to the ZMMF.

Typical real-time results for both angular and linear positions for three different control configurations are shown in Figure 8.42 and Table 8.3.

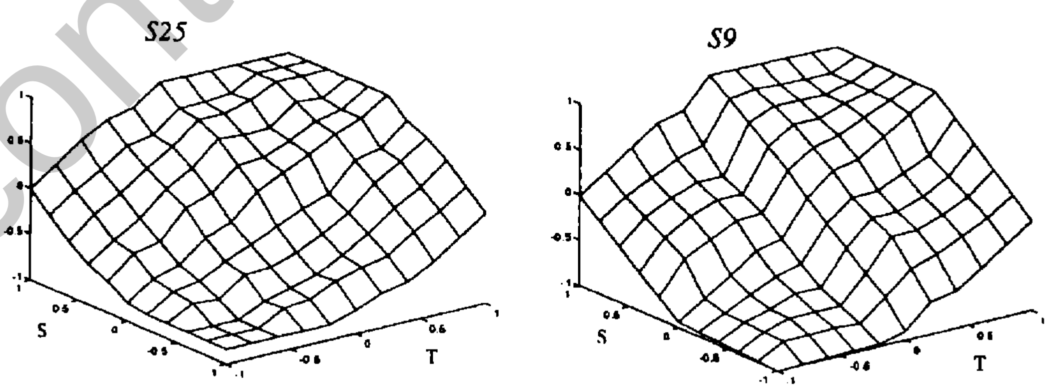
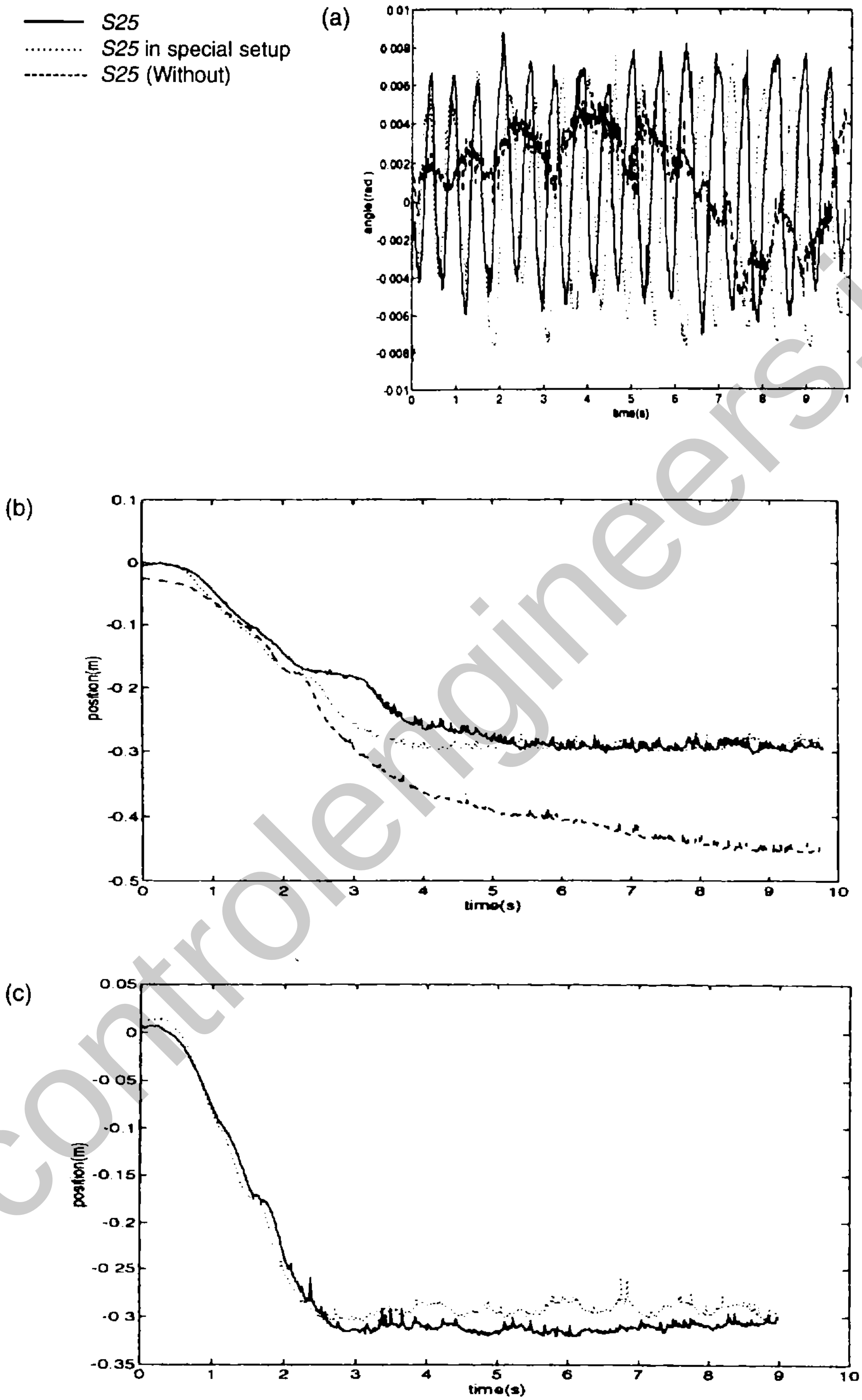


Figure 8.41 Control surfaces of *S25* and *S9*.

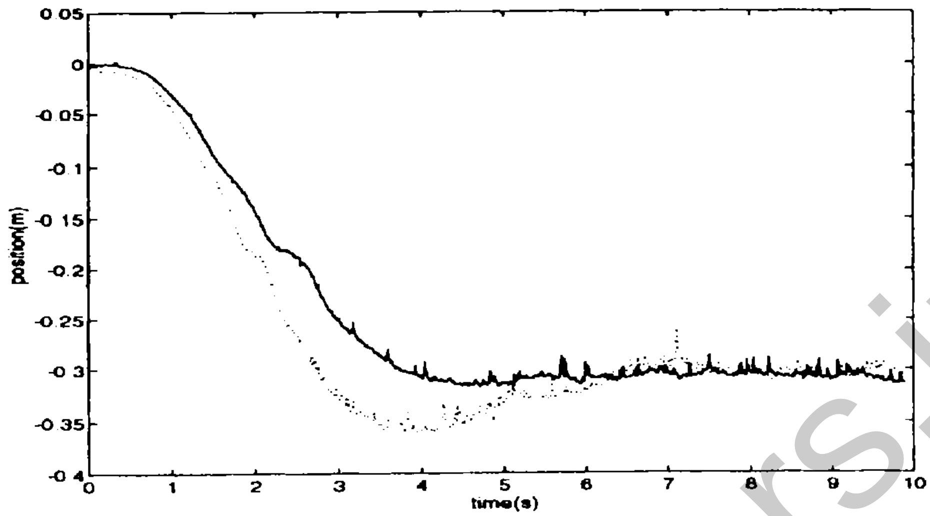
اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



controlengineers.ir

controlengineers.ir
@controlengineers

(d)



(e)

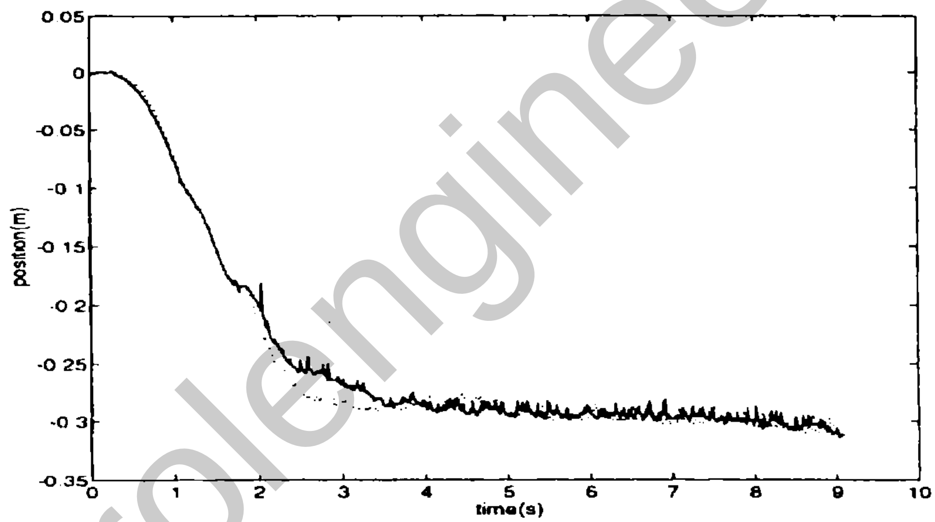


Figure 8.42 Typical real-time results for wine-balancing experiment.

(a) *S25* angular positions (—— *S25*, ······· *S25* with rules set, ------ *S25* without zooming).

(b) *S25* linear cart positions as in part a.

(c) *S9* linear cart positions (—— *S25*, ······· S25 with membership functions modification - MFs)

(d) *H34* cart positions (—— *H34*, ······· *H34* with special MFs)

(e) *H18* cart positions (—— *H18*, ······· *H18* with special MFs)

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table 8.3 Results of Real-time Experiments

FLC	*S25(Without)*	*S25*	*S9*	*H34*	*H18*
Number of Rules	25	25	9	34	18
Steady-state oscillations in position (cm)	32.0	1.3	1.3	2.2	1.1
Steady-state oscillations in angle ($x10^{-3}$ rad.)	10	15	16	12	10
Rise time (5%) (step of 30 cm) (s)	-	5.0	2.5	3.6	3.4
Oscillations in position in special setup (cm)	-	1.6	1.8	1.5	1.4
Oscillations in position in special setup (cm)	-	15	18	11	14
Rise time (5%) in special setup (s)	-	3.5	2.2	5.2	2.6

The control configurations were chosen as:

(i) Sensory-fused controller without zooming *S25* (without),
(ii) Sensory-fused with zooming *S25*,
(iii) Sensory-fused with reduced rule set *S9*,
(iv) Sensory-fused and hierarchical rule set (*S25* rules plus 9 zoomed rules = 34), *H34*,
(v) Sensory-fused with reduced rules and hierarchical set (*S9* rules plus 9 zoomed rules = 18), *H18*.

The results are somewhat expected. The system consists of sensor noise, mechanical function, and various disturbances which would heavily test the robustness of the controller. The real-time results are expected to be rougher than simulation (see Figures 8.33–8.34) as expected, since the latter depends on the mathematical model. The sizes of transient oscillations are five to ten times larger in the latter case. Both *S25* and *H34* real-time experiments were also compared with quadratic D-stabilizability robust pole assignments by Garcia and Bernussou (1995), which involves an iterative solution of a parmeter-dependent, discrete-time Riccati equation. The results of this comparison are available through a videotape comparison which can be obtained by mailing the post card at the end of the book. In the author's opinion, both controllers were quite satisfactory, but fuzzy controllers provided more robustness and disturbance-free qualities. Readers can, after a review of the videotape, judge for themselves.

In summary, this example provides at least a real-time proof of the concept of fuzzy control application for complex systems. It is quite arguable that a four-state inverted pendulum is "complex," but it is felt here that this experience can be considered as a motivating *start* for the reader.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The next section will introduce schemes to reduce a rule base through structural perturbation, decomposition, and decentralization.

8.3.3 Hybrid Fuzzy Control Systems

The integration of fuzzy control or fuzzy logic with standard or nonstandard approaches of control, hierarchy, decentralization, model reduction, search, optimization, and clustering will be briefly presented. Some of the proposed approaches in this section are intentionally left as an open research problem to entice some interest in the readers.

Fuzzy-PID Controller The set-point accuracy of standard PID controllers and nonlinear characteristics of fuzzy controllers can be integrated into a hybrid fuzzy-PID architecture.

The notions of hybridization and hierarchy in which either a fuzzy controller is combined with another controller, e.g., conventional or to use fuzzy logic for both control purpose and plant behavior (state identification). Here, what we mean by *state identification* is a classification of various states that a given system can attain throughout its dynamic history. As an example, consider the case of a flexible link robot (Akbarzadeh *et al.*, 1994b) where the link can be "straight," "oscillating," "bent to the right," "bent to the left," etc. The consequences of the rules which determine the *approximate* behavior of the robot would, in turn, influence the behavior of a low-level fuzzy controller either through the inference engine or membership functions or both. Sayyarrodsari and Homaifar (1995) have presented a similar concept in their work in which a fuzzy partitioning of the input space of the controller would allow the higher-level hierarchy to define the *firing boundaries* of the lower-level rules within the inference engine.

The adaptation of a PID (multiterm) controller was already presented in Section 8.2.2 (see Figure 8.43). Here, a different point of view and structure is presented (Pedrycz, 1993).

Consider the hybrid structure shown in Figure 8.44. The switch s/w provides a combination of the control signal u_1 by the fuzzy logic controller and u_2 by the PID controller. The control u is given by

$$u = \alpha u_1 + (1 - \alpha)\, u_2 \tag{8.3.15}$$

where

$$\alpha = \frac{c(e,\Delta e)}{c(e,\Delta e) + f(e,\Delta e)}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

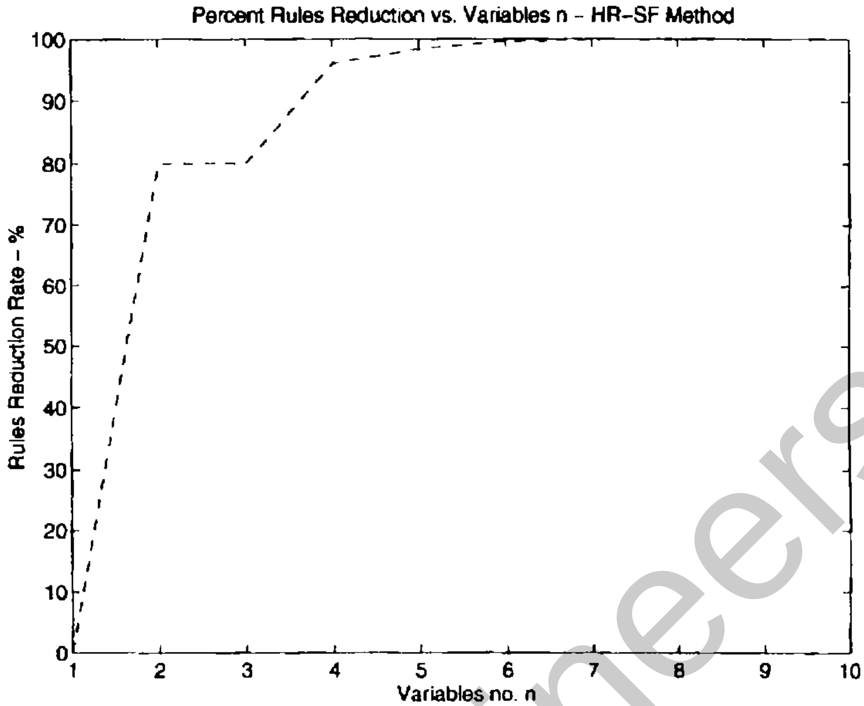


Figure 8.43 Percent of size reduction of the rule base for combined hierarchy and sensory fusion method.

and $c(e,\Delta e)$ and $f(e,\Delta e)$ are two fuzzy sets representing a degree or a measure of being close to and far from the zero error as functions of error and change of error.

Figure 8.45 shows a schematic of these fuzzy sets. In this way, the control (8.3.15) would behave as follows: When error is large positive or negative, i.e., the "far" set membership value is close to one, i.e., $\alpha \approx 0$ and $u \approx u_2$, a fuzzy controller. When the error e nears the zero value (positive or negative), $\alpha \approx 1$ and control $u \approx u_1$, a predominantly PID controller.

This scheme of hybrid control can also be used between a fuzzy logic controller and any other conventional schemes such as *sliding mode*, *optimal*, etc.

Decentralized-Fuzzy Control Another approach to introduce fuzzy logic control in large-scale systems is to apply a set of fuzzy rules for each decentralized local controller as discussed in Chapter 5. Consider a linear discrete-time, large-scale system

$$x(k+1) = Ax(k) + \sum_{i=1}^{N} B_i u_i(k) \tag{8.3.16}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

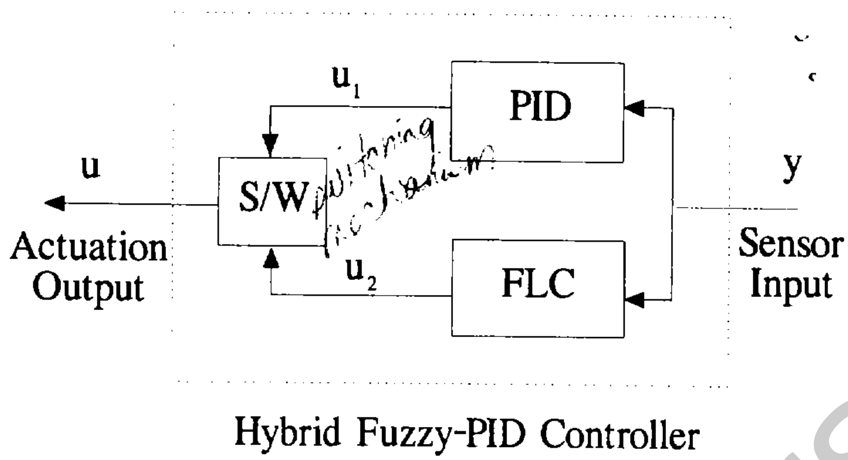


Figure 8.44 A hybrid fuzzy-PID control structure.

with ith output equation

$$y_i(k) = C_i x(k) \tag{8.3.17}$$

for $i = 1, \ldots, N$. Assume now that each local output can be used to activate the ith local fuzzy controller given by a structured set of rules given below:

$$\begin{aligned} &u_i(k) = FLC_i(y_i) \\ &\mathrm{R}^i\text{: IF } y_i^{(k)} \text{ is } A^i \text{ and } \Delta y_i^{(k)} \text{ is } B^i \text{ THEN } u_i^{(k)} \text{ is } C^i \end{aligned} \tag{8.3.18}$$

where k can be changed from k to $k-1$, $k-2, \ldots$

Figure 8.46 shows a decentralized fuzzy control architecture. Here each local control signal will be the defuzzified output of rules of the type given by Equation (8.3.18).

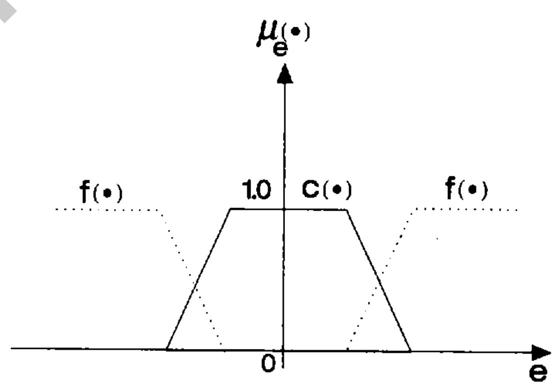


Figure 8.45 Fuzzy sets representing notions of *close* and *far* for error *e*.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

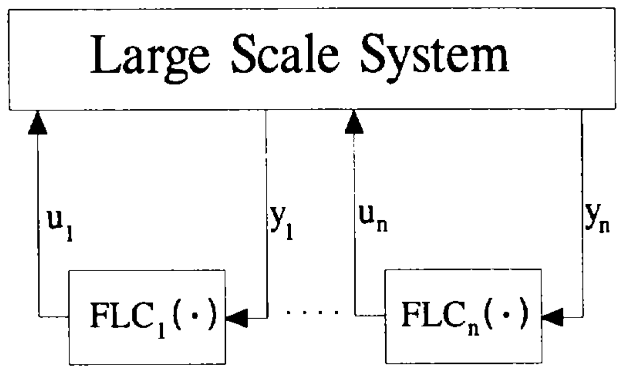


Figure 8.46 A decentralized fuzzy control architecture.

An alternative architecture for a decentralized fuzzy control of a large-scale system is shown in Figure 8.47. Here, the typical local controller would be described by the following relation

$$u(t) = K_p e(t) + K_I \int e(t) d\tau + K_D \dot{e}(t)$$

and the fuzzy tuner's typical rules can be written as

> IF e is A^1 and Δe is B^1 THEN K_p is small
> and K_I is medium and K_D is Big

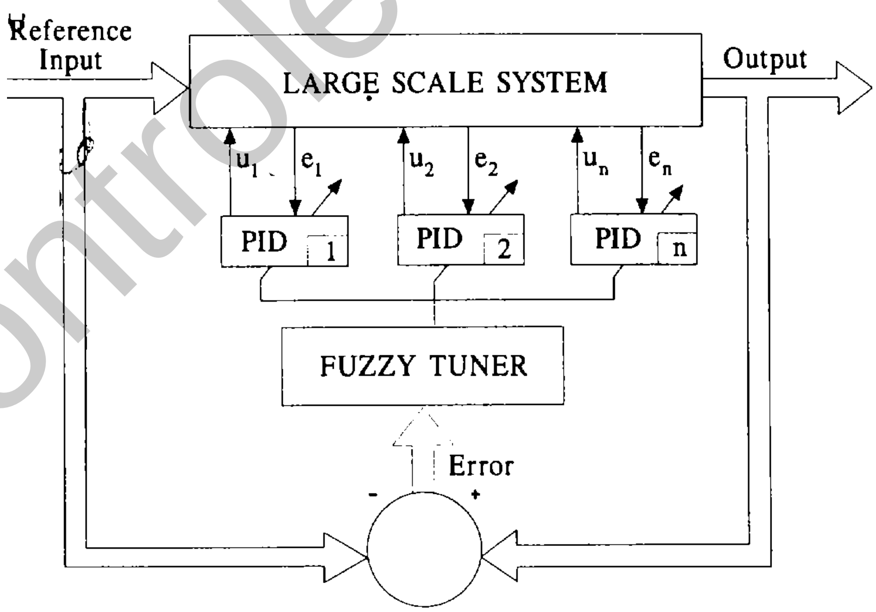


Figure 8.47 An architecture for fuzzy-tuned decentralized PID control of a large-scale system.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Example 8.3.7 In this example a decentralized fuzzy PD control structure, considered by Kumbla and Jamshidi (1995), is briefly presented. Figure 8.48 shows the system.

SOLUTION: The system was both simulated for two links of a Rhino XR-2 educational robot and controlled in real time using a decentralized (independent joint) control of the robot. Figure 8.49 shows the hardware for a single joint of the robot, while the membership functions for inputs and outputs of the fuzzy PD controller are shown in Figure 8.50. Figure 8.51 shows typical simulation results, while the experimental results are shown in Figure 8.52. The results show a rather strong comparison between the fuzzy and conventional PD controllers. The simulation results show that while the conventional controller responses head up for overshoot, the fuzzy PD controller has eliminated the overshoot without sacrificing speed. The experimental results also indicate the strong comparison between the two controllers. However, the velocity response of joint 2 has much less overshoot without slowness in speed of response.

Interaction Predicted Fuzzy Control One of the most attractive approaches for hierarchical control has been the interaction prediction (see Section 4.3 and Chapter 4 of Jamshidi, 1983). One reason for the attractiveness of this method is the simplicity of the coordinator's problem, while reducing

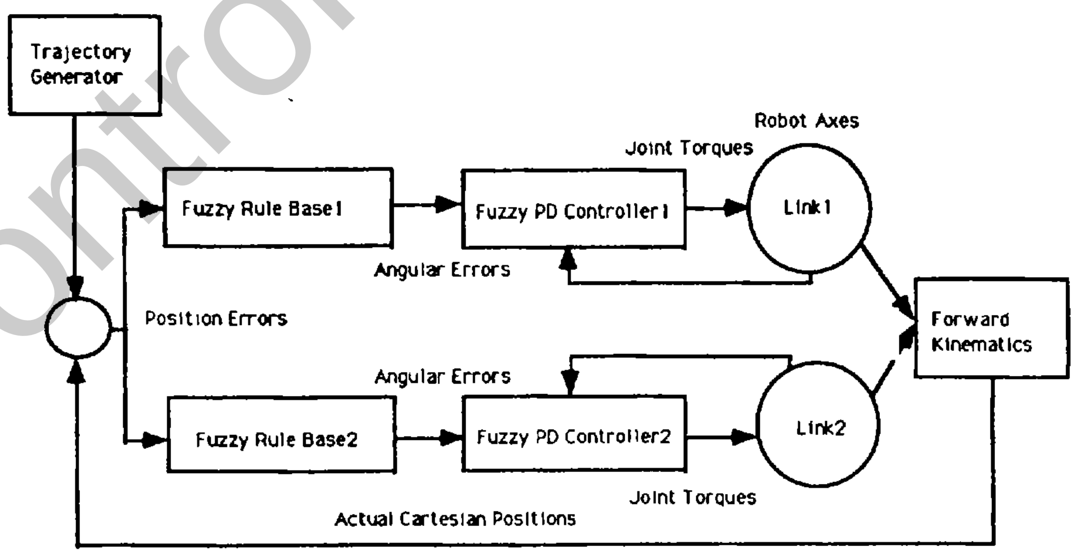


Figure 8.48 A two-link decentralized robot control system.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

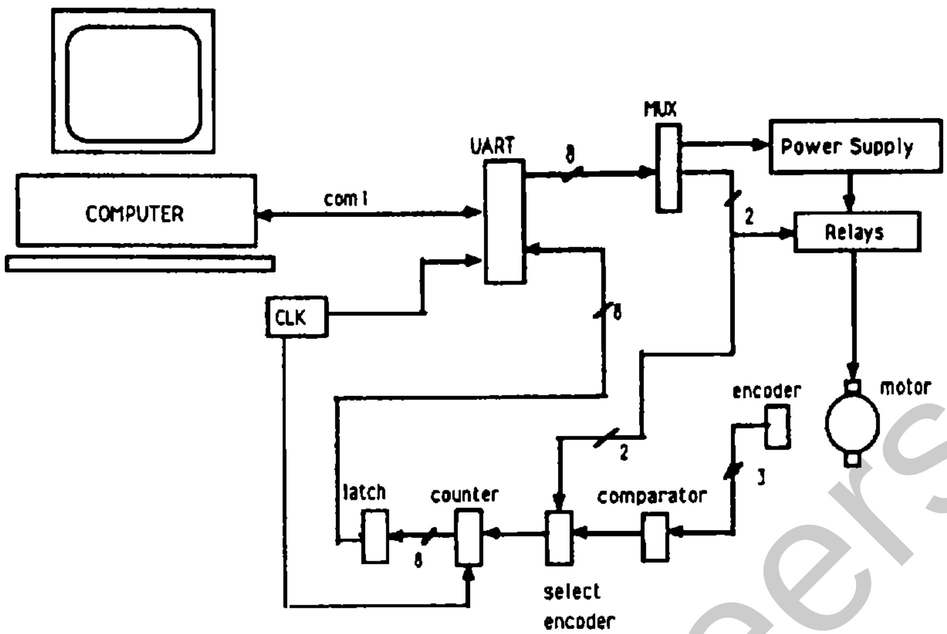


Figure 8.49 Schematic of the hardware for a single link.

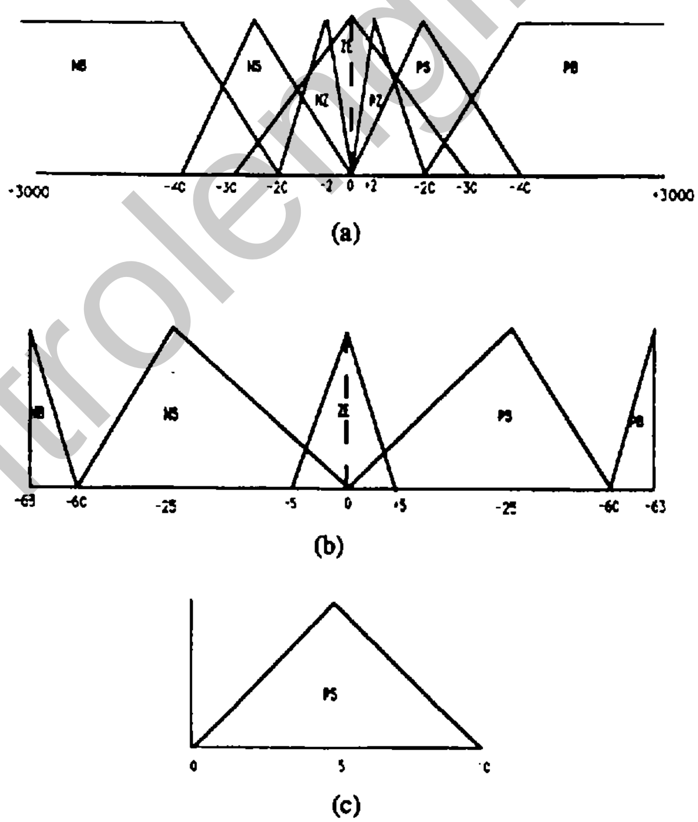
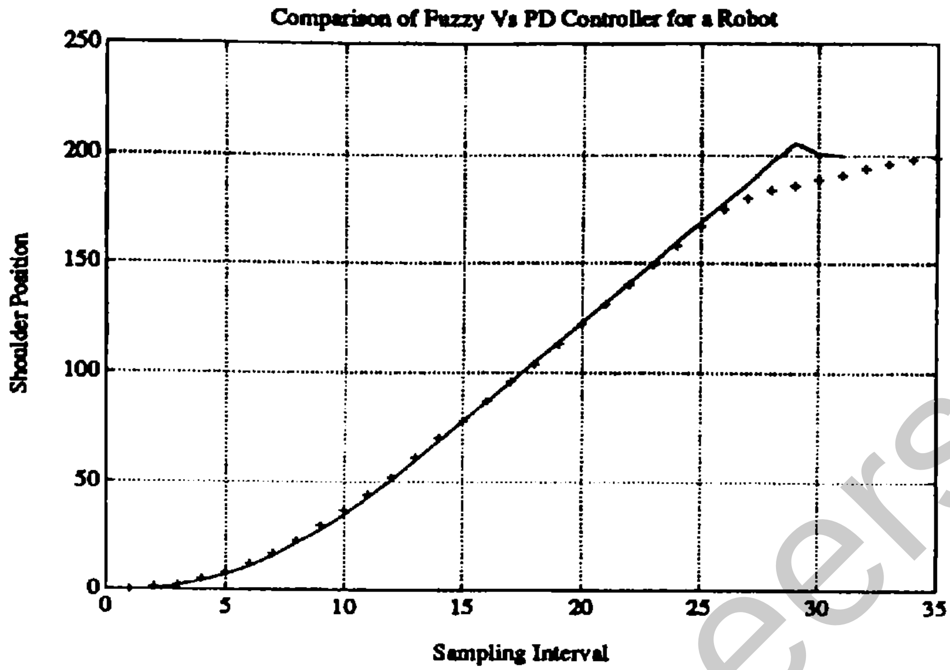


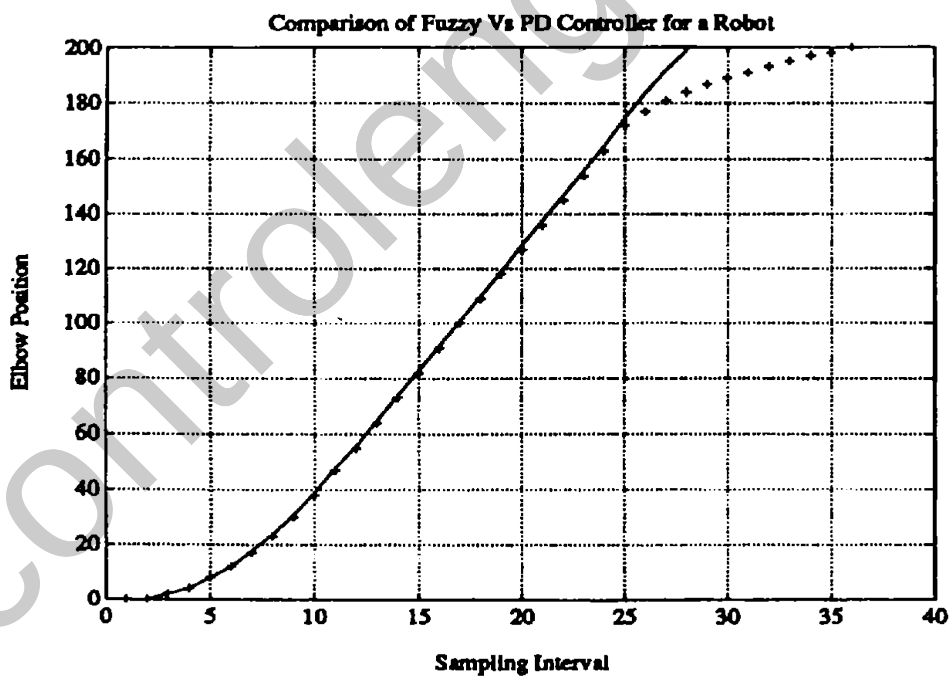
Figure 8.50 Fuzzy membership functions for (a) position error, (b) output voltage and (c) velocity.

controlengineers.ir

controlengineers.ir
@controlengineers



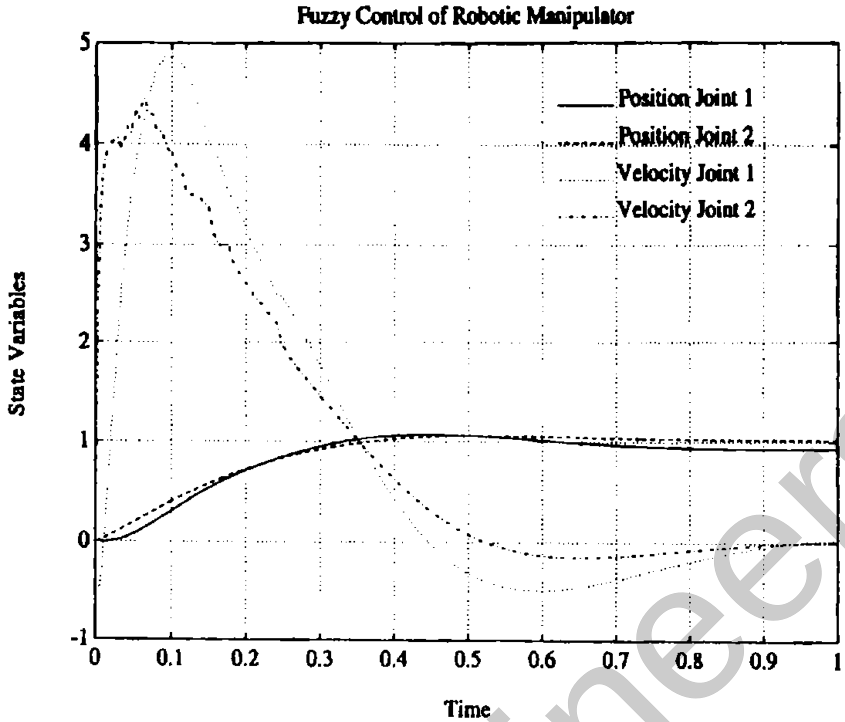
(a) shoulder



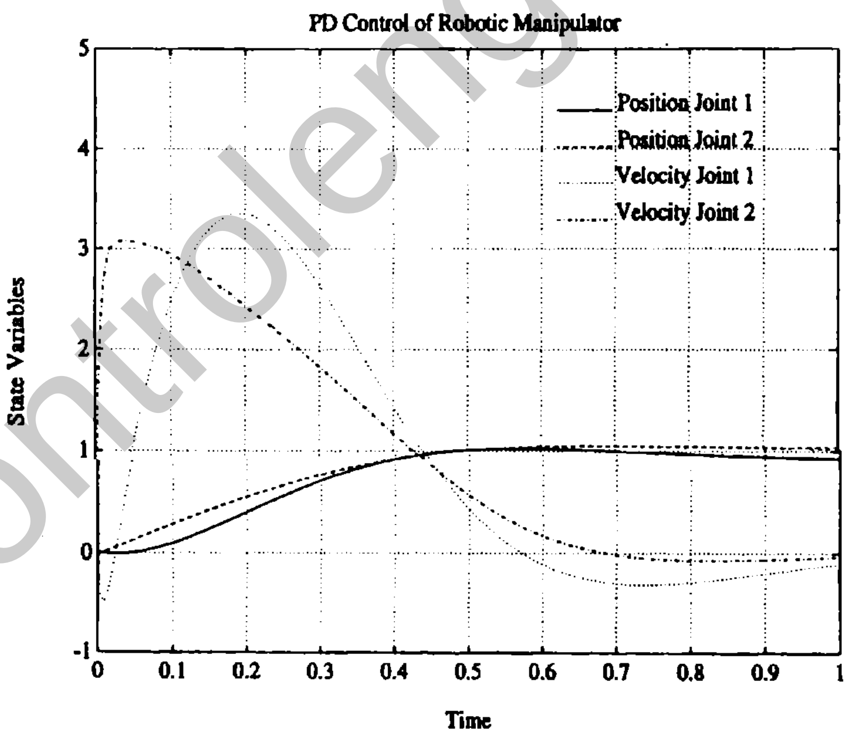
(b) elbow joint

Figure 8.51 Simulation results for decentralized fuzzy PD and conventional PD controllers for a robot.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers



(a)



(b)

Figure 8.52 A fuzzy controller experimental response (a) and a PD controller response (b) of a robotic manipulator.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

the overall system's order to a finite number of subproblems. In spite of these favorable attributes, the fact remains that at the low level, a series of computationally intensive optimal control problems must be solved.

The object of this section is to introduce fuzzy logic into the interaction prediction approach to hierarchical control. The reader is cautioned that the content of this section is merely a proposal for detailed further research and investigation.

Consider a large-scale, discrete-time linear system

$$x(k+1) = Ax(k) + Bu(k) \tag{8.3.19a}$$

$$x(0) = x_o \tag{8.3.19b}$$

where all the terms are defined before (see Chapter 4 of Jamshidi, 1983). Assume that the order of (8.3.19a) is very large, i.e., $n >> 1$, and let it be decomposed into N subsystems each being described by,

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + z_i(k) \tag{8.3.20a}$$

$$x_i(0) = x_{io} \tag{8.3.20b}$$

where the order of (8.3.20) is now n_i with constraint $n = \sum_{i=1}^{N} n_i$.

The interaction term $z_i(k)$ is assumed to follow the relation

$$\mathbf{z}_i(k) = \sum_{\substack{j=1 \\ i \neq j}}^{N} G_{ij} \mathbf{x}_j(k) \tag{8.3.21}$$

and the overall Hamiltonian of the problem is given by

$$\begin{aligned} H_i(k) &= \frac{1}{2} x_i^T(k) Q_i x_i(k) + \frac{1}{2} u_i^T(k) R_i u_i(k) \\ &+ \alpha_i^T z_i(k) - \sum_{\substack{j=1 \\ i \neq j}}^{N} \alpha_j^T G_{ji} x_i(k) + \\ &P_i^T(k+1)\left(A_i x_i(k) + B_i u_i(k) + z_i(k)\right) \end{aligned} \tag{8.3.22}$$

and $\alpha_i(k)$ is the $n_i \times 1$ vector of Lagrange multipliers corresponding to the interconnection constraints in (8.3.21). The remaining terms are self-explanatory. It is also well known (see Equation 4.3.56) that the coordinator's policy (interaction balance) is given by

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$\begin{bmatrix} \alpha_i(k) \\ z_i(k) \end{bmatrix}^{l+1} = \begin{bmatrix} -P_i^{(k+1)} \\ \sum_{j=1}^{N} G_{ji} x_j(k) \end{bmatrix}^{l} \tag{8.3.23}$$

The subsystem problem consists of a discrete-time Riccati solution to (8.3.20) with known $z_i(k)$ and a quadratic cost function (Jamshidi, 1983), i.e.

$$u_i(k) = -F_i(k)x_i(k) \tag{8.3.24}$$

where $F_i(k)$ is related directly to $K_i(k)$ and is the symmetric $n_i \times n_i$ positive-definite Riccati matrix in discrete-time form.

In order to solve the low-level control problem with fuzzy logic, let us introduce the following set of rules for the ith subsystem:

$$\begin{aligned} &\text{IF } \mathbf{x}_i(k) \textit{ is } \mathbf{A}_1^i \;\; \mathbf{x}_i(k-1) \text{ is } \mathbf{A}_2^i \ \ldots \\ &\text{THEN } \mathbf{u}_i(k) = \mathbf{F}_i(k)\mathbf{x}_i(k) \end{aligned} \tag{8.3.25}$$

where A_j^i, j = 1, … are the ith subsystem's vectors of fuzzy labels and $F_i(k)$ is an $m_i \times n_i$ feedback matrix which can be determined arbitrarily for now. Note that the consequence of (8.3.25) could also be linguistic, such as "Then $u_i(k)$ is NM," where NM stands for negative medium. Note also that the rule set (8.3.25) represents n_i rules for the ith subsystem. Having observed this, if there are, say, five linguistic labels per fuzzy set $\mathbf{A}_j^i$, then the size of the rule set would be $5n_i$, where n_i is usually a large number, e.g., 5, 10, or even 15. This indicates that the number of rules per subsystem can become too large. Figure 8.53 shows a block diagram for the proposed interaction prediction fuzzy control system. The coordinator's problem can be simply the upper half of the relation (8.3.23). As an alternative, one can use fuzzy rules in place of the above relation, e.g.,

$$\text{IF } (x_1 \cap x_2 \ldots \cap x_N) \text{is } A_1 \text{ THEN } z_1^{l+1} = z_1^l$$

Hybrid Fuzzy Control—An Application Class In this final section, three hybrid fuzzy control approaches, as applied to robotic systems, will be presented. The presentation here is based on the research being performed by three associates of the author which was reported by Tunstel *et al.* (1995). These three control applications are based on: a) fuzzy-genetic optimiza-

controlengineers.ir
@controlengineers

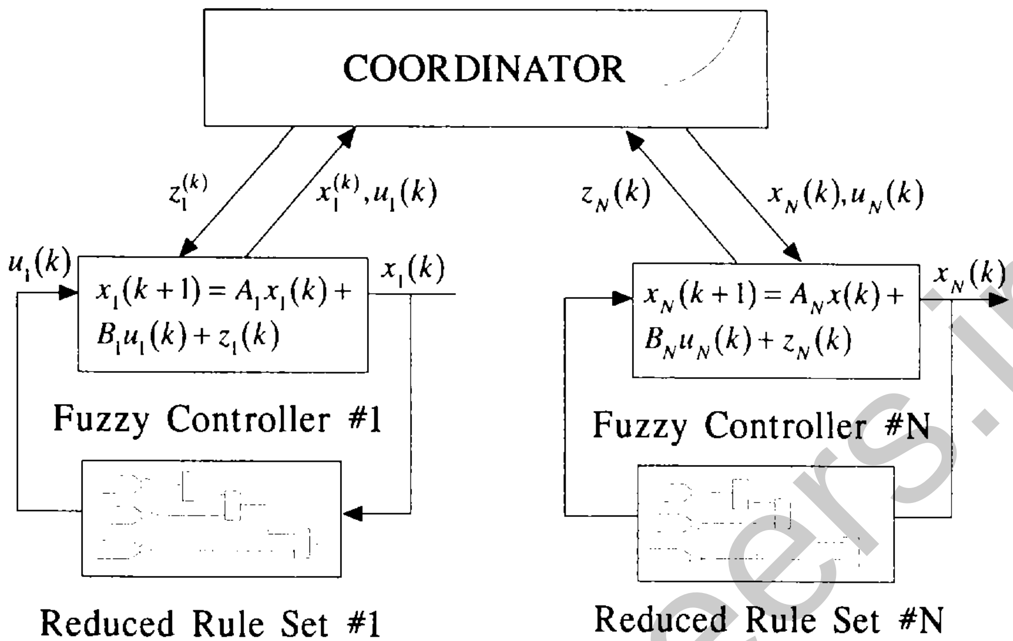


Figure 8.53 A proposed interaction predicted hierarchical fuzzy control architecture.

tion, b) fuzzy-genetic programming, and c) fuzzy-neural network paradigms for robot control.

Fuzzy-GA Paradigm In this first approach, genetic algorithms (GAs) are combined with fuzzy logic to create a hierarchical hybrid *intelligent control* paradigm for distributed-parameter systems. The application area is a flexible robot. These robots are particularly suitable for hybrid approaches since they entail complex mathematical models which depend on both spacial and temporal characteristics. Genetic algorithms are outside of the scope of the present book, but a brief definition is given here.

> *Genetic algorithms* are probabilistic optimization methods which represent adaptation procedures based on mechanics of natural genetics and natural selection. They have two essential components, survival of the fittest and recombination. Each physical problem is coded into 0 - 1 bit strings and the recombination will be done through the processes of crossover and mutation as shown in Figure 8.54. The location of the cross-over point and mutation bits are obtained randomly. The decision on which offspring are to be retained for future recommendation will depend on their relative fitness function value, e.g., a cost or a benefit function. Figure 8.55 shows the basic cycle of the GAs (Homaifar, 1994). Once the GA has converged to the optimal solution, the final bit string

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

will be decoded back to the solution of the original physical problem. Interested readers may consult any textbook on the subject (one is given by Goldberg, 1989).

Figure 8.56 shows the proposed fuzzy-GA paradigm to control a distributed parameter system. By incorporating genetic algorithms in the architecture, the control structure has the ability of learning and optimizing its knowledge base. When developing the hierarchical controller, some initial knowledge is expected to be supplied through expert knowledge for feature extraction and the lower-level control. However, because of the increasing complexity of distributed parameter systems, it is reasonable to assume that the initial knowledge is not optimal. Therefore, a learning ability to improve and optimize the knowledge base is essential.

The feature extraction module is the most important unit in this architecture. Development of the lower level controller depends on a proper choice of features. For a flexible robot, these features could be spatial

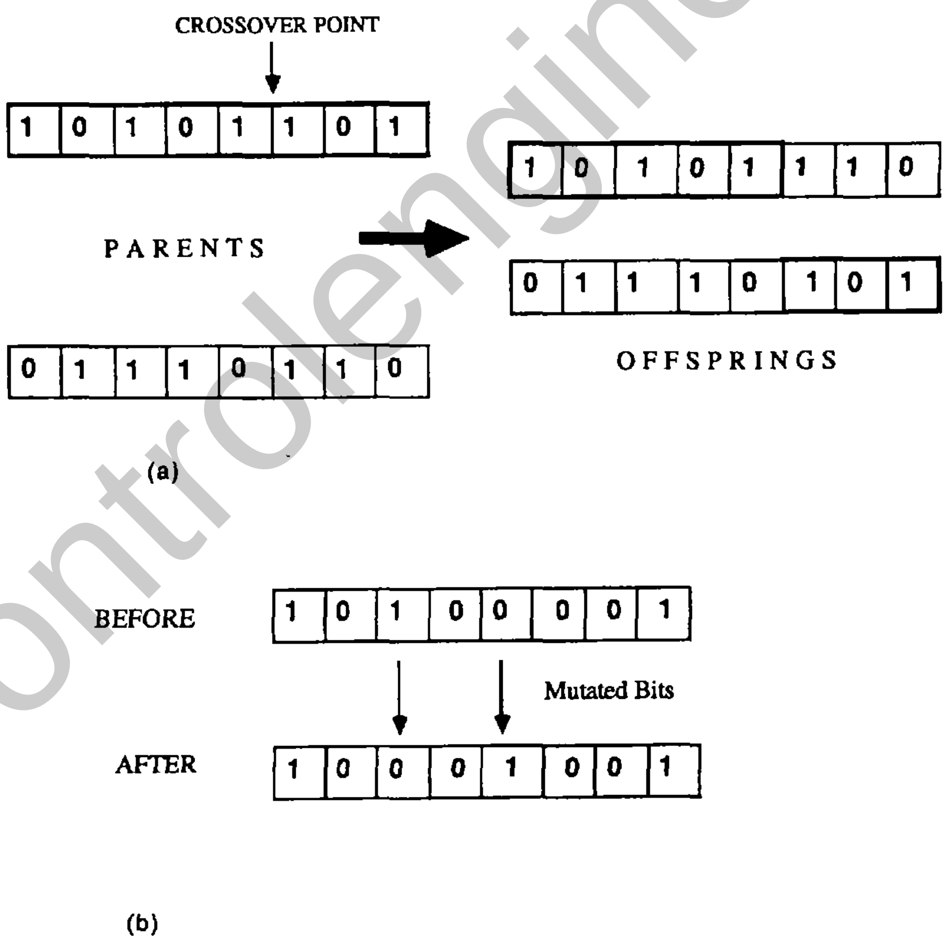


Figure 8.54 The two recombination procedures of GAs.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

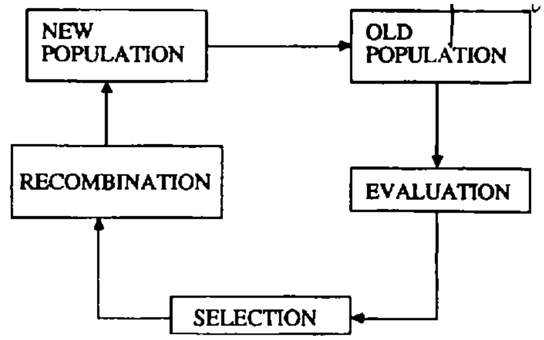


Figure 8.55 The basic cycle of a genetic algorithm.

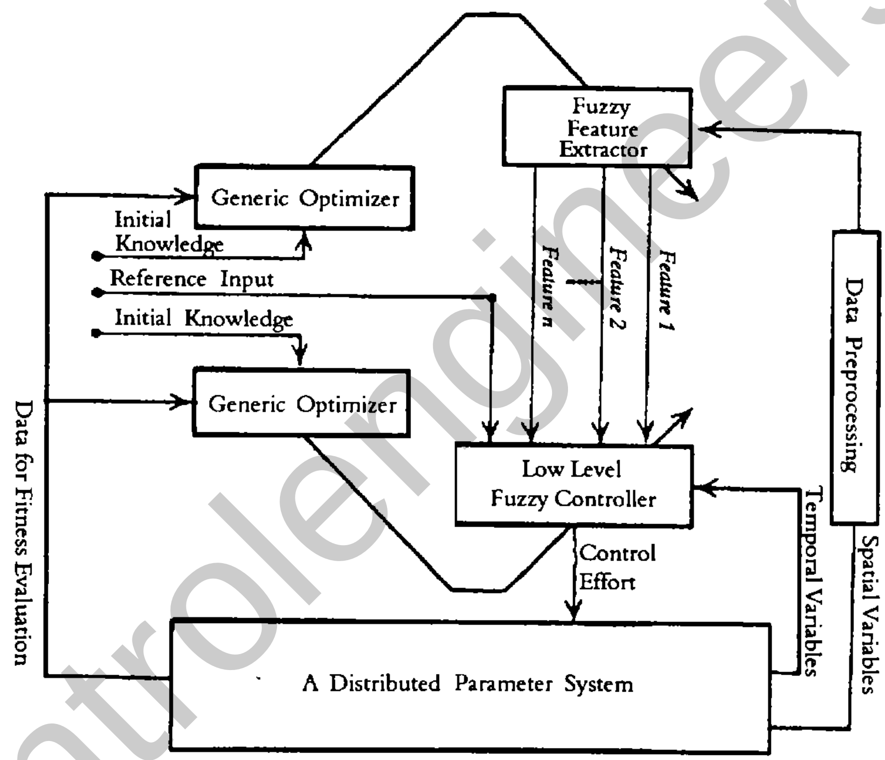


Figure 8.56 GA-based hybrid hierarchical control paradigm.

features such as straight, gently curved, oscillatory, etc. The concepts of average and variance are borrowed from statistics for classification and extraction of various features of the system (arm motions). By nature, these classifications are fuzzy. For example, when one states that a line is *oscillatory*, a legitimate question is, "how oscillatory, a little or a lot?"

As seen in Figure 8.56, the second level of hierarchy uses measured data from the system to determine the features of the system. The table is used and expanded here and constitutes the basis of the expert knowledge

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

from which the fuzzy rules are drawn, using three fuzzy features of the arm and their correspondence to fuzzy sets of $\bar{\delta}$ and σ_{δ}.

The extraction of system's features can be done through a set of fuzzy rules. As an example, consider the rule (Akbarzadeh *et al.*, 1994a) for a flexible beam:

> IF $\bar{\delta}$ is Zero, σ_{δ} is Zero
> THEN *Straight* is High, *Oscillatory* is Low,
> and *Gently Curved* is Zero

Here, $\bar{\delta}$ is the curvature mean and σ_{δ} is the curvature variance in fuzzified terms and the terms *Straight*, *Oscillatory*, and *Gently Curved* are three fuzzy "features" of the system (flexible robot arm).

At the first level of the hierarchy (see Figure 8.56), the objective of the fuzzy controller is to track a desired trajectory while keeping oscillations along the robot arm at a minimum. First, a table of expert's knowledge is presented, and based on this table, certain rules are drawn. The genetic algorithm will upgrade this knowledge base based on the performance fitness of the rule base. Visualizing different arm features is an important part of the controller design. The control depends to a large extent on the system's features. *Velocity* and *Error* θ have nine possible combinations (linguistic labels). Therefore, a total of 36, 4×9, rules complete the design of the controller. A sample rule in the proposed fuzzy controller may be as follows,

> IF *Straight* is High, *Oscillatory* is Low,
> *Error q* is Zero and *Velocity* is Positive
> THEN *Torque* is Negative.

This project is ongoing and the simulation results have been satisfactory and the experimental work is well underway to implement the architecture on a DSP (digital signal processing) chip for a testbed robot arm.

Fuzzy-GP Paradigm In this section, fuzzy logic and genetic programming are used to control a mobile robot through a behavioral decomposition of its tasks. GP, as an intelligent system paradigm, is not considered here as it is outside the scope of the text. However, GP, unlike GA, does not include a coding or decoding, and the algorithm is operating on the linguistic structure of the fuzzy rules themselves. In genetic programming, the population consists of computer programs (individuals) that are hierarchi-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

cal compositions of functions and terminals (arguments) of various sizes and shapes (Tunstel and Jamshidi, 1996). These individuals take part in a simulated evolutionary process wherein the population evolves over time in response to selective pressure induced by the relative fitness of the individuals in a particular problem environment. As an example, consider the following two fuzzy rules:

R1: IF e is A_1 and Δe is B_1 THEN u is NB
R2: IF e is A_2 and Δe is B_2 and $\Delta^2 e$ is C_2 THEN u is PB

GP would depict these two rules as shown in Figure 8.57. In the above rules, A1, A2, B1, B2, C2, NB, and PB are fuzzy linguistic labels. The antecedents' variables are *error*, *first change of error*, and *second change of error*. The consequence is a control variable u. In Figure 8.57, the collection of symbolic segments,

ANT, CONSQ, F_AND, IF-THEN and F_OR

constitutes what is called the *function set* of the GP. The reader may con-

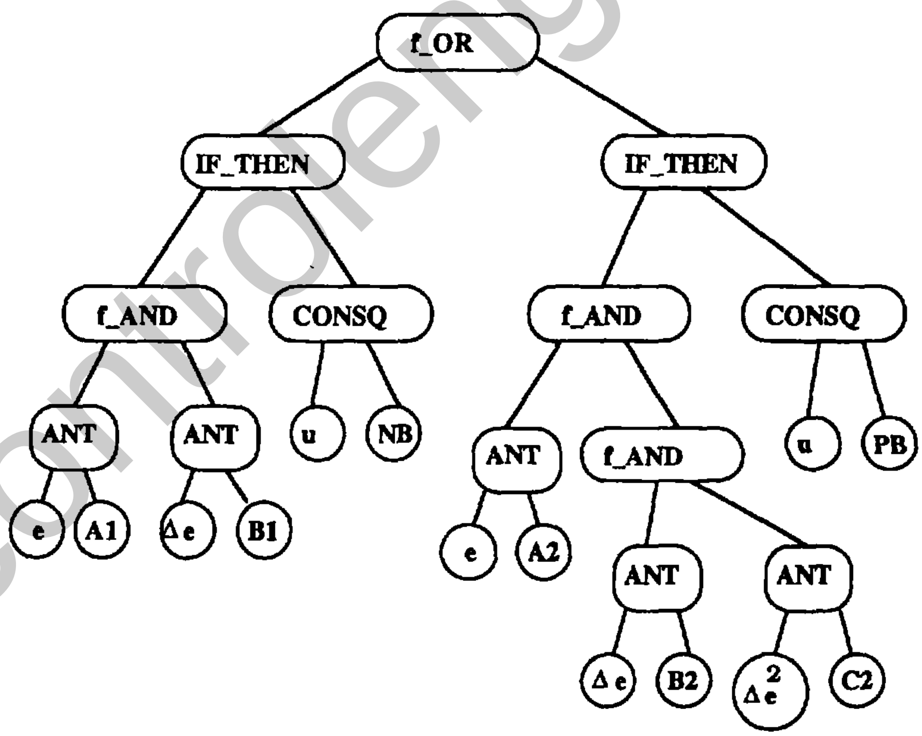


Figure 8.57 A two-rule-base tree depiction in genetic programming.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

sult literature for a more detailed study of GP.

Tunstel and Jamshidi (1996) used GP to evolve the rule base, while assuming that the membership functions are specified *a priori* and are fixed. They have decomposed the behavior of a mobile robot into a bottom-up hierarchy of increased behavioral complexity in which a behavior at a given level is dependent upon behaviors at the level(s) below. In this framework a finite set of primitive behaviors are combined synergistically to produce more intelligent composite behaviors in service of goal-directed operations. Primitive behaviors reside at the lowest level and are encoded as fuzzy rule bases with distinct control policies governed by fuzzy inference. As an example, Figure 8.58 illustrates a possible hierarchy for indoor navigation. The figure implies that goal-directed navigation can be decomposed as a behavioral function of goal seeking and route following. These behaviors can be further decomposed into the primitive behaviors with functional dependencies indicated by the adjoining lines. Wall following and obstacle avoidance are self-explanatory. The doorway behavior implies one that can guide a robot through narrow passageways with very shallow depth. The "goto" behavior implies one that will direct a robot to navigate along a straight-line trajectory to a particular location.

In developing the architecture, Tunstel *et al.* (1995) takes into consideration the notion that humans may not have the best solutions for designing knowledge-based controllers with interacting rule bases. The issue of just how much design should be imposed, and how much should be learned or evolved is unresolved at this time. As a tradeoff, good results have been achieved by employing GP to evolve a subset of the fuzzy behaviors (Tunstel and Jamshidi, 1996). GP evolution manipulates the linguistic variables directly associated with the behaviors and enhances diversity of a population of behaviors by allowing for rule sets of various sizes. The control scheme generalizes conventional *rule* conflict resolution (by fuzzy inference and defuzzification) to resolve conflicts among competing *behaviors*. The resultant control action can be viewed as a consensus of recommendations offered by multiple experts. Behavior coordination is achieved by a weighted, decision-making scheme in which fuzzy rules for composite behaviors govern the degree of applicability of primitive behaviors at a lower level. This feature allows certain robot behaviors to influence the overall behavior to a greater or lesser degree as required by the current goal. It serves as a form of adaptation since the control policy (behavior) dynamically changes in response to sensory input. Thus, behavior coordination is accomplished using meta-rules that provide a form of the ethological concept of inhibition observed in animal behavior.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

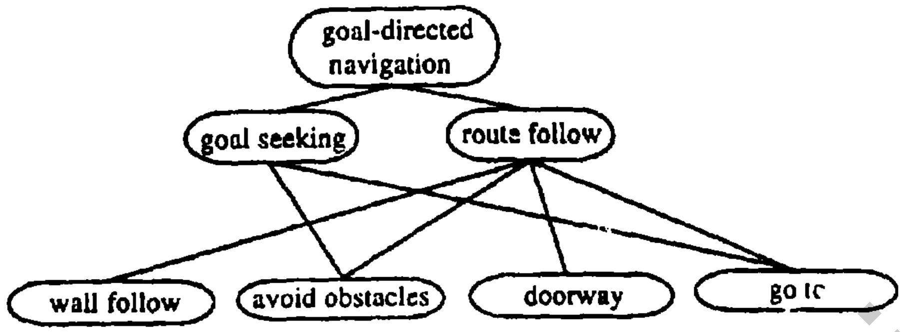


Figure 8.58 Example mobile robot behavior hierarchy.

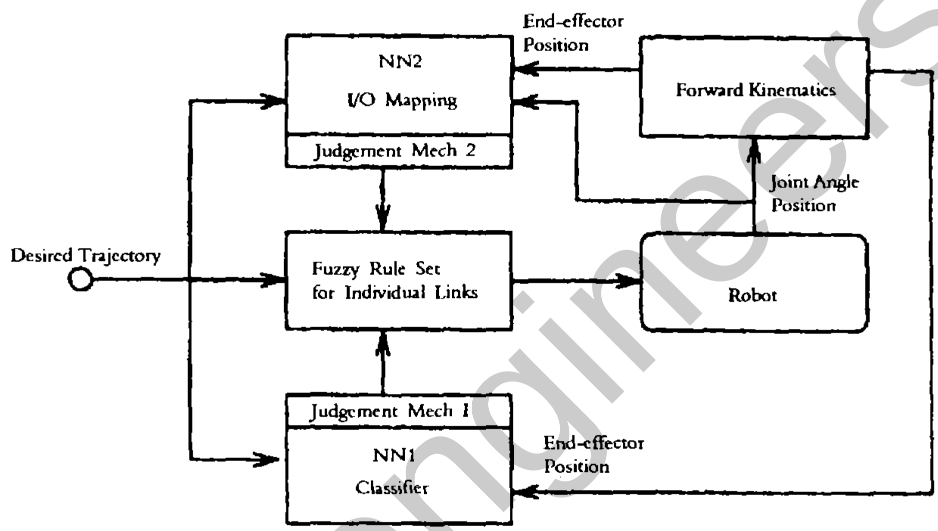


Figure 8.59 Schematic of neuro-fuzzy controller.

Within this framework, autonomous navigation is possible in both indoor and outdoor environments. Indoor experiments using physical mobile robots are underway (Tunstel and Jamshidi, 1996).

Fuzzy-NN Paradigm In this last paradigm, the learning capability of the neural network can be made use of in designing the fuzzy controller. The *self-organizing fuzzy controller* is one such combination of a neural network and a fuzzy controller. Figure 8.59 shows a schematic diagram of the system forming the self-organizing fuzzy controller. The aim of this system is to automatically form the fuzzy controller. It uses two neural networks of the *back propagation learning type*, NN1 and NN2. NN1 acts as a classifier of the dynamic responses of the system being controlled (robot system). NN2, set in judgment mechanism 2, has knowledge of the dy-

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

namic characteristics of the object system. Judgment mechanism 2 has a self-tuning mechanism to automatically determine the normalizing values of the membership functions to control the object system adequately.

In this particular case NN1 classifies the error in the trajectory of the end effector to several typical patterns such as a similar pattern to the desired response or an oscillating and diverging pattern or an oscillating and slowly converging pattern or any other pattern (Tunstel *et al.*, 1995). The result of the classification is sent to judgment mechanism 1. NN2 is made to learn the dynamic characteristics of the object system through pairs of input and system response. NN2 can then be used to simulate the object system in cases where it is too risky to control the object system with an incomplete fuzzy controller.

The judgment mechanism 1 decides whether a new fuzzy rule has to be formed and whether a tuning operation is needed for the scaling of membership functions of the currently established rules. Next, the labels of the fuzzy variables in the antecedent part of the fuzzy control rule are determined from the information of the dynamic response of the object system and the rule is integrated using the fuzzy variable in the consequent part whose label is calculated in the judgment mechanism 2. If there are no rules at the initial state in the fuzzy controller, then the judgment mechanism sets up some initial control value to control the object system. The object system is controlled by the established fuzzy rules and the information of the dynamic response is sent to NN1. The self-tuning operation of the normalizing values for currently established fuzzy control rules is repeated until the dynamic response is classified as the desired response by NN1. If the dynamic response of the object system is classified as the desired response by NN1, judgment mechanism 1 decides to form the next new fuzzy control rule. Judgment mechanism 2 mainly plays a role in

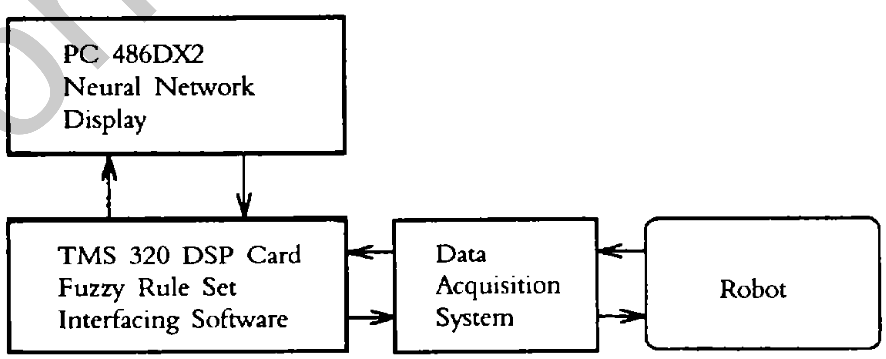


Figure 8.60 Implementation scheme of the neuro-fuzzy controller.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

determining the quantities for the label of the fuzzy variable in the consequent part and in tuning values for the fuzzy variables in both the antecedent and consequent parts.

The block diagram of the implementation scheme is shown in Figure 8.60. A Texas Instruments DSP card is used as a fuzzy controller. All the rules are downloaded into this card. This card communicates with the robot through a data acquisition system. The neural network software runs in the background using the PC processor. Information is passed back and forth from the DSP card and the PC. Further progress on this paradigm and the robotic controller will be forthcoming.

Problems

8.1 Using the parameter estimation via substitution approach discussed in Section 8.2.1, identify a fuzzy relation for a single-input, single-output system described by $y = 10 \cos(5\pi x)$.
Hint: See Example 8.2.1.

8.2. Repeat Example 8.2.2 for the voltage control of an RC circuit. Consider the dynamic equation,

$$c\frac{di}{dt} = u - Ri$$

where c is the capacitance (farad), R is the resistance (ohm), i is the current, and u is the voltage. Choose the inverses of discourse for i and u as

$$u_i = [-10, 10] \text{ and } U_u = [-30, 30]$$

Use a desired voltage value of $U_d(t) = 5 + 5\,(1 - \exp(-t / 10)$.

8.3. Let the dynamic equation of the tank fluid-level problem of (8.2.19) be replaced by

$$S\frac{dh(t)}{dt} = u(t) - s(2gh(t))$$

Repeat Example 8.2.3 using the gradient approach to centroid adaptation method.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

8.4. CADLAB Consider the dynamic model of a second-order inverted pendulum given by Equation (7.2.19) and let m = 100 grams, M = 1 kg, l = ½ meter, g = 9.8 m/sec., and desired angular position $\theta(t) = 1-\exp(-0.5t)$. Use your favorite computer software to simulate this system for a PD controller and for variations in mass m and length l. Use simulation results to devise fuzzy adaptation rules for gains K_p and K_d in terms of error e and its rate of change Δe. Simulate the inverted pendulum's adaptive controller for a time span of 5 seconds, while incorporating a 10% increase in mass m at $t = 2$ seconds and a 5% reduction in length l at $t = 3$ seconds. Use five membership functions for e, three for Δe, and use simulation results for K_p and K_d.

8.5. CADLAB Repeat Example 8.2.5 for $\theta_d(t) = \frac{\pi}{30}\sin t$, m = 150 grams, M = 1500 grams, and l = ½ meter. All other parameters remain the same.

8.6. CADLAB A fuzzy controller has six input variables and one output. If each input or output variable has three linguistic labels, how many rules are needed to represent the inference engine? How many rules would be required for a five-level hierarchy? What would be the minimum number of rules for sensory fusion?

8.7. CADLAB Physical systems may, in general, involve several sensory variables such as position, velocity, acceleration, thickness, height, voltage, current, power, etc. When attempting to fuse the sensory variables, it is often nontrivial to choose the right combination of variables. Can you devise a "rule of thumb" which would systematically allow one to fuse physical variables? *Hint:* Group physical variables, determine the interrelationships of each group, and establish aggregated variables per group of original variables.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Appendix A

A Brief Review of Fuzzy Set Theory

controlengineers.ir

A.1 Introduction

In this appendix a brief introduction is given to the *fuzzy set theory*. In support of Chapters 7 and 8, one has to be abreast of the basics of fuzzy sets. However, many introductions similar to this one have been published over the years in other reports, articles, and books on fuzzy logic and control.

In 1965, Zadeh (1965) wrote a seminal paper in which he introduced fuzzy sets, sets with *unsharp* boundaries.These sets are generally in better agreement with the human mind that works with shades of grey, rather than with just black or white. Fuzzy sets are typically able to represent linguistic terms, e.g., *warm*, *hot*, *high*, *low*. Nearly 10 years later Mamdani (1974) succeeded to apply fuzzy logic for control in practice. Today, in Japan and many other parts of the world fuzzy control is widely accepted and applied. In many consumer products like washing machines and cameras, fuzzy controllers are used in order to obtain higher machine IQ and user-friendly products. A few interesting applications can be mentioned: control of subway systems, image stabilization of video cameras, and autonomous control of helicopters. Although the United States and Europe hesitated in accepting fuzzy logic, they have become more enthusiastic about applying this technology.

In Section A.2 the differences between fuzzy sets and classical crisp sets are discussed. Section A.3 deals with the operations on fuzzy sets that are used for fuzzy control. In Section A.4 *fuzzy logic* as a logical system is explained. This introduction is meant to provide the basic knowledge that is indispensible for understanding background materials.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

A.2 Fuzzy Sets versus Crisp Sets

In the classical set theory, a set is denoted as a so-called *crisp set* and can be described by its characteristic function as follows:

$$\mu_C : U \rightarrow \{0,1\} \tag{A.1}$$

In Equation (A.1) U is called the *universe of discourse,* i.e., a collection of elements that can be continuous or discrete. In a crisp set each element of the universe of discourse either belongs to the crisp set ($\mu_C = 1$) or does not belong to the crisp set ($\mu_C = 0$).

Consider a characteristic function μ_{Chot} representing the crisp set *hot,* a set with all "hot" temperatures. Figure A. 1 graphically describes this crisp set, considering temperatures higher than 25 degrees Celsius as hot. (Note that for all the temperatures t we have $t \in U$.)

The definition of a *fuzzy set* (Zadeh, 1965) is given by the characteristic function

$$\mu_F : U \Rightarrow [0,1] \tag{A.2}$$

In this case the elements of the universe of discourse can belong to the fuzzy set with any value between 0 and 1. This value is called the *degree of membership.* If an element has a value close to 1, the degree of membership, or "truth" value is high. The characteristic function of a fuzzy set is called the *membership function,* for it gives the degree of membership for each element of the universe of discourse. If now the characteristic function of μ_{Fhot} is considered, one can express the human opinion, for example, that 24 degrees is still fairly hot, and that 26 degrees is hot, but not as hot as 30 degrees and higher. This results in a gradual transition from membership (completely true) to nonmembership (not true at all). Figure A.2 shows the membership function μ_{Fhot} of the fuzzy set F_{hot}.

controlengineers.ir

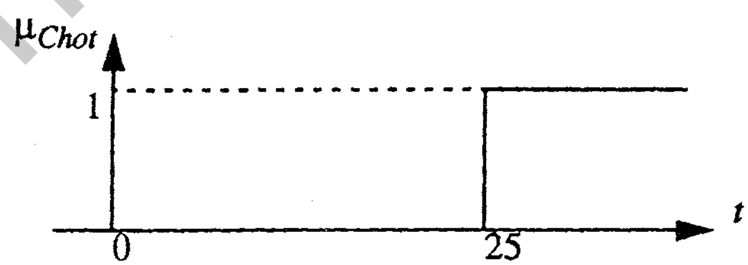


Figure A.1. The characteristic function μ_{Chot}.

controlengineers.ir
@controlengineers

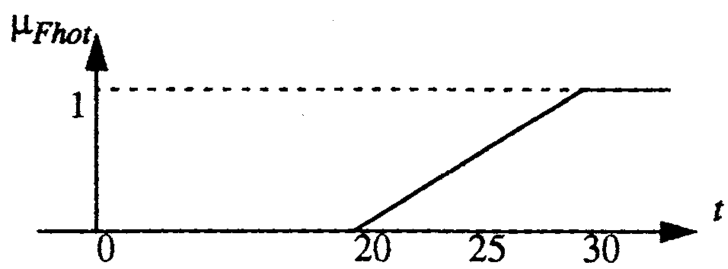


Figure A.2. The membership function μ_{Fhot}.

In this figure the membership function has a linear transition. However, every individual can constuct a different transition according to his own opinion. Membership functions can have many possible shapes. In practice the transitions are often linearized to simplify the computations.

Example A.1 Suppose someone wants to describe the class of cars having the property of being expensive by considering cars such as BMW, Buick, Cadillac, Ferrari, Fiat, LADA, Mercedes, Nissan, Peugot, and Rolls Royce. Describe its fuzzy set "expensive cars."

SOLUTION: Some cars, like Ferrari or Rolls Royce, definitely belong to the class "expensive," while other cars, like Fiat or LADA do not belong to it. But there is a third group of cars, which are not really expensive, but which are also not cheap. Using fuzzy sets, the fuzzy set of "expensive cars" is, e.g.,

{(*Ferrari, 1*),(*Rolls Royce, 1*),(*Mercedes, 0.9*),(*BMW,0.8*),(*Cadillac, 0.8*), (*Nissan, 0.7*),(*Buick,0.6*),(*Peugot, 0.5*),(*Fiat,0.2*),(*LADA,0.1*)}

Example A.2 Suppose someone wants to define the set of natural numbers "close to 5." It is desired to find a fuzzy set representation.

SOLUTION:This can be expressed in the discrete case by the fuzzy set:

$$\underset{\sim}{5} = (3, 0.2) + (4, 0.8) + (5, 1) + (6, 0.8) + (7, 0.2)$$

The underscore ~ under number 5 designates fuzziness. The membership function in the continuous case of the fuzzy set of real numbers "close

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

to 5" is e.g.,

$$\mu_{\underset{\sim}{5}} = \frac{1}{1+(x-5)^2} \tag{A.3}$$

and the fuzzy set $\underset{\sim}{5}$ contains, e.g., the elements (5,1), (6,0.8) In general we denote any countable or discrete universe X a fuzzy set with

$$A = \sum_{x_i \in X} \mu_A(x_i)/x_i \tag{A.4}$$

or for the continuous case:

$$A = \int_{x \in X} \mu_A(x_i)/x_i \tag{A.5}$$

Note that the $\int$ sign doesn't denote the mathematical integral.

A.3 The Shape of Fuzzy Sets

The membership function of a fuzzy set can have different shapes. This depends on its definition. Membership functions with straight lines with the top as an interval are called Π-functions (trapezoidal function) (see Figure A.3).

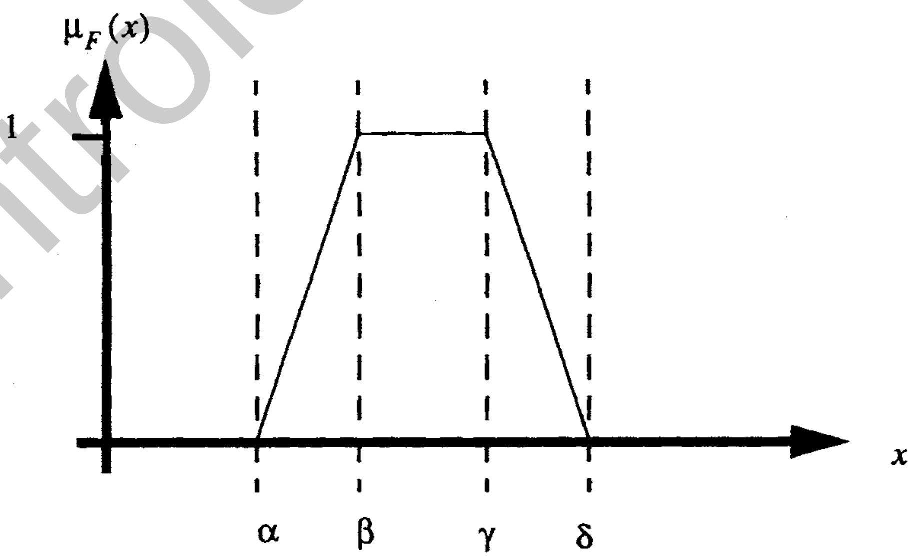


Figure A.3. Example of a Π -function.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

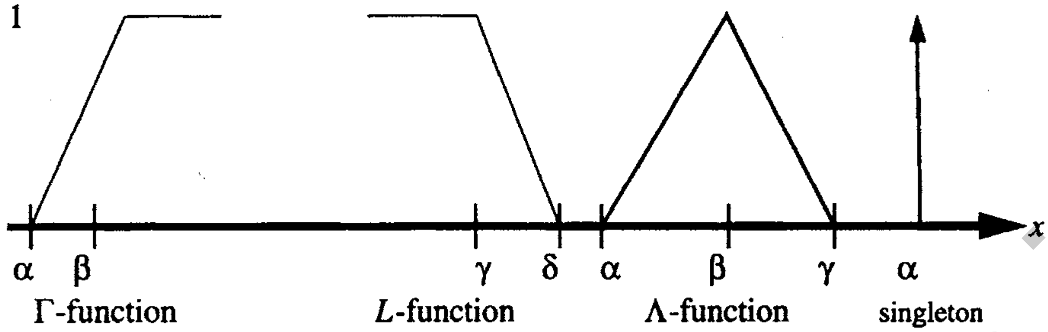


Figure A.4. Examples of the Γ, *L*, Λ, and singleton.

Other common shapes and forms are shown in Figure A.4, above.

Definition A.1 The function Π: $X \to [0,1]$ is defined by four parameters $(\alpha, \beta, \gamma, \delta)$:

$$\Pi(x;\alpha,\beta,\gamma,\delta)=\begin{cases}0 & x<\alpha\\(x-\alpha)/(\beta-\alpha) & \alpha\le x\le\beta\\1 & \beta\le x\le\gamma\\1-(x-\gamma)/(\delta-\gamma) & \gamma\le x\le\delta\\0 & x>\delta\end{cases} \tag{A.6}$$

Further we have a decreasing membership function with straight lines, the *L*-function, an increasing membership function with straight lines, the Γ-function, a triangular function with straight lines, the Λ-function, and a membership function with the membership function value 1 for only one value and the rest zero, the singleton. They are all special cases of the Π-function. This is shown in the equations (A.7)–(A.10). Suppose that the underlying domain is [–6,6] then the following equations hold:

$$\Gamma(x;\alpha,\beta)=\Pi(x;\alpha,\beta,6,6) \tag{A.7}$$

$$L(x;\gamma,\delta)=\Pi(x;-6,-6,\gamma,\delta) \tag{A.8}$$

$$\Lambda(x;\alpha,\beta,\delta)=\Pi(x;\alpha,\beta,\beta,\delta) \tag{A.9}$$

$$\text{singleton}\ (x;\alpha)=\Pi(x;\alpha,\alpha,\alpha,\alpha) \tag{A.10}$$

controlengineers.ir

controlengineers.ir
@controlengineers

A.4 Fuzzy Sets Operations

As in the traditional crisp sets, logical operations, e.g., union, intersection, and complement, can be applied to fuzzy sets (Zadeh, 1965). Since some of these operations are used for fuzzy control, the necessary operations are discussed in this section.

Union The union operation (and the intersection operation as well) can be defined in many different ways. Here, the definition that is used in most cases is discussed. The union of two fuzzy sets A and B with the membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C, written as $C = A \cup B$, whose membership function is related to those of A and B as follows:

$$\forall x \in U: \mu_C(x) = \max[\mu_A(x), \mu_B(x)] \tag{A.11}$$

where U is the universe of discourse. The operator in this equation is referred to as the *max-operator.*

Intersection According to the *min-operator,* the intersection of two fuzzy sets A and B with the membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively, is a fuzzy set C, written as $C = A \cap B$, whose membership function is related to those of A and B as follows:

$$\forall x \in U: \mu_C(x) = \min[\mu_A(x), \mu_B(x)] \tag{A.12}$$

Both the intersection and the union operation are explained by Figure A.5.

Complement: The complement of a fuzzy set A is denoted $\bar{A}$ as with a membership function defined as (see also Figure A.6):

$$\forall x \in U: \mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{A.13}$$

Most of the properties that hold for classical sets (e.g., commutativity, associativity, and idempotence) hold also for fuzzy sets, manipulated by the specific operations in equations (A.11)-(A.13), except for two properties:

1. *Law of contradiction* $(A \cap \bar{A} \neq \varnothing)$: One can easily notice that the intersection of a fuzzy set and its complement results in a fuzzy sets with membership values up to 1/2 and thus does not equal the

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

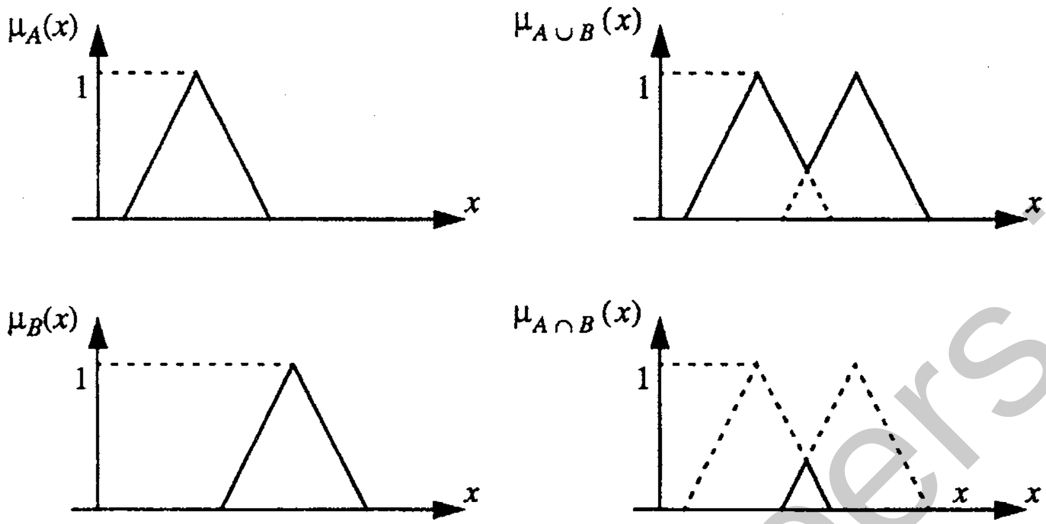


Figure A.5. The fuzzy set operations union and intersection.

empty set (see the equation below and Figure A.7).

$$\forall x \in U{:}\, \mu_{A\cap\bar{A}}(x) = \min\left[\mu_A(x), \left(1-\mu_A(x)\right)\right] \le \frac{1}{2} \ne \varnothing \qquad \text{(A.14)}$$

2. *Law of excluded middle* $\left(A \cup \bar{A} \ne U\right)$: The union of a fuzzy set and its complement does not give the universe of discourse (see Figure A.7).

$$\forall x \in U{:}\, \mu_{A\cap\bar{A}}(x) = \max\left[\mu_A(x), \left(1-\mu_A(x)\right)\right] \le \frac{1}{2} \ne U \qquad \text{(A.15)}$$

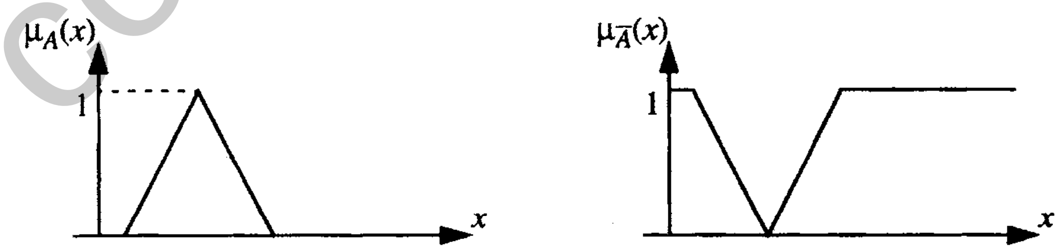
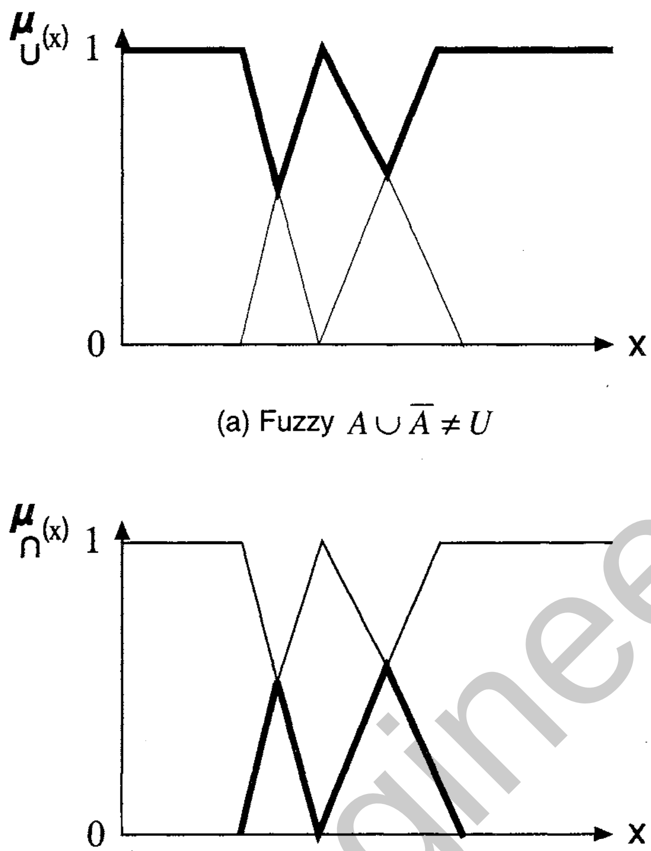


Figure A.6. Fuzzy set and its compliment.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir



(a) Fuzzy $A \cup \overline{A} \neq U$

(b) Fuzzy $A \cap \overline{A} \neq \varnothing$

Figure A.7. Excluded middle laws for fuzzy sets.

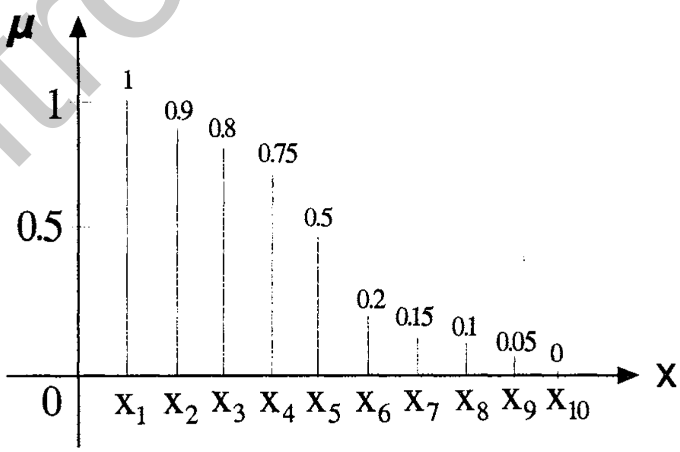


Figure A.8. Fuzzy set *A* of Example A.3.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Alpha-cut Fuzzy Sets It is the crisp domain in which we perform all computations with today's computers. The conversion from fuzzy to crisp sets can be done by two means—*defuzzification* and *alpha-cut sets*. The former is presented in Chapter 7, while the latter is defined here.

Definition A.2 Given a fuzzy set A, the alpha-cut (or lambda-cut) set of A is defined by

$$A_\alpha = \{x | \mu_A(x) \geq \alpha\} \tag{A.16}$$

Note that by virtue of the condition on $\mu_A(x)$ in Equation (A.16), i.e., a common property, the set A_α in (A.16) is now a crisp set. In fact, any fuzzy set can be converted to an infinite number of cut sets.

Example A.3 Consider a fuzzy set

$$A = \left\{ \frac{1}{x_1} + \frac{0.9}{x_2} + \frac{0.8}{x_3} + \frac{0.75}{x_4} + \frac{0.5}{x_5} + \frac{0.2}{x_6} + \frac{0.15}{x_7} + \frac{0.1}{x_8} + \frac{0.05}{x_9} + \frac{0}{x_{10}} \right\}$$

It is desired to find the number of α-cut sets.

SOLUTION: The fuzzy set A is shown in Figure A.8. The α-cut sets A_1, $A_{0.8}$, $A_{0.5}$, $A_{0.1}$, A_{0+} and A_0 are defined by

$$A_1 = \{x_1\}, \quad A_{0.8} = \{x_1, x_2, x_3\}$$
$$A_{0.5} = \{x_1, x_2, \ldots, x_5\}$$
$$A_{0.1} = \{x_1, x_2, \ldots, x_8\}$$
$$A_{0^+} = \{x_1, x_2, \ldots, x_9\} \text{ and } A_0 = U$$

Note that, by definition, the 0-cut set A_0 is the universe of discourse. Figure A.9 shows these α-cut sets.

Extension Principle In fuzzy sets, just as in crisp sets, one needs to find a means to extend the domain of a function, i.e., given a fuzzy set A and a function $f(\cdot)$, then what is the value of function $f(A)$? This notion is called the *extension principle* which was first proposed by Zadeh (Jamshidi *et al.*, 1993).

controlengineers.ir
@controlengineers

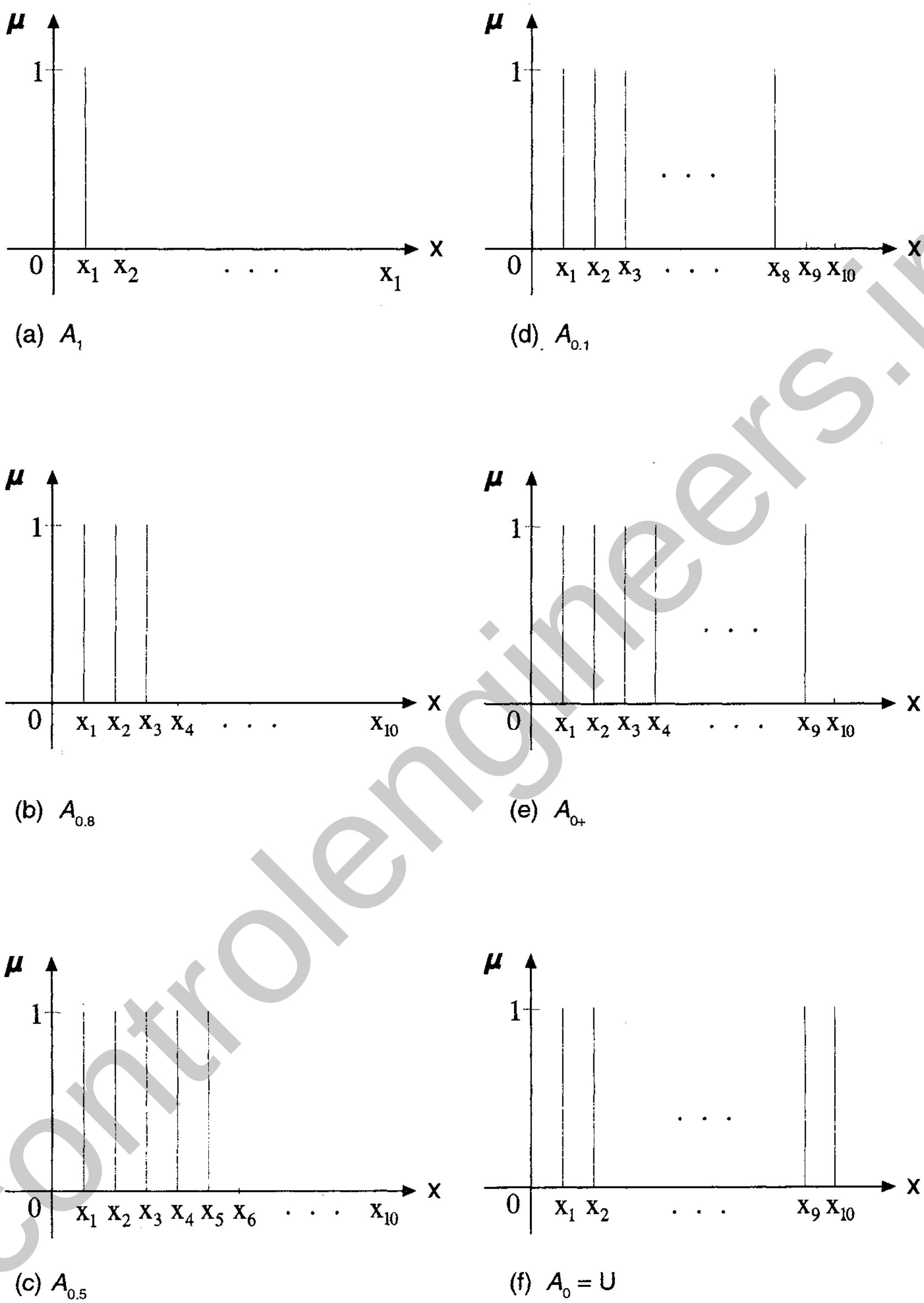


Figure A.9. Schematic of 5 α-cut sets for fuzzy set of Example A.3.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Let the function f be defined by

$$f : U \rightarrow V \tag{A.17}$$

where U and V are domain and range sets, respectively. Define a fuzzy set $A \subset U$ as,

$$A = \left\{ \frac{\mu_1}{u_1} + \frac{\mu_2}{u_2} + \cdots + \frac{\mu_n}{u_n} \right\} \tag{A.18}$$

Then the extension principle asserts that the function f is a fuzzy set, as well, which is defined below:

$$B = f(A) = \frac{\mu_1}{f(u_1)} + \frac{\mu_2}{f(u_2)} + \cdots + \frac{\mu_n}{f(u_n)} \tag{A.19}$$

In other words, the resulting fuzzy set has the same membership values corresponding to the functions of the elements u_i, $i = 1, 2, \ldots, n$. The following two examples illustrate the use of the *extension principle*.

Example A.4 Given two universes of discourse $U_1 = U_2 = \{1, 2, \ldots, 10\}$ and two fuzzy sets (numbers) defined by

"Approximately 2" = $\underset{\sim}{2} = \frac{0.5}{1} + \frac{1}{2} + \frac{0.8}{3}$

and

"Almost 5" = $\underset{\sim}{5} = \frac{0.6}{3} + \frac{0.8}{4} + \frac{1}{5}$

It is desired to find "approximately 10," i.e., $\underset{\sim}{10} = \underset{\sim}{2} \times \underset{\sim}{5}$.

SOLUTION: The function $f = u_1 \times u_2 : \rightarrow v$ represents the arithmatic product of these two fuzzy numbers is given by

$$\begin{aligned}
\underset{\sim}{10} = \underset{\sim}{2} \times \underset{\sim}{5} &= \left(\frac{0.5}{1} + \frac{1}{2} + \frac{0.8}{3} \right) \times \left(\frac{0.6}{3} + \frac{0.8}{4} + \frac{1}{5} \right) = \frac{\min(0.5, 0.6)}{3} \\
&+ \frac{\min(0.5, 0.8)}{4} + \frac{\min(0.5, 1)}{5} + \frac{\min(1, 0.6)}{6} + \frac{\min(1, 0.8)}{8} \\
&+ \frac{\min(1, 1)}{10} + \frac{\min(0.8, 0.6)}{9} + \frac{\min(0.8, 0.8)}{12} + \frac{\min(0.8, 1)}{15} \\
&= \frac{0.5}{3} + \frac{0.5}{4} + \frac{0.5}{5} + \frac{0.6}{6} + \frac{0.8}{8} + \frac{0.6}{9} + \frac{1}{10} + \frac{0.8}{12} + \frac{0.8}{15}
\end{aligned}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The above resulting fuzzy number has its *prototype*, i.e., value 10 with a membership function 1 and the other 8 pairs are spread around the point (1, 10).

The complexity of the extension principle would increase when more than one member of $u_1 \times u_2$ is mapped to only one member of v; one would take the maximum membership grades of these members in the fuzzy set A. The following example illustrates this case.

Example A.5 Consider the two fuzzy numbers,

$$\underset{\sim}{2} = \text{“approximately 2”} = \frac{0.5}{1} + \frac{1}{2} + \frac{0.5}{3}$$

$$\underset{\sim}{4} = \text{“approximately 4”} = \frac{0.8}{2} + \frac{0.9}{3} + \frac{1}{4}$$

It is desired to find $\underset{\sim}{8}$.

SOLUTION: The product $\underset{\sim}{2} \times \underset{\sim}{4}$ would be given by the following expression:

$$\underset{\sim}{2} \times \underset{\sim}{4} = \frac{\min(0.5, 0.8)}{2} + \frac{\min(0.5, 0.9)}{3} + \frac{\max\{\min(0.5, 1), \min(1, 0.8)\}}{4}$$
$$+ \frac{\max\{\min(1, 0.9), \min(0.5, 0.8)\}}{6} + \frac{\min(1, 1)}{8} + \frac{\min(0.5, 0.9)}{9}$$
$$+ \frac{\min(0.5, 1)}{12} = \frac{0.5}{2} + \frac{0.5}{3} + \frac{0.8}{4} + \frac{0.9}{6} + \frac{1}{8} + \frac{0.5}{9} + \frac{0.5}{12}$$

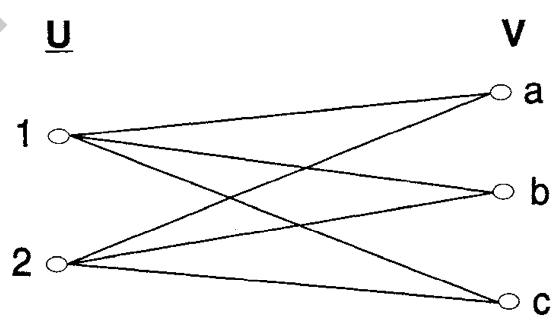


Figure A.10. A crisp Sagittal diagram.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Fuzzy Relations Consider the Cartesian product of two universes U and V, defined by

$$U \times V = \{(u, v) | u \in U, v \in V\} \tag{A.20}$$

which combines elements of U and V in a set of ordered pairs. As an example, if $U = \{1, 2\}$ and $V = \{a, b, c\}$, then $U \times V = \{ (1, a), (1, b), (1, c), (2, a), (2, b), (2, c) \}$. The above product is said to be a *crisp relation* which can be expressed by either a matrix expression

$$R_c = U \times V = \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \tag{A.21}$$

or in a so-called *Sagittal* diagram (see Figure A.10) (Jamshidi *et al.*, 1993, Chapter 2).

In classical set relations, one can perform operations on crisp relations using *max-min composition*, similar to those in Example A.5.

The fuzzy relations, similarly, map elements of one universe, say U, to elements of another universe V through Cartesian product, but the "strength" of relationship is measured by the grade of a membership function (Ross, 1995). In other words, a fuzzy relation R is a mapping:

$$R : U \times V \rightarrow [0, 1] \tag{A.22}$$

The following example illustrates this relationship, i.e.,

$$\mu_R(u, v) = \mu_{A \times B}(u, v) = \min(\mu_A(u), \mu_B(v)) \tag{A.23}$$

Example A.6 Consider two fuzzy sets $A_1 = \frac{0.2}{x_1} + \frac{0.9}{x_2}$ and $A_2 = \frac{0.3}{y_1} + \frac{0.5}{y_2} + \frac{1}{y_3}$. Determine the fuzzy relation among these sets.

SOLUTION: The fuzzy relation R, using (A.23), is

$$R = A_1 \times A_2 = \begin{bmatrix} 0.2 \\ 0.9 \end{bmatrix} \times \begin{bmatrix} 0.3 & 0.5 & 1 \end{bmatrix} = \begin{bmatrix} \min(0.2, 0.3) & \min(0.2, 0.5) & \min(0.2, 1) \\ \min(0.9, 0.3) & \min(0.9, 0.5) & \min(0.9, 1) \end{bmatrix}$$

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

$$= \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.9 \end{bmatrix}$$

In crisp or fuzzy relations, the composition of two relations, using the max-min rule, is given below. Given two fuzzy relations $R(u, v)$ and $S(v, w)$, then the composition of these is

$$T = R \circ S \text{ with } \mu_T(u,w) = \max_{v \in V}\left\{\min\left(\mu_R(u,v),\ \mu_S(v,w)\right)\right\}$$

or using the max-product rule, the characteristic function is given by

$$\mu_T(u,w) = \max_{v \in V}\left\{\mu_R(u,v) \cdot \mu_S(v,w)\right\}$$

The same compositional rules hold for crisp relations. In general $R \circ S \neq S \circ R$. The following example illustrates this.

Example A.7 Consider two fuzzy relations

$$R = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \!\! \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.6 & 0.8 \\ 0.7 & 0.9 \end{bmatrix} \end{array} \text{ and } S = \begin{array}{c} \\ y_1 \\ y_2 \end{array} \!\! \begin{array}{c} \begin{array}{cc} z_1 & z_2 \end{array} \\ \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \end{array}$$

It is desired to evaluate $R \circ S$ and $S \circ R$.

SOLUTION: Using the max-min composition we have

$$T_1 = R \circ S = \begin{bmatrix} 0.3 & 0.8 \\ 0.3 & 0.8 \end{bmatrix}$$

where, for example, the (1, 1) element is obtained by max {min (0.6, 0.3), min (0.8, 0.2)} = 0.3.

The max-min composition of $S \circ R$ results in

$$S \circ R = \begin{bmatrix} 0.3 & 0.3 \\ 0.7 & 0.8 \end{bmatrix} \neq R \circ S$$

which is expected.

Using the max-product rule, we have

$$T_2 = R \circ S = \begin{bmatrix} 0.18 & 0.64 \\ 0.21 & 0.72 \end{bmatrix}$$

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

where, for example, the term (2, 2) is obtained by $\max\{(0.7)\,(0.1)\,,\,(0.9)\,(0.8)\} = 0.72$.

The max-product composition $S \circ R$ results in

$$S \circ R = \begin{bmatrix} 0.3 & 0.3 \\ 0.7 & 0.8 \end{bmatrix} \neq R \circ S$$

which is, once again, expected.

A.5 Fuzzy Logic and Approximate Reasoning

In this final section of this appendix, a brief introduction to fuzzy logic and approximate reasoning is given. Parts of this section are based on the work of Ross (1995).

Predicate Logic Let a predicate logic proposition P be a linguistic statement contained within a universe of propositions which are either completely *true* or *false*.

The truth value of the proposition, P, can be assigned a binary truth value, called $T(P)$, just as an element in a universe is assigned a binary quantity to measure its membersnip in a particular set. For binary (Boolean) predicate logic, $T(P)$ is assigned a value of 1 (truth) or 0 (false). If U is the universe of all propositions, then T is a mapping of these propositions to the binary quantities (0, 1), or

$$T : U \rightarrow \{0,1\}$$

Now let P and Q be two simple propositions on the same universe of discourse that can be combined using the following five logical connectives,

(i) disjunction ($\vee$)
(ii) conjunction ($\wedge$)
(iii) negation ($-$)
(iv) implication ($\rightarrow$)
(v) equality ($\leftrightarrow$ *or* $\equiv$)

to form logical expressions involving two simple propositions. These connectives can be used to form new propositions from simple propositions.

Now define sets A and B from universe X where these sets might represent linguistic ideas or thoughts. Then a *propositional calculus* will exist

controlengineers.ir
@controlengineers

for the case where proposition P measures the truth of the statement that an element, x, from the universe X is contained in set A and the truth of the statement that this element, x, is contained in set B, or more conventionally

P: truth that $x \in A$

Q: truth that $x \in B$, where truth is measured in terms of the truth value, i.e.,

If $x \in A$, $T(P) = 1$; otherwise $T(P) = 0$.

If $x \in B$, $T(Q) = 1$; otherwise $T(Q) = 0$, or using the characteristic function to represent truth (1) and false (0),

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

The above five logical connectives can be used to create compound propositions, where a compound proposition is defined as a logical proposition formed by logically connecting two or more simple propositions. Just as one is interested in the truth of a simple proposition, predicate logic also involves the assessment of the truth of compound propositions. For two simple proposition cases, the resulting compound propositions are defined below in terms of their binary truth values,

$P: x \in A,\ \overline{P}: x \notin A$

$P \vee Q \Rightarrow x \in A \text{ or } B$

$\qquad \textit{Hence},\ T(P \vee Q) = \max(T(P), T(Q))$

$P \wedge Q \Rightarrow x \in A \text{ and } B$

$\qquad \textit{Hence},\ T(P \wedge Q) = \min(T(P), T(Q))$

$\textit{If } T(P) = 1, \textit{ then } T(\overline{P}) = 0; \textit{ If } T(P) = 0, \textit{ then } T(\overline{P}) = 1$

$P \leftrightarrow Q \Rightarrow x \in A, B$

$\qquad \textit{Hence},\ T(P \leftrightarrow Q) = T(P)$

$\qquad\qquad\qquad\qquad\quad = T(Q)$

The logical connective "implication" presented here is also known as the classical implication, to distinguish it from an alternative form due to Lukasiewicz, a Polish mathematician in the 1930s, who was first credited with exploring logics other than Aristotelian (classical or binary logic) logic. This classical form of the implication operation requires some explanation.

For a proposition P defined on set A and a proposition Q defined on set

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

B, the implication "P implies Q" is equivalent to taking the union of elements in the complement of set A with the elements in the set B. That is, the logical implication is analogous to the set-theoretic form,

$$P \to Q \equiv \overline{A} \cup B \;\; \textit{is true} \equiv \textit{either "not in A" or "in B"}$$

So that $(P \to Q) \leftrightarrow (\overline{P} \vee Q)$

$$T(P \to Q) = T(\overline{P} \vee Q) = \max(T(\overline{P}), T(Q))$$

This is linguistically equivalent to the statement, "P implies Q is true" when either "not A" or "B" is true. Graphically this implication and the analogous set operation is represented by the Venn diagram in Figure A.11 As noted, the region represented by the difference $A \setminus B$ is the set region where the implication "P implies Q" is false (the implication "fails"). The shaded region in Figure A.11 represents the collection of elements in the universe where the implication is true, i.e., the shaded area is the set

$$\begin{aligned} &\overline{A \setminus B} = \overline{A} \cup B = \left(\overline{A \cap \overline{B}}\right) \\ &\textit{If } x \textit{ in } A \; \underline{\textit{and}} \; x \textit{ is not in } B \textit{ then} \\ &A \to B \equiv \textit{fails } A \setminus B \; \textit{(difference)} \end{aligned} \tag{A.24}$$

Now, with two propositions (P and Q) each being able to take on one of two truth values (true or false, 1 or 0), there will be a total of $2^2 = 4$ propositional situations. These situations as illustrated in Table A.1, along with the appropriate truth values, for the propositions P and Q and the various logical connectives between them in the truth table above.

Suppose the implication operation involves two different universes of

controlengineers.ir

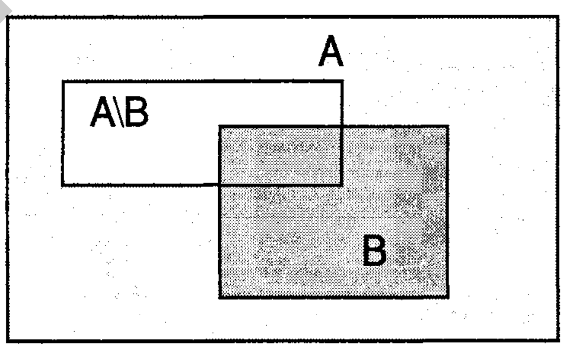


Figure A.11. Venn diagram for implication $P \to Q$.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

TABLE A.1

P	Q	$\overline{P}$	$P \vee Q$	$P \wedge Q$	$P \to Q$	$P \leftrightarrow Q$
T(1)	T(1)	F(0)	T(1)	T(1)	T(1)	T(1)
T(1)	F(0)	F(0)	T(1)	F(0)	F(0)	F(0)
F(0)	T(1)	T(1)	T(1)	F(0)	T(1)	F(0)

IF A, THEN B

This rule could be translated into a relation using the Cartesian product of sets A and B, that is

$$\mathrm{R} = \mathrm{A} \times \mathrm{B}$$

Now suppose a new antecedent, say A' is known. Can we use *Modus Ponens* deduction to infer a new consequent, say B', resulting from the new antecedent? That is, in rule form

IF A': THEN B' ?

The answer, of course, is yes, through the use of the composition relation. Since "A implies B" is defined on the Cartesian space $X \times Y$, B' can be found through the following set-theoretic formulation,

$$B' = A' \circ R = A' \circ \Big((A \times B) \cup (\overline{A} \times Y) \Big) \tag{A.29}$$

Modus Ponens deduction can also be used for the compound rule,

IF A, THEN B, ELSE C

using the relation defined as,

$$R = (A \times B) \cup (\overline{A} \times C) \tag{A.30}$$

and hence $B' = A' \circ R$.

Example A.8 Let two universes of discourse described by $X = \{1, 2, 3, 4, 5, 6\}$and $Y = \{1, 2, 3, 4\}$ and define the crisp set $A = \{3, 4\}$ on X and $B = \{2, 3\}$ on Y. Determine the deductive inference IF A, THEN B.

SOLUTION: The deductive inference yields the following characteristic function in matrix form, following the relation,

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ 1 & \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

controlengineers.ir
@controlengineers

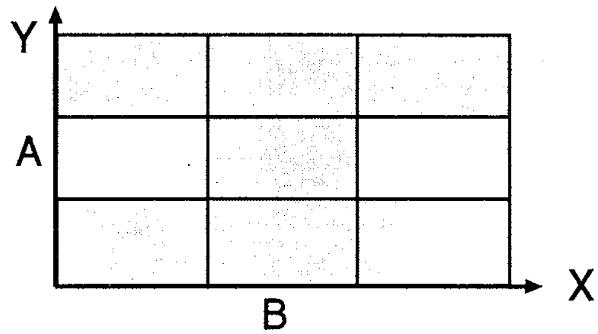


Figure A.12. The Cartesian space for the implication IF *A*, THEN *B*.

inference scheme used in forward chaining rule-based expert systems. It is an operation whose task is to find the truth value of a consequent in a production rule, given the truth value of the antecedent in the rule. Modus Ponens deduction concludes that, given two propositions, a and a-implies-b, both of which are true, then the truth of the simple proposition b is automatically inferred. Another useful tautology is the *Modus Tollens* inference, which is used in backward-chaining expert systems. In Modus Tollens an implication between two propositions is combined with a second proposition and both are used to imply a third proposition. Some common tautologies are listed below.

$$\overline{B} \cup B \leftrightarrow X \qquad (A \wedge (A \rightarrow B)) \rightarrow B \quad (\textit{Modus Ponens}) \tag{A.26}$$

$$\begin{aligned} &A \cup X; \quad \overline{A} \cup X \leftrightarrow X \quad (\overline{B} \wedge (A \rightarrow B)) \rightarrow \overline{A} \quad (\textit{Modus Tollens}) \\ &A \leftrightarrow B \end{aligned} \tag{A.27}$$

Contradictions Compound propositions that are always false, regardless of the truth value of the individual simple propositions comprising the compound proposition, are called contradictions. Some simple contradictions are listed below.

$$\begin{aligned} &\overline{B} \cap B \\ &A \cap \phi; \overline{A} \cap \phi \end{aligned} \tag{A.28}$$

Deductive Inferences The *Modus Ponens* deduction is used as a tool for inferencing in rule-based systems. A typical IF–THEN rule is use to determine whether an antecedent (cause or action) infers a consequent (effect or action). Suppose we have a rule of the form,

controlengineers.ir

controlengineers.ir
@controlengineers

IF A, THEN B

This rule could be translated into a relation using the Cartesian product of sets A and B, that is

$$R = A \times B$$

Now suppose a new antecedent, say A' is known. Can we use *Modus Ponens* deduction to infer a new consequent, say B', resulting from the new antecedent? That is, in rule form

IF A': THEN B' ?

The answer, of course, is yes, through the use of the composition relation. Since "A implies B" is defined on the Cartesian space $X \times Y$, B' can be found through the following set-theoretic formulation,

$$B' = A' \circ R = A' \circ \left((A \times B) \cup \left(\bar{A} \times Y \right) \right) \tag{A.29}$$

Modus Ponens deduction can also be used for the compound rule,

IF A, THEN B, ELSE C

using the relation defined as,

$$R = (A \times B) \cup \left(\bar{A} \times C \right) \tag{A.30}$$

and hence $B' = A' \circ R$.

Example A.8 Let two universes of discourse described by $X = \{1, 2, 3, 4, 5, 6\}$ and $Y = \{1, 2, 3, 4\}$ and define the crisp set $A = \{3, 4\}$ on X and $B = \{2, 3\}$ on Y. Determine the deductive inference IF A, THEN B.

SOLUTION: The deductive inference yields the following characteristic function in matrix form, following the relation,

$$R = (A \times B) \cup \left(\bar{A} \times Y \right) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Fuzzy Logic The extension of the above discussions to fuzzy deductive inference is straightforward. The fuzzy proposition P has a value on the

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

closed interval [0, 1]. The truth value of a proposition P is given by

$$T(P) = \mu_A(x) \text{ where } 0 \le \mu_A \le 1$$

Thus, the degree of truth for $P: x \in A$ is the membership grade of x in A. The logical connectives of negation, disjunction, conjunction, and implication are similarly defined for fuzzy logic, e.g., disjunction

$$P \vee Q \Rightarrow x \text{ is } A \text{ or } B$$

$$T(P \vee Q) = \max\left(T(P), T(Q)\right)$$

the implication is given by

$$P \to Q \Rightarrow x \text{ is } A \text{ THEN } x \text{ is } B$$

or

$$T(P \to Q) = T\left(\overline{P} \vee Q\right) = \max\left(T\left(\overline{P}\right), T(Q)\right)$$

Thus, a fuzzy logic implication would result in a fuzzy rule

$$P \to Q \Rightarrow \quad \text{If } x \text{ is } A \text{ THEN } y \text{ is } B$$

and is eqivalent to the following fuzzy relation

$$R = (A \times B) \cup \left(\overline{A} \times Y\right) \tag{A.31}$$

with a grade membership function,

$$\mu_R(x, y) = \max\left\{\left(\mu_A(x) \wedge \mu_B(y)\right), \left(1 - \mu_A(x)\right)\right\}$$

Example A.9 Consider two universes of discourse $X = \{1, 2, 3, 4\}$ and $Y = \{1, 2, 3, 4, 5, 6\}$. Let two fuzzy sets A and B be given by

$$A = \frac{0.8}{2} + \frac{1}{3} + \frac{0.3}{4}$$

$$B = \frac{0.4}{2} + \frac{1}{3} + \frac{0.6}{4} + \frac{0.2}{5}$$

It is desired to find a fuzzy relation R corresponding to IF A, THEN B.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

SOLUTION: Using the relation in (A.31) would give

$$A \times B = \begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.8 & 0.6 & 0.2 & 0 \\ 0 & 0.4 & 1 & 0.6 & 0.2 & 0 \\ 0 & 0.3 & 0.3 & 0.3 & 0.2 & 0 \end{bmatrix} \end{array}$$

$$\overline{A} \times Y = \begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \end{bmatrix} \end{array}$$

and hence $R = \max\left\{A \times B,\ \overline{A} \times Y\right\}$

$$R = \begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.2 & 0.4 & 0.8 & 0.6 & 0.2 & 0.2 \\ 0 & 0.4 & 1 & 0.6 & 0.2 & 0 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \end{bmatrix} \end{array} \tag{A.32}$$

Approximate Reasoning The primary goal of fuzzy systems is to formulate a theoretical foundation for reasoning about imprecise propositions, which is termed *approximate reasoning* in fuzzy logic technological systems.

Let us have a rule-based format to represent fuzzy information. These rules are expressed in conventional antecedent-consequent form, such as

Rule 1: IF x is A THEN y is B where A and B represent fuzzy propositions (sets)

Now let us introduce a new antecedent, say A', and we consider the following rule:

Rule 2: IF x is A', THEN y is B'

From information derived from Rule 1, is it possible to derive the con-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

sequent Rule 2, B'? The answer is yes, and the proceedure is a fuzzy composition. The consequent B' can be found from the composition operation

$$B' = A' \circ R \tag{A.33}$$

Example A.10 Reconsider the "fuzzy system" of Example A.9. Let a new fuzzy set A' be given by $A' = \frac{0.5}{1} + \frac{1}{2} + \frac{0.2}{3}$. It is desired to find an approximate reason (consequent) for the rule IF A' THEN B'.

SOLUTION: The relations (A.32) and (A.33) are used to determine B'.

$$B' = A' \circ R = \begin{bmatrix} 0.5 & 0.5 & 0.8 & 0.6 & 0.5 & 0.5 \end{bmatrix}$$

or

$$B' = \frac{0.5}{1} + \frac{0.5}{2} + \frac{0.8}{3} + \frac{0.6}{4} + \frac{0.5}{5} + \frac{0.5}{6}$$

Note the inverse relation between fuzzy antecedents and fuzzy consequences arising from the composition operation. More exactly, if we have a fuzzy relation $R: A \rightarrow B$, then will the value of the composition $A \circ R = B$? The answer is no, and one should not expect an inverse to exist for fuzzy composition. This is not, however, the case in crisp logic, i.e., $B' = A' \circ R = A \circ R = B$, where all these latter sets and relations are crisp. The following example illustrates the nonexistence of the inverse.

Example A.11 Let us reconsider the fuzzy system of Examples A.9 and A.10. Let $A' = A$ and evaluate B'.

SOLUTION: We have

$$B' = A' \circ R = A \circ R = \frac{0.3}{1} + \frac{0.4}{2} + \frac{0.8}{3} + \frac{0.6}{4} + \frac{0.3}{5} + \frac{0.3}{6} \neq B$$

which yields a new consequent, since the inverse is not guaranteed. The reason for this situation is the fact that fuzzy inference is imprecise, but approximate. The inference, in this situation, represents approximate linguistic characteristics of the relation between two universes of discourse.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Problems

A.1. Given two fuzzy sets A and B on X as follows

$\mu(x_i)$	x_1	x_2	x_3	x_4	x_5	x_6
A	0.1	0.5	0.7	0.8	0.6	0.1
B	0.9	0.8	0.6	0.3	0.1	0.0

Calculate

a. $\overline{A}$

b. $A \cup B$

c. $A \cap B$

d. $A \cap \overline{A}$

e. $\overline{A \cap B}$

f. $\overline{A} \cup \overline{B}$

g. $A \cup \overline{A}$

Find

a. $(\overline{A})_{0.6}$

b. $(A \cup \overline{A})_{0.6}$

c. $(A \cap B)_{0.6}$

d. $(A \cap \overline{A})_{0.5}$

e. $(A \cup B)_{0.6}$

f. $(\overline{A} \cup \overline{B})_{0.5}$

g. $(\overline{A \cap B})_{0.5}$

A.2. Prove the DeMorgan's laws ($\overline{A \cap B} = \overline{A} \cup \overline{B}$ and $\overline{A \cup B} = \overline{A} \cap \overline{B}$) hold for A and B on X.

A.3. Given the fuzzy numbers

$$J = \underset{\sim}{3} = \frac{0.1}{2} + \frac{1}{3} + \frac{0.1}{4}$$

$$K = \underset{\sim}{2} = \frac{0.1}{1} + \frac{1}{2} + \frac{0.2}{3}$$

Calculate $L = J \times K = \underset{\sim}{6}$

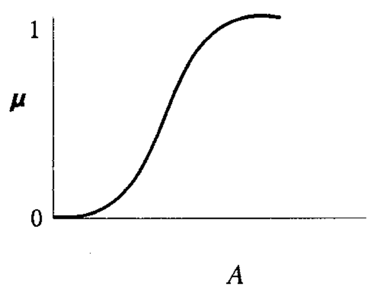
A.4. For the following sets, $x = \{1, 2, 3, 4, 5\}$ and $y = \{1, 2, 3, \ldots, 9, 10\}$ and given the function $y = f(x) = 2x - 1$, construct a relation, R, on $X \times Y$ to express this function.

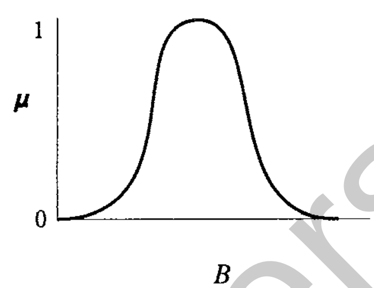
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

A.5. For the two fuzzy sets shown below, A and B, sketch the membership functions of

(i) $\mu_{A \cup B}$ (ii) $\mu_{A \cap B}$ (iii) $\mu_{\overline{A} \cup B}$



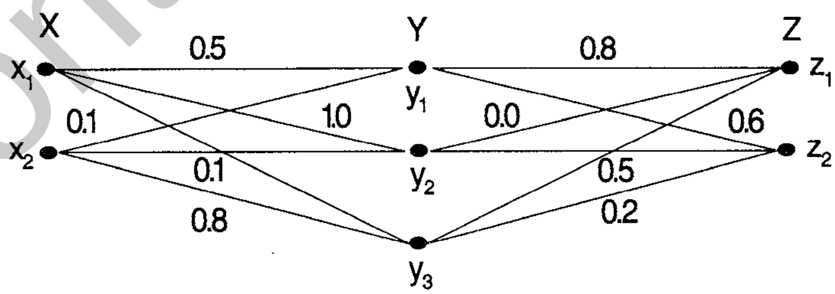


A.6. Prove that the excluded middle laws do *not* hold for fuzzy sets; i.e., prove that

(i) $A \cup \overline{A} \neq U$ (ii) $A \cap \overline{A} \neq \varnothing$

A.7. For a fuzzy set with membership function $\mu_A(x) = \dfrac{1}{1+x}$ for $x = \{0, 1, 2, 3, 4, 5\}$, and $f(x) = x(3 - x)$, find $\mu f(x)$ using the extension principle.

A.8. For the following mapping with "fuzzy strengths," find R, a fuzzy relation on $X \times Y$, find S, a fuzzy relation on $Y \times Z$, and find T, a fuzzy relation $X \times Z$.



controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

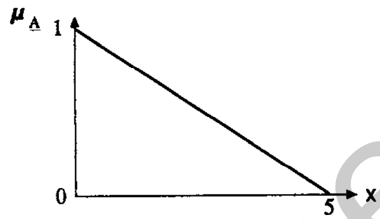
A.9. If P: x is A on X
Q: y is B on Y
S: y is C on Y

Prove that

$$T\left[(P \to Q) \vee (\overline{P} \to S)\right] = 1$$
$$\forall\ x, y \in x \times y$$

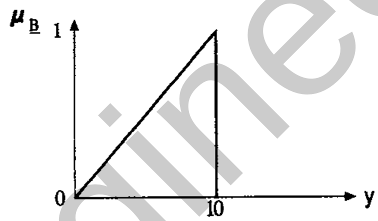
A.10 Given

$$A = \int_{x \in [0,5]} \frac{1 - 0.2x}{x} \quad \to$$



and

$$B = \int_{y \in [0,10]} \frac{0.1y}{y} \quad \to$$



(a) Construct a fuzzy relation R for the conditional statement:
If x is A, then y is B

(b) Given x is $A_1 = \left\{\frac{1}{2}\right\}$ determine y is B_1 = ?

A.11. For the fuzzy numbers

"Approximately 5": $\frac{0.1}{3} + \frac{0.5}{4} + \frac{1}{5} + \frac{0.5}{6} + \frac{0.1}{7}$

"Approximately 1": $\frac{0.2}{0} + \frac{1}{1} + \frac{0.5}{2} + \frac{0}{3}$

Find 5 + 1, 5 − 1, and 5 × 2.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

A.12. Given two fuzzy sets:

$$A = zero = \frac{0}{-2} + \frac{0.3}{-1} + \frac{1.0}{0} + \frac{0.3}{+1} + \frac{0}{+2}$$

defined on the universe $X = [-5, 5]$, and

$$B = positive\ medium = \frac{0}{0} + \frac{0.4}{1} + \frac{1.0}{2} + \frac{0.4}{3} + \frac{0}{4}$$

defined on the universe $Y = [-5, 5]$,

(a) Construct the relation for the rule: IF A THEN B, i.e., IF x is *zero* THEN y is *positive medium* based on $\mu_R(x, y) = \min[\mu_A(x), \mu_B(y)]$ and $\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y)$.

(b) If $A' = positive\ small = \frac{0}{1} + \frac{0.5}{0} + \frac{1.0}{+1} + \frac{0.5}{2} + \frac{0}{+3}$, find B'.

(c) Using the fuzzy relations R found in part (a), check if $B = A \circ R$.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Appendix B

The Fuzzy Logic Development Kit

B.1 Introduction

Developed by Dreier (1994a), FULDEK™* is an acronym for the FUzzy Logic DEvelopment Kit, a comprehensive fuzzy logic development environment that runs on IBM PCs and compatibles using the Windows™** operating system. The FULDEK program is fast, easy to learn, and user-friendly. It has a graphical editor to build membership functions and rules, and a simulation environment to test the control rules in a closed-loop simulation. The FULDEK program can generate fuzzy inference rules from observed data automatically, and it can generate fuzzy logic computer code automatically. The intuitive ordering of menus and functions in the FULDEK program helps experienced fuzzy logic users as well as newcomers. This appendix, taken in part from Dreier (1994b), describes key features of the FULDEK program and presents a simulation and program examples of its operation.

B.2 Description of the FULDEK Program

The current version of the FULDEK program comes in three levels. The *starter kit* provides enough rules and membership functions to explore fuzzy logic and produce useful fuzzy rule bases (FRB). The *student kit* more than doubles the capacity of the starter kit and adds a FORTRAN code generator and the AutoRule function. The FORTRAN code generator

* FULDEK is a trademark of Bell Helicopter Textron, Inc.
** Windows is a registered trademark of Microsoft Corporation.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

reads the FRB and writes fuzzy logic code in FORTRAN to a file. That file is compilable and ready to use as is, or it may be inserted into a user's simulation code or embedded controller. The AutoRule tool reads observed data from a file and generates fuzzy inference rules and a complete FRB. This option is particularly helpful in parameter or plant identification. The *professional kit* more than doubles the capacity of the student kit and adds ADA, BASIC, MATLAB and C code generators and phase-plane analysis. Other important features of the program include

1) operation in the Windows environment, with or without a mouse;
2) six types of membership functions and the flexibility to allow the user to define functions;
3) rule interaction using text and three graphic modes;
4) shortcuts to reduce editing time and testing;
5) the capability to link the user's fuzzy inference rules to a linear model of the plant to be controlled and then preform a time domain simulation (the simulation, including phase plane analysis, helps the user assess the performance of the fuzzy logic controllers).

The FULDEK program has two major sections, the *EDITOR* option and the *RUN* option. These options divide the development task into two parts. The first part, the *EDITOR*, identifies the fuzzy variables, membership functions, and rules. The second part, the *RUN* section, analyzes the interaction of the rules and the behavior of a dynamic system linked to the rules.

B.3 *EDITOR* Option

The user manipulates files, edits fuzzy variables, membership functions, rules, and ASCII files, selects defuzzification and composition methods, sets scaling factors for the fuzzy variables, and calls the AutoRule tool in the *EDITOR* option.

The *EDITOR* option is loaded at execution start and presents a menu bar. Clicking with the mouse or depressing a shortcut key activates all of the menu items. The major menu items are described below.

The File Menu The *Files* menu item provides access to all ASCII files that FULDEK uses, using a file-finder familiar to Windows users. The basic FULDEK file is the Fuzzy Rule Base or FRB. The FRB contains all the information that describes the fuzzy variables, the membership functions, and the rules. The FULDEK program also has a built-in full-screen editor

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

that is useful when the user wishes to modify an ASCII file. Printer setup and access are also available.

The Edit Menu The *Edit* menu item comprises operations that describe fuzzy variables, membership functions, and fuzzy rules. A new feature enables the user to extract knowledge from observed data. The user may also select the defuzzification and inference methods.

The *Names* operation declares fuzzy variables to be INPUT, OUTPUT, or BOTH. Each fuzzy variable has left and right physical values that form the boundaries of its universe of discourse and map a physical value into a scaled value between –1 and +1. The INPUT and OUTPUT classifications are self-explanatory. The BOTH classification permits a fuzzy variable to appear as a feedback variable or a chaining variable. A typical variable definition is shown in Figure B.1.

The *Membership* operation describes and builds fuzzy membership functions. The FULDEK program provides six standard types (singleton, rectangular, triangular, trapezoidal, Gaussian, and sigmoidal). The *AutoGen* feature quickly builds and distributes these functions, reducing the user's editing effort. Alternatively, the user may manually build a function with up to nine *x-y* pairs. An instruction bar at the bottom of the screen guides the user through the process. In Figure B.2, the membership function TTH_POS 1 is shown. The user sees the numerical values that define the function, and a picture of the function with a simple click of the mouse.

The *Defaults* operation lets the user select the composition method and the defuzzification method. Composition refers to the way that compound antecedents are combined. The standard conjunction method (AND operation) uses the MIN function, and the standard disjunction method (OR operation) uses the MAX function. The FULDEK program also offers multiplication as a substitute for MIN, and limited addition as a substitute for MAX. Defuzzfication refers to the manner in which conclusions from competing rules are combined to arrive at a consensus answer. Kosko (1992) describes three methods. They are Correlation Minimum Inference, Correlation Product Inference, and Max-Product (Winner-Take-All) Inference. The FULDEK program offers all three.

The *Rules* operation works intuitively, enabling the user to write rules in a natural manner. The FULDEK program uses a point-and-click rule editor and has an instruction bar at the bottom of the page that gives the user instant hints on how to write a rule. The user constructs an antecedent

controlengineers.ir

controlengineers.ir
@controlengineers

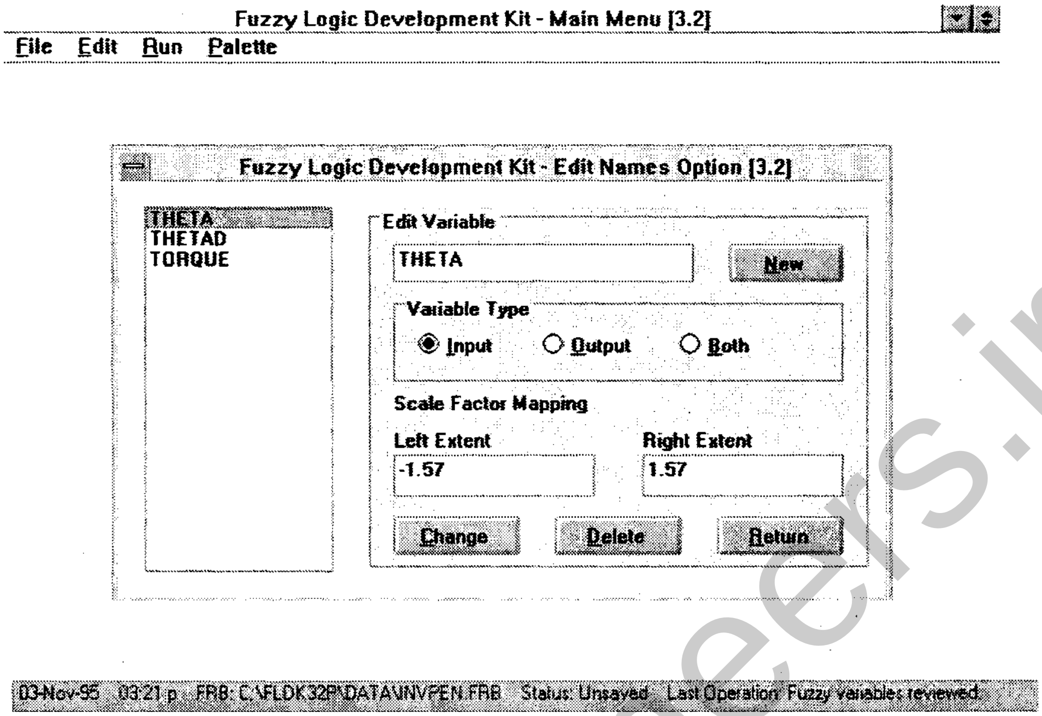


Figure B.1. The FULDEK program's Edit option window defining variable THETA.

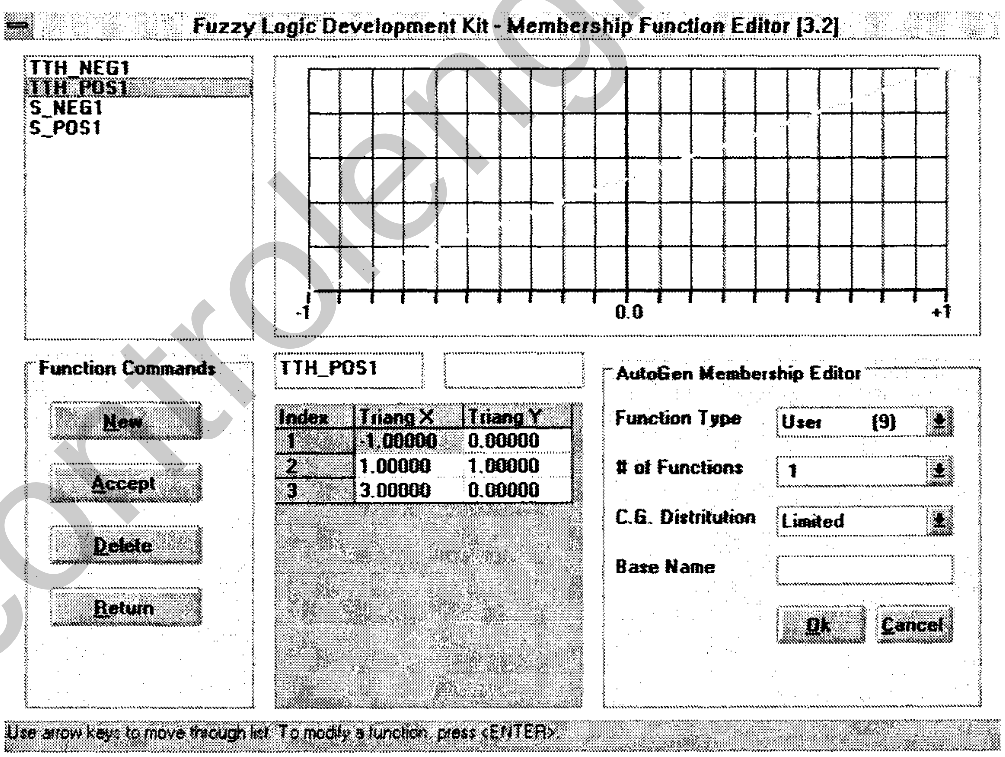


Figure B.2. The FULDEK program's Edit option window defining a triangular membership function for TTH_POS 1.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

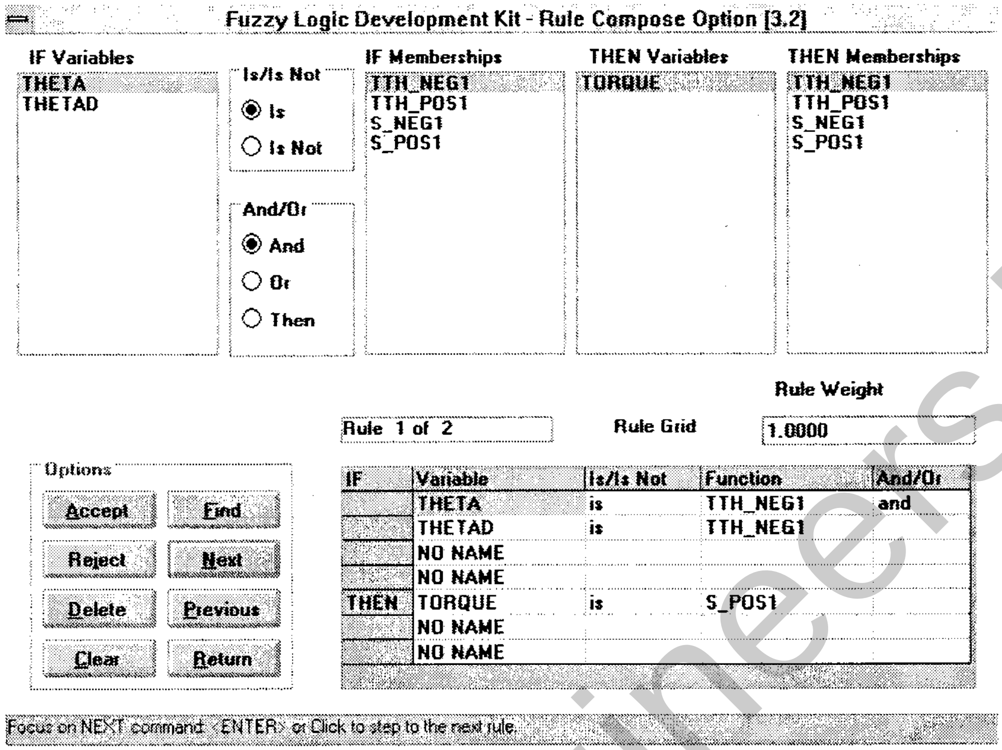


Figure B.3. The FULDEK program's rule composition window.

phrase by selecting a fuzzy variable, the IS or IS NOT function, and a membership function. If more than one antecedent clause is required, select AND, or OR, and build the second antecedent clause in the same manner. Up to four antecedent clauses may be used. The conclusions are built in a similar manner. Up to three are allowed in a rule. In many cases, this reduces the number of rules a user must write. Just before accepting a rule into the rule base, the rule building page looks something like Figure B.3.

The *EDITOR* has several smart features that deserve mention. If the user wishes to remove a fuzzy variable or a membership from the FRB, the FULDEK program checks the rules for that variable or function. If it is found, the FULDEK program warns the user and then offers (1) to stop the deletion process, (2) to continue the deletion by discarding the entire rule that has the variable or function, or (3) to continue with the deletion by proposing a new rule which does not contain the variable. See Figure B.4 for an example of this feature. Another time saver is the grammar checker in the rule editor. The program flags incorrectly written rules, and, when the correction is unambiguous, corrects them. Another smart feature of the FULDEK program generates fuzzy inference rules from observed data. *AutoRule* (Dreier and Cunningham, 1996) reads a time history or other

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

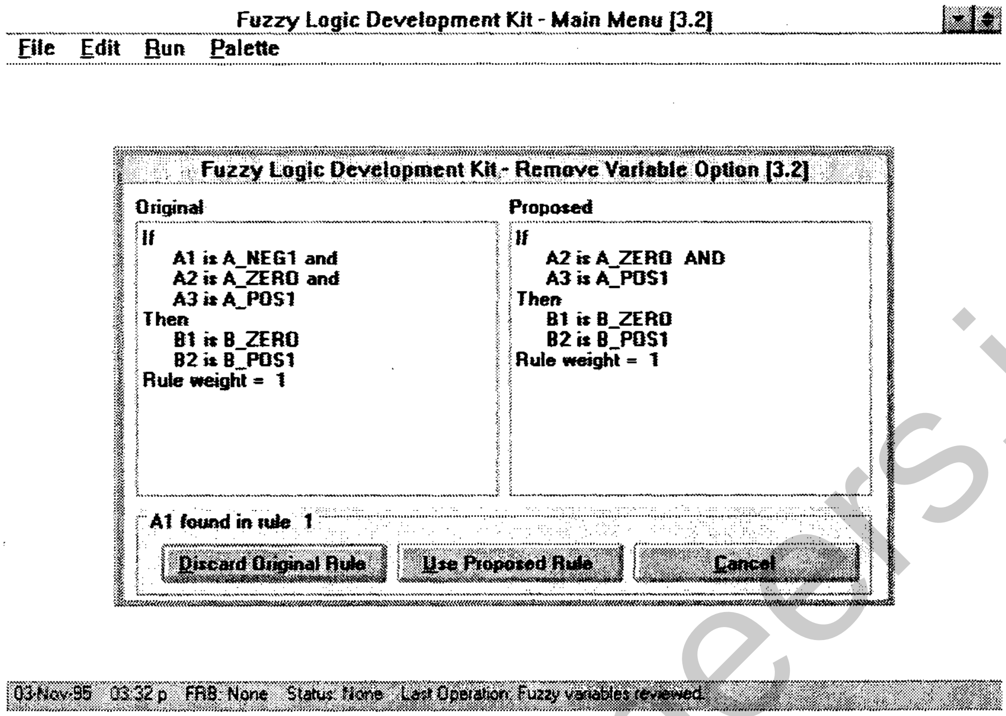


Figure B.4. The FULDEK program's rule deletion feature.

collection of input/output data and generates a power set of fuzzy rules that "fit" the data. Pruning to a minimum set of rules is available after the fit. This is a useful feature for parameter identification and for situations when only input/output experimental data is available, i.e., access for a mathematical model may not be possible. Figure B.5 shows the AutoRule screen after a highly nonlinear function has been reduced to a set of fuzzy inference rules.

B.4 The *RUN* Option

The user exercises the rules in the *RUN* option. Three different formats are available. The first format is a step-by-step examination as one input variable sweeps through its universe of discourse while the other input variables, except feedback variables, hold at user-defined values. The output of this slice through the rule hyperplane is a list of the fuzzy variables and their values.

The second *RUN* format, called the *Map* option, is a continuous sweep of one or two input variables through their universes of discourse. The user selects one or two fuzzy variables to sweep, and one fuzzy variable to examine the output from the rule base. All other variables, except those typed as BOTH, are held at user-defined initial values. The user may also select the detail in the view, either coarse or fine. The output is a graphical

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

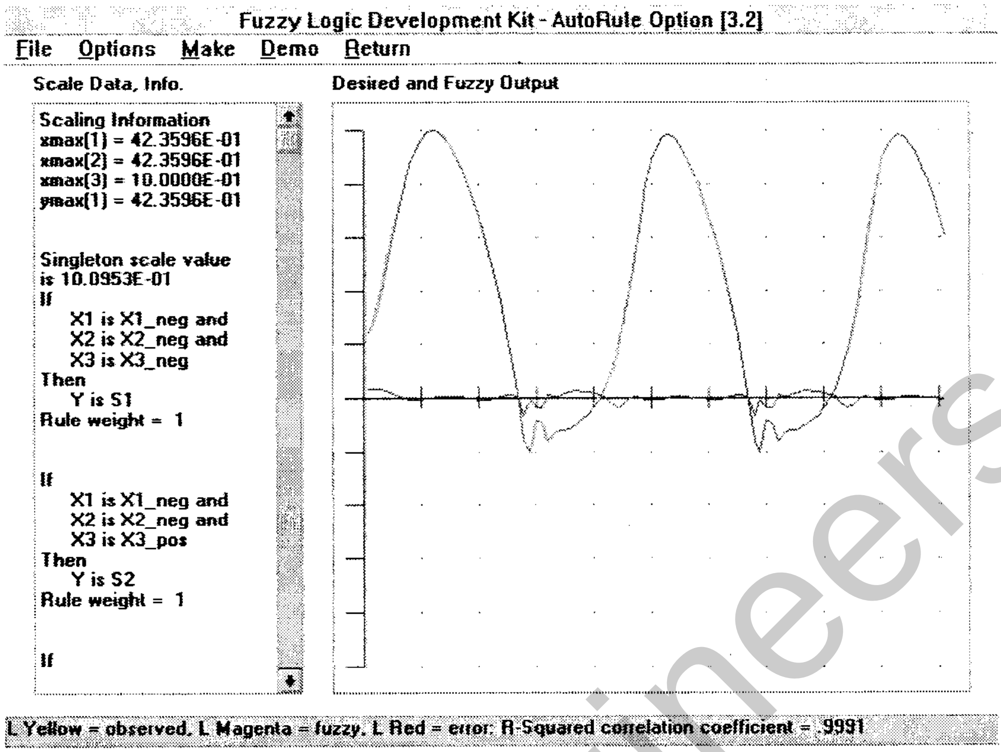


Figure B.5. The FULDEK program's AutoRule generating feature.

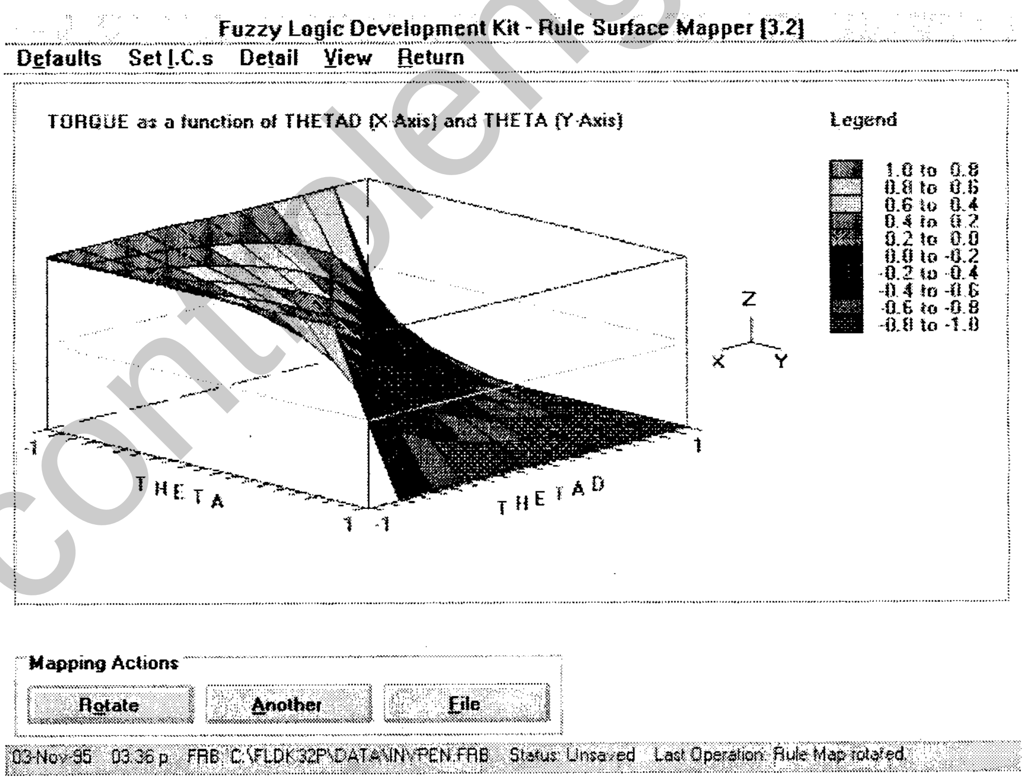


Figure B.6. The FULDEK program's rule (control) surface.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

display of a line or surface in the rule hyperplane, similar to the 3-D surface in Figure B.6. In some cases, the surface is presented edge on. In order to see the surface from different perspectives, a ROTATE button turns the surface 90 degrees counterclockwise about the vertical axis with each click. The File button in Figure B.6 allows the user to create an ASCII file that contains a defuzzified fuzzy look-up table of the inferencing or control. The *MAP* option provides a way to detect unintentional discontinuities in the rules, gaps in the influence of a variable on the outcome, and a succinct summary of the control logic over the range of each variable. The *RUN* option tools above provide excellent examination of the fuzzy rules in a stand-alone sense, but they are not actually controlling anything.

The third format links a state-space linear model to the fuzzy rules to form a "closed loop" simulation. This format is most useful for those interested in evaluating the performance of the rules in a dynamic environment. The output is presented in conventional X versus Time or in phase-plane. The state-space linear model is represented in the form

$$\dot{x} = A*x + B*u + B_e*u_e$$
$$y = C*x + D*u + D_e*u_e$$

where x is the state vector, y is the output or observability vector, u is the control vector, and u_e, is an external input vector which introduces disturbances and/or noise. The A, B, C, and D matrices are the usual dynamics and control observability and feedforward matrices, and B_e and D_e are the disturbance influence and noise influence matrices. The FULDEK program can express up to 20 first-order differential equations in this way.

In the closed loop simulation, the time varying inputs to the fuzzy rules' controller are the outputs of the dynamic plant, and the inputs to the dynamic plant are the outputs of the fuzzy rule base. The details of the link are beyond the scope of this appendix. Briefly though, the user merely matches a fuzzy variable name with an element in the any of the vectors described above. An interactive screen lets the user create these links easily.

The state-space model is stored in an ASCII file. The external inputs are defined with a second ASCII file. The user selects the integration step size and end time, and the variables to be plotted as a function of time. The user may also select regular time history simulation in X versus Time format, or a phase-plane analysis which plots two states against each other as a function of time.

Figure B.7 shows the operation of a closed loop simulation. In this simulation, the controller drives the ubiquitous inverted pendulum (the THETA trace on the upper-left grid) from an initial displacement of one radian up to vertical without overshoot, and it does that using just *two* rules. The

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

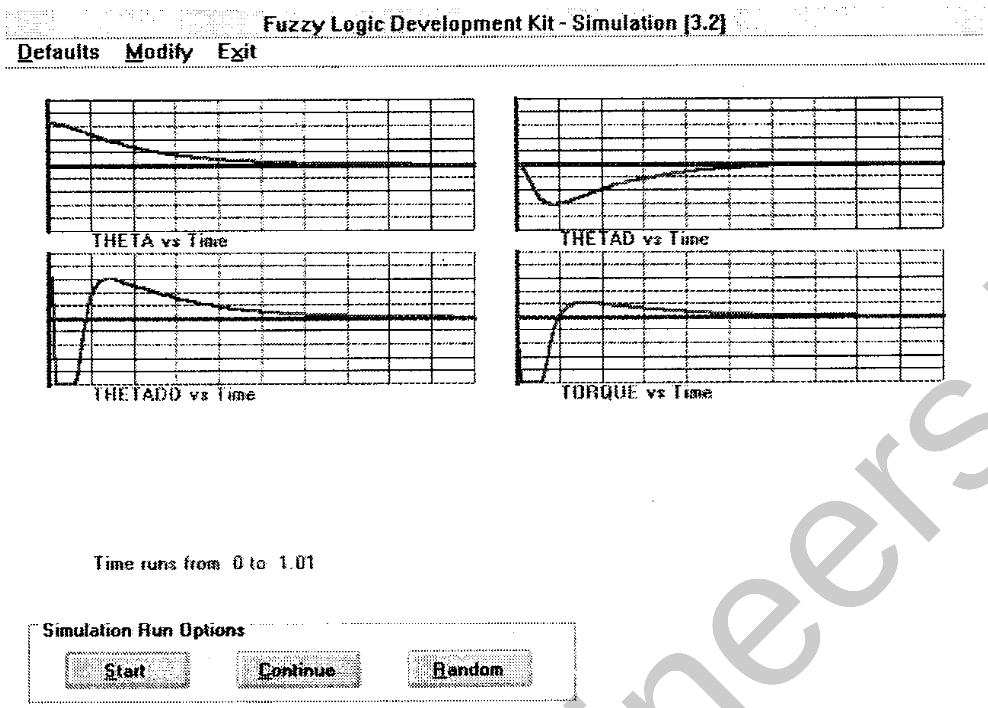


Figure B.7. The FULDEK program's simulation results of an inverted pendulum using two rules.

design of the two-rule controller extends beyond the scope of this appendix. However, the rules themselves are straightforward. They are

Rule 1

If THETA is POS and THETAD is POS
Then TORQUE is NEG_Q

Rule 2

IF THETA is NEG and THETAD is NEG
Then TORQUE is POS_Q

While it may seem that some rules are missing, Lin and Cunningham (1994) have shown that with proper definitions of linguistic terms POS and NEG, these two rules are sufficient.

B.5 Post-Processing Feature of FULDEK

The Fuzzy Rule Base is not much good if it can only demonstrate the usefulness of fuzzy logic controllers in linear simulations within the confines of the FULDEK program. Therefore, the professional version of the

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

FULDEK program offers post-processors that read fuzzy rule bases and write fuzzy logic controller code in ADA, BASIC, C, FORTRAN, or MATLAB. The source code is immediately available to compile and use or may be edited to fit within a user's embedded controller scheme. This is a very useful, time-saving option when building an embedded controller. The following section illustrates a real-time fuzzy control laser tracking system using a BASIC code generator.

B.6 A Real-Time Laser Beam Fuzzy Controller

In this section, the FULDEK program's BASIC code generator is used to design a real-time fuzzy controller for a laser beam tracking system. The experiment consists of a laser on a rotating pedestal so that it will center itself between two solar cells. If the output voltages from the two solar cells are low (i.e., the beam from the laser is not shining on either one of them), the motor should rotate the laser at a fast rate clockwise. When the output voltage from the solar cell on the right is higher than the output from the left solar cell, the motor should rotate the laser beam slowly clockwise. Similarly if the output from the left solar cell is greater than the output from the solar cell on the right, the motor should rotate the beam slowly counterclockwise. If the outputs from the two solar cells are approximately equal, and medium voltage, the motor should stop.

The voltages from the two solar cells are digitized using Computer Boards CIO-DAS08-PGA Analog to Digital converter PC expansion board. The board also has digital outputs available that we used to control a UDN2998W dual full-bridge motor driver.

The FULDEK program was used to specify fuzzy membership functions and rules. (See Figure B.8 and Table B.1.) It takes these memberships and rules and creates a software program to implement them. Here the option to create Quick Basic code was used.

The A/D conversion board has eight analog input channels, and four digital outputs. It also comes with basic software to control it. We used two analog input channels (Channels 1 and 2) to make the outputs from the two solar cells available to the fuzzy logic software. The output "Z" from the fuzzy logic software had four possible values (-0.1 for counterclockwise slow, 0 for stop, +0.1 for clockwise slow, and +0.3 for clockwise fast). The output Z from the fuzzy logic program was converted to a number that selects the appropriate digital output for the motor controller. The digital output is a 4 bit word (0 –15); we used the two least significant bits to control the motor driver. The driver has an enable and phase connection to control the direction of the motor. With both low (i.e., digital output 0)

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table B.1 Fuzzy Logic Rules for Laser Beam

1 of 6	If X is XS and Y is XS	Then Z is ZL
2 of 6	If X is XL and Y is XS	Then Z is ZS
3 of 6	IF X is XM and Y is XM	Then Z is ZZ
4 of 6	IF X is XS and Y is XL	Then Z is ZNS
5 of 6	IF X is XMD and Y is XS	Then Z is ZS
6 of 6	If X is XS and Y is XM	Then Z is ZNS

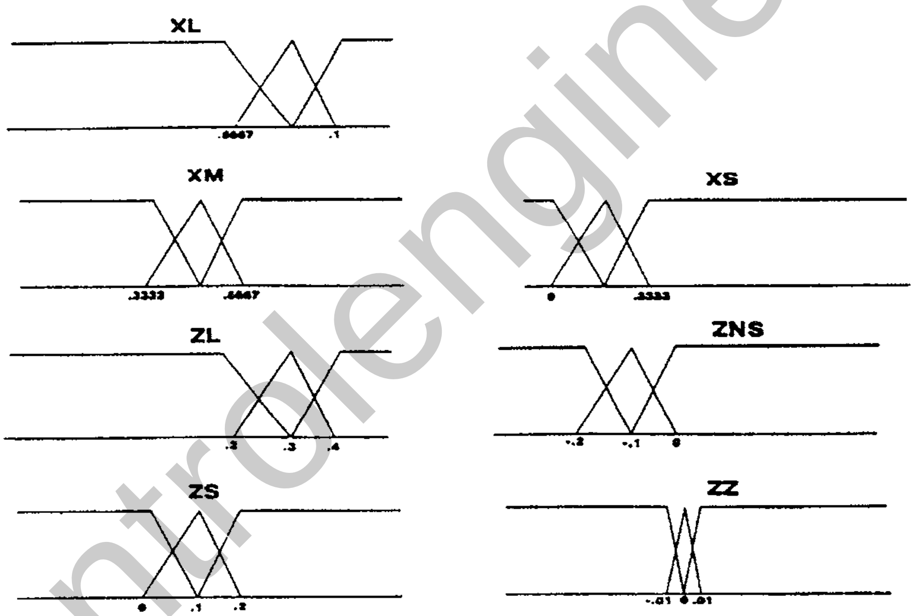


Figure B.8 Membership functions.

the motor turns the pedestal clockwise. With enable low and phase high (i.e., digital output 1) the motor turns the pedestal counterclockwise. With enable high the motor stops (i.e., digital output 2). The speed of the motor was controlled by varying the width of time the digital output (pulse) was held on. While the computer reads in new values of X and Y and calculates an output, the motor is turned off.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

The components used to implement this project are

Meredith Instruments Helium Neon Laser
A TDK RM-S Power Supply for the laser
A Convex lens to widen the beam from the laser
Two Archer Silicon Solar Cells
Apricot UDN2998W Dual Full-Bridge Motor Driver
A pottery wheel used as the rotating pedestal
Global Specialties Proto Board to implement the motor driver
Computer Boards CIO DAS08-PGA Analog to Digital Conversion Board
Packard Bell AXCEL 386/SX PC to run the software and operate the AID board

The two solar cells were mounted in a 6" × 3" × 13" cardboard box to limit the background light. The pottery wheel pedestal and laser were located 30" away from the solar cells (See Figure B.9 for a schematic drawing of implementation and Figure B.10 for a detailed schematic of the integration of the system.)

The implementation worked as expected. The motor control was pulsed because the software was not fast enough to allow the motor to run full speed while new input values were read in and new output values calculated. (See Figure B.11 and Table B.2 for some experimental results.) Note the results are as expected; when both X and Y are low ($< .3333$), then Z is .3 which is converted into fast clockwise (digital output 0 with long duration pulse) motion of the pedestal. When X is high ($> .667$) and Y is low or

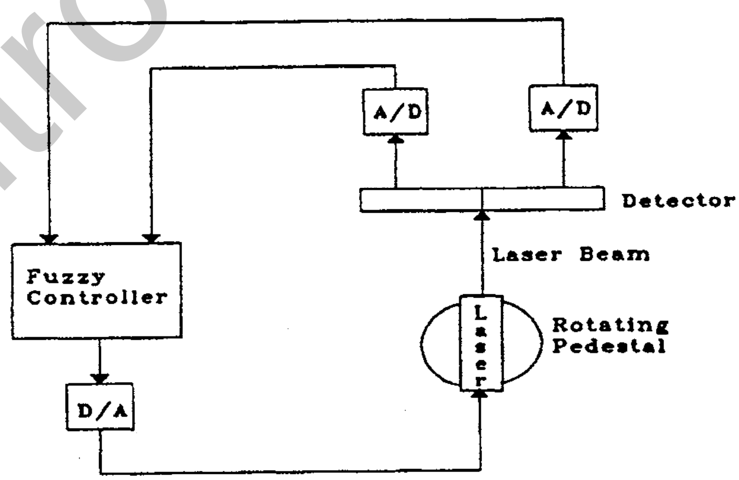


Figure B.9. Schematic Drawing of Implementation of Laser Beam System.

controlengineers.ir

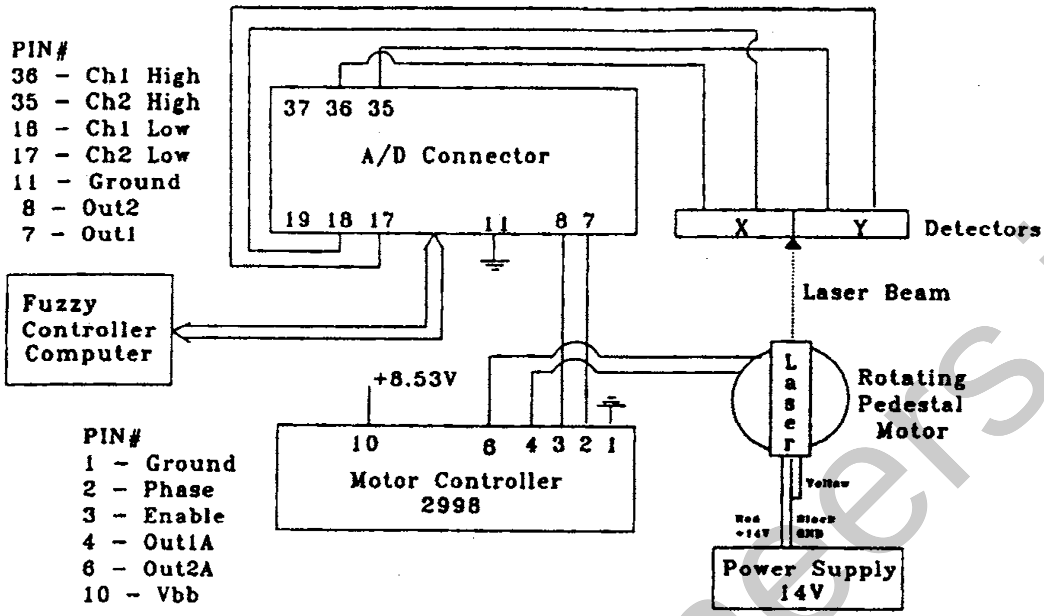


Figure B.10. Detailed schematic of the integration.

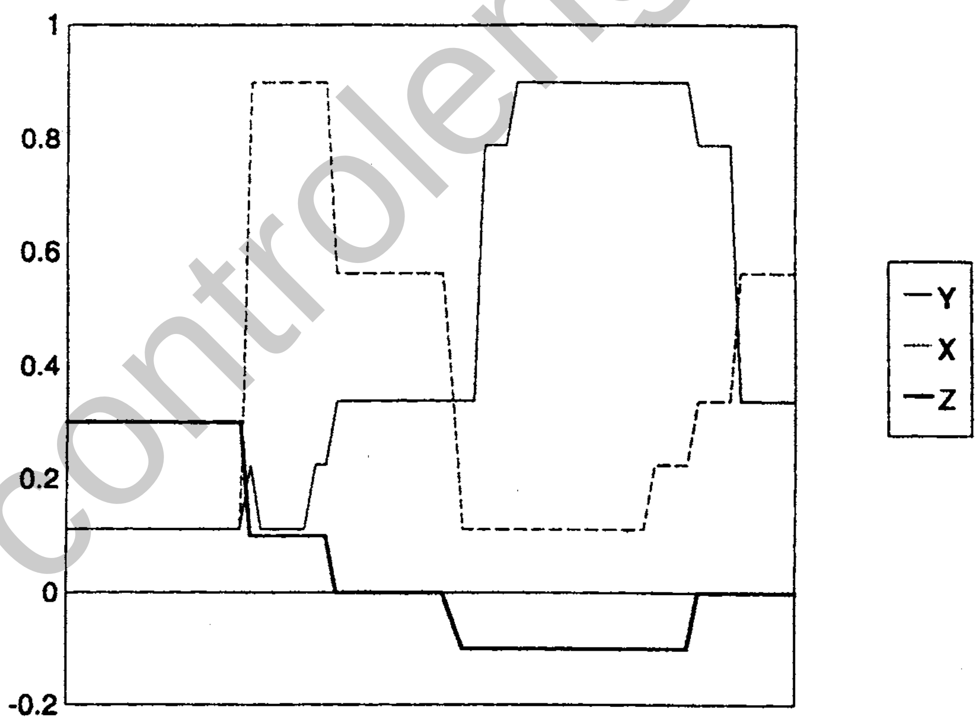


Figure B.11 Experimental results.

controlengineers.ir
@controlengineers
اختصاصی مهندسی کنترل

controlengineers.ir
@controlengineers

Table B.2 *X, Y,* and *Z* Data

X	*Y*	*Z*
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.1125008	0.1125008	0.3
0.9000064	0.2250016	0.09999999
0.9000064	0.1125008	0.09999999
0.9000064	0.1125008	0.09999999
0.9000064	0.1125008	0.09999999
0.9000064	0.1125008	0.09999999
0.9000064	0.1125008	0.09999999
0.9000064	0.2250016	0.09999999
0.9000064	0.2250016	0.09999999
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.1125008	0.3375024	-0.0999999

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Table B.2 (cont'd) *X, Y,* and *Z* Data

X	Y	Z
0.1125008	0.3375024	-0.0999999
0.1125008	0.7875056	-0.0999999
0.1125008	0.7875056	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.1125008	0.9000064	-0.0999999
0.2250016	0.9000064	-0.0999999
0.2250016	0.9000064	-0.0999999
0.2250016	0.9000064	-0.0999999
0.2250016	0.9000064	-0.0999999
0.3375024	0.7875056	0
0.3375024	0.7875056	0
0.3375024	0.7875056	0
0.3375024	0.7875056	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0
0.562504	0.3375024	0

controlengineers.ir

medium (< .667), Z is 0.1 which is converted to slow clockwise motion (digital output 0 with short duration pulse). When Y is high and X is low or medium, then Z is -0.1 which is converted to slow counterclockwise motion (digital output 1 with short duration pulse). When both X and Y are medium ($.3333 < XY < .667$), Z is 0 which is converted to motor stopped (digital output 2).

controlengineers.ir
@controlengineers

Below is the user's written code for interfaces, A/D converter, etc., followed by the BASIC code generated by the FULDEK program.

```
DIM ANS(2)                        'Array for input values
DIM D%(13)                        'Array for parameters used by the
CALL subroutine
COMMON SHARED D%()                'Allow subroutine access to this
variable.
DECLARE SUB CIO8 (MD%. BYVAL DUMMY%, F%)   'Declare subrou-
tine and pass
                                  'required parameters.
F% = 0                            'Set error flag to 0.
'------------------------------------------------------------------------

CLS
LOCATE 10, 5
PRINT "NOTE:"
PRINT
PRINT "This program assumes the CIO-DAS08 is set up for 5V BIPOLAR at
the"
PRINT "200 HEX (512 decimal)."
LOCATE 22, 10
PRINT "Press Q to quit or any other key to continue:"

loopl:
 QUITS = INKEY$
IF QUITS = "" THEN GOTO loopl
IF QUITS = "Q" OR QUITS = "q" 1 THEN GOTO QUIT
CLS
'------------------------------------------------------------------------
'       Initialize CIO-DAS08 using Mode 0
'------------------------------------------------------------------------

MD%=0                             'Set MODE to 0 (initialize CIO-DAS08
D%(0) = &H200

D%(l) = 0                         'A/D range +/-5V
D%(2) = 40
                                  This variable is used by Mode 50 'and 51.

CALL CIO8(MD%, VARPTR(D%(0)), F%) '       Call the subroutine
IF F% <> THEN GOTO ERRMSG                 'Check for error messages

'------------------------------------------------------------------------

MD% = 1                 'Set MODE to 1 (Mode to set MUX up)
D%(0) = 1               'Lower scan limit (channel)
D%(l) = 2               'Upper scan limit (channel)
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers


```
CALL CIO8(MD%, VARPTR(D%(0)), F%) 'Call the subroutine
IF F% <> THEN GOTO ERRMSG 'Check for error messages

'-----------------------------------------------------------------------------

MD% = 2                  'MODE = 2 (Sets next channel)
D%(0) = 1                First channel scanned set to 1
CALL CIO8(MD%, VARPTR(D%(0)), F%)         'Call the subroutine
IF F% <> THEN GOTO ERRMSG                 'Check for error messages

'-----------------------------------------------------------------------------

REPEAT:
CLS
FOR I= 1 TO 2
MD%= 3                   'Set MODE to 3 (Check MUX set-up).
CALL C108(MD%, VARPTR(D%(0)), F%)         'Call the subroutine
IF F% <> THEN GOTO ERRMSG                 'Check for error messages

'-----------------------------------------------------------------------------
' Store variables returned by MODE 3.
'-----------------------------------------------------------------------------

CHAN% = D%(0)        'Store number of next channel to be sampled
LOW% = D%(l)                 'Store number of lowest channel to be
scanned
HI% = D%(2)          'Store number of highest channel to be scanned

'-----------------------------------------------------------------------------

' Do one A/D conversion and increment the MUX on the CIODAS08 one
' channel using Mode 4.
' Define the variable for MODE (MD%).

'-----------------------------------------------------------------------------

MD% = 4                  'MODE = 4 (One A/D conversion).
CALL CI08(MD%, VARPTR(D%(0)), F%)         'Call the subroutine
IF F% <> THEN GOTO ERRMSG        'Check for error messages

'-----------------------------------------------------------------------------

' The CIO-DAS08 has now read the channel defined by Mode 1 and Mode 2
and
' the MUX is set to read the next channel.
Print out the status of the MUX obtained by MODE 3.

'-----------------------------------------------------------------------------

LOCATE 7, 10: PRINT "Channel          Data"
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

```
LOCATE 9 + CHAN%, 12: PRINT CHAN%
X = ((.0070313 * D%(0)) + 0)
LOCATE 9 + CHAN%, 28: PRINT D%(0); "("; X; "Units ) "
LOCATE 18, 5: PRINT "MUX is set to read from CHAN "; LOW%;"; to
CHAN "; H1%
'
ANS(I) = X
NEXT I
'
X = ANS(1)
Y = ANS(2)
PRINT "X = "; X
PRINT "Y = "; Y
'
'
*********************************************************************

'THE FOLLOWING SECTION, WRITTEN BY FULDEK. TAKES THE
INPUTS X AND Y
'FROM THE SOLAR CELLS AND GENERATES THE OUTPUT FOR THE
MOTOR CONTROLLER
'*********************************************************************
'
'
'Fuzzify the inputs

IF X < .666667 THEN
xmu = 0!
ELSEIF X < .833333 THEN
xmu = (X - (.666667)) * 6!
ELSEIF X < 1! THEN
 xmu = (1! - X) * 6!
 ELSE
 xmu = 0!
 END IF
 muXismf1 = xmu
 '
 IF X < .333333 THEN
 xmu = 0!
ELSEIF X < .5 THEN
xmu = (X - (.333333)) * 5.9988
ELSEIF X < .666667 THEN
xmu = (.666667 - X) * 5.9988
ELSE
xmu = 0!
 END IF
 muXismf2 = xmu
```

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

```
IF X < 0! THEN
xmu = 0!
ELSEIF X < .166667 THEN
xmu= (X- (0!)) * 6!
ELSEIF X < .333333 THEN
xmu = (.333333 - X) * 6!
ELSE
xmu = 0!
  END IF
  rnuXisrnf3 = xmu
  '
  IF Y < .666667 THEN
  xmu = 0!
ELSEIF Y < .833333 THEN
xmu = (Y - (.666667)) * 6!
ELSEIF Y < 1! THEN
xmu=(l! -Y) *6!
ELSE
xmu = 0!
  END IF
  muYismfl = xmu

  IF Y < .333333 THEN
  xmu = 0!
ELSEIF Y < .5 THEN
xmu = (Y - (.333333)) * 5.9988
ELSEIF Y < .666667 THEN
xmu = (.666667 - Y) * 5.9988
ELSE
xmu = 0!
  END IF
  muYismf2 = xmu
  '
  '
  IF Y < 0! THEN
  xmu = 0!
ELSEIF Y < .166667 THEN
xmu = (Y - (0!)) * 6!
ELSEIF Y < .333333 THEN
xmu = (.333333 - Y) * 6!
ELSE
xmu = 0!
  END IF
  muYismf3 = xmu
  '
  '
  'Perforrn the inference
  '
  rule1 = muXismf3
  IF rule1 > muYismf3
```

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

```
THEN rule1 = muYismf3
END IF
rule2 = muXismfl
IF rule2 > muYisrnf3
THEN rule2 = muYismf3
END IF
rule3 = muXismf2
IF rule3 > muYismf2 THEN
rule3 = muYistaf2
END IF
rule4 = muXismf3
IF rule4 > muYismfl THEN
rule4 = muYismfl
END IF
rule5 = muXismf2
IF rule5 > muYismf3 THEN
rule5 = muYismf3
END IF
rule6 = muXismf3
IF rule6 > muYismf2
THEN rule6 = muYismf2
END IF

'Defuzzify the outputs


num = 0!
den=0!
nun = num + (.03) * rule1
den = den + (.1) * rule1
num = num + (.01) * rule2
den = den + (.1) * rule2
num = num + (O!) * rule3
den = den + (.01) * rule3
num = num + (-.01) * rule4
den = den + (.1) * rule4
num = num + (.01) * rule5
den = den + (.1) * rule5
num = num + (-.01) * rule6
den = den + (.1) * rule6
IF den <= 0! THEN den = 1!
Z = num / den * 1!
PRINT "values"; X Y, Z
'
'*************************************************************************
'THIS PART ACTIVATES A DIGITAL OUTPUT TO THE MOTOR
'CONTROLLER, BASED ON THE VALUE OF Z.
'*************************************************************************
```

controlengineers.ir
@controlengineers

```
'MD% = 0
'D%(0) = 512
'D%(l) = 0
'D%(2) = 40
'CALL CIO8(MD%, VARPTR(D%(0)), F%)
MD% = 14
D%(0) = 2
CALL CIO8(MD%, VARPTR(D%(0)), F%)
'INPUT "PUT IN A NUMBER BETWEEN -.1 AND .33"; Z
IF Z < -.01 THEN D%(0) = 1
IF Z > -.05 AND Z < .05 THEN D%(0) = 2
IF Z > .05 AND Z < .15 THEN D%(0) = 0
IF Z > .15 THEN D%(0) = 0: GOTO FAST
CALL CIO8(MD%, VARPTR(D%(0)), F%)
FOR J = 1 TO 100
NEXT J
D%(0) = 2
CALL CIO8 (MD%, VARPTR(D%(0 ) ) , F%)
FOR K = 1 TO 10
NEXT K
GOTO P

FAST:
FOR J = 1 TO 400
NEXT J
D%(0) = 0
CALL CI08(MD%, VARPTR(D%(0)), F%)
FOR K= 1 T0 2
NEXT K
GOTO P
QUIT:
D%(0) = 2
CALL CI08(MD%, VARPTR(D%(0)), F%)
END

ERRMSG:
CLS
PRINT "ERROR "; F%; ."CALLING MODE "; MD%; "OCCURRED"
P:
PRINT "HIT Q OR q TO QUIT"

errloop:
PAUSE$ = INKEY$
IF PAUSE$ = "" THEN GOTO REPEAT
IF PAUSE$ = "q" OR PAUSE$ = ."Q" THEN GOTO QUIT ELSE GOTO
REPEAT
```

controlengineers.ir

controlengineers.ir
@controlengineers

B.7 New Options in Version 4.0 of the FULDEK Program

The development of Version 4.1 of the FULDEK program is nearing completion. The major new feature of this version is the AutoTune™ automatic rule tuner using a Genetic Algorithm (GA). The basic method of a GA and its implementation in the FULDEK program are described below.

A genetic algorithm, as used in the FULDEK program, is an optimization scheme that uses the biological paradigm of "survival of the fittest" to improve not just one, but several estimates of an optimum solution to a problem. Each estimate of an optimum solution is expressed as a string of l's and 0's which code the variables that the user may manipulate. One can think of the strings as chromosomes, and each solution estimate as an individual. The chromosomes comprise several substrings, or genes, each of which codes an individual variable that the user can manipulate. The GA begins with a population of many strings, each representing a possible solution. Each solution (individual) is evaluated by measuring the performance of the operation to be optimized. In this application, the performance is a penalty function which the GA tries to minimize. The smaller the penalty function, the higher a performance score for a given individual. Once all the individuals in a population have been evaluated, a selection method is employed to determine which individuals may "mate" to produce superior offspring (hopefully). The offspring are generated by dividing the chromosomes of the parents at a randomly selected site and joining the head of one chromosome with the tail of the other. Occasionally, one of the bits may even mutate, changing a 1 to a 0, or vice versa. The offspring populate the next generation which is subjected to the same performance evaluation. This process continues ad nauseam, or until the observer stops the process based on a maximum generation count, or because no improvement in the best solution has been recorded, etc.

The FULDEK program offers great flexibility in the usual parameters that control the action of the GA. For instance, the user selects the number of individuals in a population, and the number of generations in an epoch. After all the generations in an epoch have been tested, the latest generation may be written to an ASCII file for later use, or it may be used immediately to seed another epoch. The probability of *crossover* (the GA term describing the head-to-tail swapping of genetic material) and mutation may be varied to encourage or retard innovation in the offspring. The Roulette Wheel and the Tournament methods are available for selection (the GA term describing how parents are chosen to mate) and both may be augmented with an Elitist strategy which guarantees that a certain number of the best individuals are passed to the next generation, no matter what selection method is employed. The number of bits that the GA uses to code

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

a gene can be either 8 or 16. This, along with the lower and upper values for a variable specify the precision for a given variable. For instance, if the problem at hand has three variables that may be manipulated, and the number of bits is set to 8, then the chromosome for an individual is 24 bits long, and the precision of any one variable is given by the formula

$$\pi = (UpperValue - LowerValue)/(2^{nbits} - 1)$$

The upper and lower values for variable extents (the numbers that define the domain of a fuzzy variable, i.e., its universe of discourse), and the upper and lower values for membership function breakpoints are specified by the user, or the user may employ the FULDEK program defaults.

The heart of the GA in the FULDEK program is the definition of the penalty function. This function penalizes states and controls, changes in states and controls, time to perform an action, and final values. The function is defined below:

$$J = z^T * S * z + \int_0^T \left(z^T * Q * z + \Delta z^T * Q_\Delta * \Delta z + u^T * R * u + \Delta u^T * R_\Delta * \Delta u + L \right) dt$$

where the vector z is the difference between observed and desired states, the vector u is the control input to the plant, the delta operator on a vector means the difference between the vector now and one time-step ago, Q, R, and S are diagonal, positive, semi-definite penalty matrices, and L is a scalar that is used to penalize the time required to achieve a certain state. By careful assignment of values in the Q, R, and S matrices, and the L value, a very elaborate penalty function is defined.

The FULDEK program recognizes the need to have some membership functions or extents always look like a reflection of others, or be the same as others. Therefore, the user may specify that a certain function or extent is the reflection or a copy of another function or extent. This reduces the chromosome size, improving the speed a small amount, but also ensures that a fuzzy rule base that started out with membership functions and/or extents that were symmetrically distributed about the origin remain so.

Comparing Figures B.12 and B.13 demonstrates the effectiveness of the GA to find a good solution. In Figure B.12, a simple mass under fuzzy logic control is tracking a target. The upper-left trace is the mass, the upper right is the target, and the lower-right trace is the force commanded by the fuzzy logic controller. While the mass does a fairly good job tracking the target, the control inputs are unacceptably wild—the "ride" is very bumpy.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

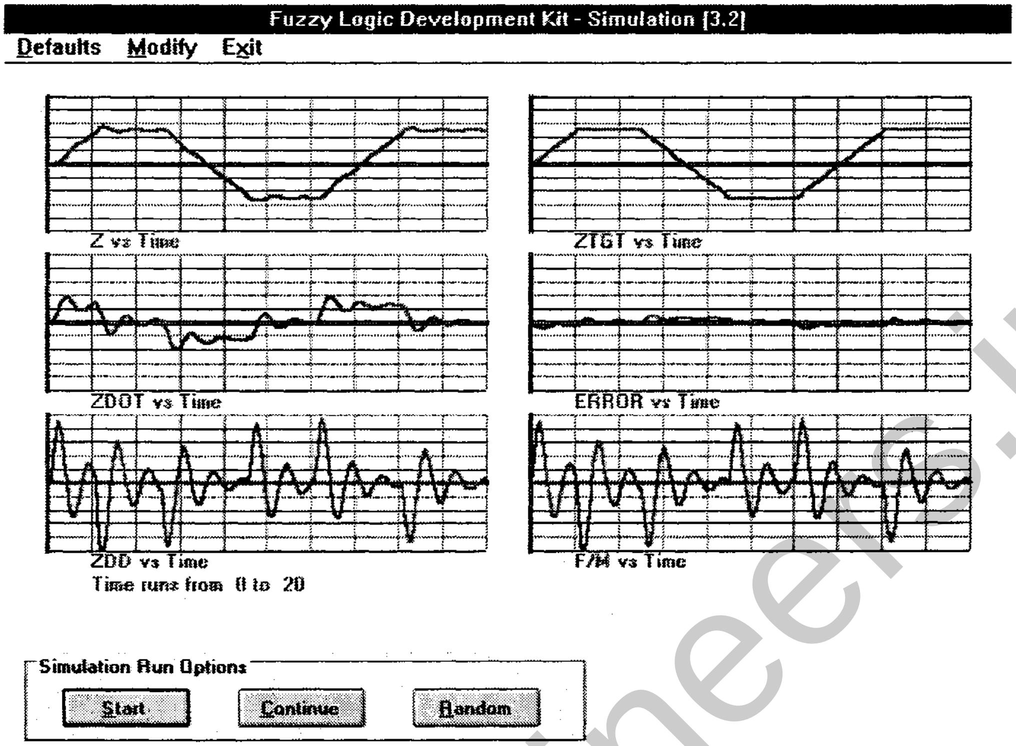


Figure B.12 A simple tracker before GA tuning.

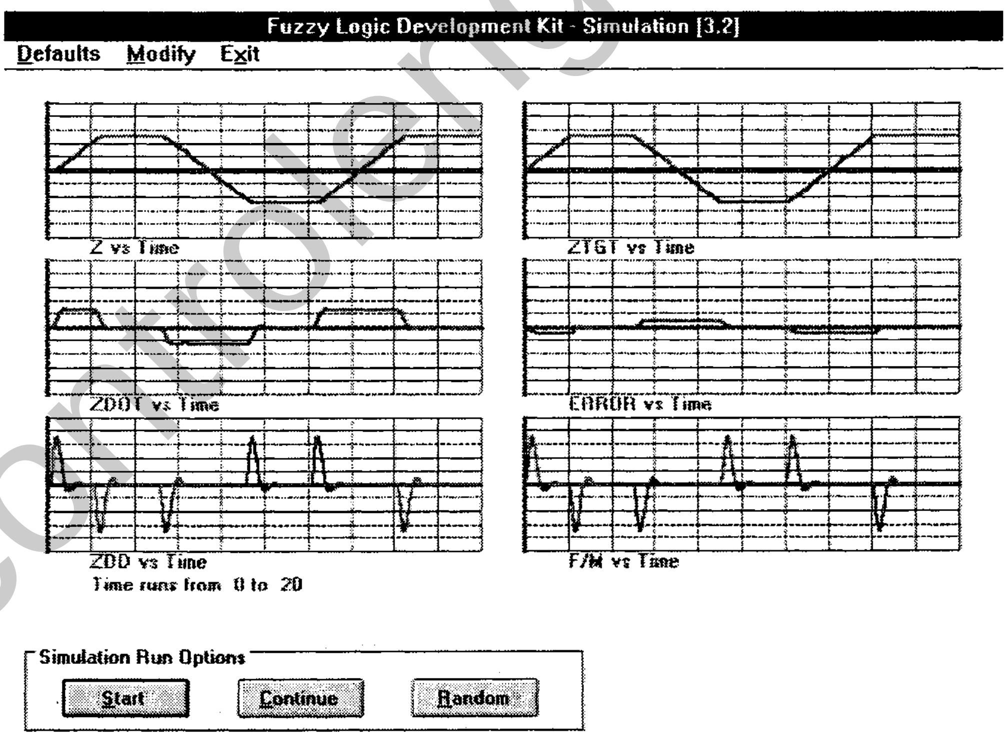


Figure B.13 The simple tracker after GA tuning.

controlengineers.ir

controlengineers.ir
@controlengineers

B.8 Conclusion

The FULDEK program is a user-friendly program that runs under the Windows environment. Its screens and operations are laid out in an intuitive manner, and it provides many tools to write and test fuzzy logic inference rules. The FULDEK program has matured and will continue to mature, with the primary goal to make fuzzy logic affordable for the beginner and yet comprehensive for the experienced user. Ways to use fuzzy logic are limited only by imagination. The FULDEK program is a good tool for getting there.

For more Information about the FULDEK program, including a price list, use the post cards at the end of the text to contact TSI Enterprises, Inc., P. O. Box 14126, Albuquerque, New Mexico 87191-4126, fax (505) 291-0013, phone (505) 298-5817.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers
controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

References

Akbarzadeh, M., Jamshidi, M., and Vadiee, N. 1994b. A hierarchical fuzzy controller using line-curvature feature extraction for a single link flexible arm. *Proc. of Third IEEE Int. Conf. on Fuzzy Systems.* June 26, pp. 524-529.

Akbarzadeh, T., Kumbla, K.K., Jamshidi, M., and Colbaugh, R. 1994a. Intelligent control of flexible and redundant robots in *Waste Management from Risk to Remediation*. Rohinton K. Bhada, Abbas Ghassemi, Timothy J. Ward, Mohammed Jamshidi, and Mohsen Shahinpoor, editors. ECM Press, Albuquerque, NM.

Alang-Rashid, N.K. 1992. Dissertation, "Nuclear Reactor Control Using Tunable Fuzzy Logic Controllers," Nuclear Engineering Dept. of the University of New Mexico.

Alang-Rashid, N.K. and Heger, A.S. 1992a. A method for transforming operators' skill to fuzzy logic controllers. *Control Theory and Advanced Technology,* Vol. 8, No. 23.

Alang-Rashid, N.K. and Heger, A.S. 1992b. Tuning of fuzzy logic controllers by parameter estimation method. In *Environmental and Intelligent Manufacturing Systems,* Vol. 3, *Fuzzy Logic and Control-Software and Hardware Applications,* ed. M. Jamshidi. Prentice Hall, Englewood Cliffs, NJ.

Anderson, B.O.D. 1982. Transfer function matrix description of decentralized fixed modes. *IEEE Trans. Automatic Control,* AC-27:1176-1182.

Anderson, B.O.D., and Clements, D. J. 1981. Algebraic characterization of fixed modes in decentralized control. *IFAC J. Automatica* 17:703-712.

Anderson, B.O.D., and Moore, J. B. 1981. Time-varying feedback laws for decentralized control. *IEEE Trans. Aut. Cont.* AC-26:1133-1138. (See also *Proc. 19th IEEE CDC,* Albuquerque, NM.)

Anderson, J. H. 1967. Geometrical approach to reduction of dynamical systems. *Proc. IEE* 114:1014-1018.

Aoki, M. 1968. Control of large-scale dynamic systems by aggregation. *IEEE Trans. Aut. Cont.* AC-13:246-253.

Aoki, M. 1971. Aggregation, in D. A. Wismer, ed., *Optimization Methods for Large-Scale Systems...with Applications,* chapter 5, pp. 191-232. McGraw-

controlengineers.ir

controlengineers.ir
@controlengineers

Hill, New York.

Aoki, M. 1978. Some approximation methods and control of large-scale systems. *IEEE Trans Aut. Cont.* AC-23:173-182.

Aracil, J., Garcia-Cezero, A., Barreiro, A., and Ollero, A. 1988. Stability analysis of fuzzy control systems: A geometrical approach. Ed. C. A. Kulikowski, R. M. Huber (Hrsg.). *AI, Expert Systems and Languages in Modelling and Simulation.* North Holland, Amsterdam, pp. 323-330.

Aracil, J., Garcia-Cezero, A., Barreiro, A., and Ollero, A. 1991. Fuzzy control of dynamical systems: Stability analysis based on the conicity criterion. *Proc., IFSA '91,* Vol. Engineering, Brussels, pp. 5-8.

Aracil, J., Garcia-Cezero, A., and Ollero, A. 1989. Stability indices for the global analysis of expert systems. *IEEE Trans. on Systems, Man and Cybernetics*, 19, pp. 998-1007.

Arafeh, S., and Sage, A. P. 1974a. Multi-level discrete time system identification in large scale systems. *Int. J. Syst. Sci.* 5:753-791.

Arafeh, S., and Sage, A. P. 1974b. Hierarchical system identification of states and parameters in interconnected power system. *Int. J. Syst. Sci.* 5:817-846.

Araki, M. 1978a. Input-Output stability of composite feedback systems. *IEEE Trans. Aut. Cont.*AC-21:254-259.

Araki, M. 1978b. Stability of large-scale nonlinear systems—Quadratic-order theory of composite-system method using M-matrices. *IEEE Trans. Aut. Cont.* AC-23:129-142.

Araki, M., and Kondo, B. 1972. Stability and transient behavior of composite nonlinear systems. *IEEE Trans. Aut. Cont.* AC-17:537-541.

Ardema, M. D. 1974. Singular perturbations in flight mechanics. NASA, TMX-62, p. 380.

Asatani, K., Iwazumi, T., and Hattori, Y. 1971. Error estimation of prompt jump approximation by singular perturbation theory. *J. Nucl. Sci. Technol.* 8:653-656.

Åström, K.J. and Wittenmark, B. 1991. *Computer Controlled Systems, Theory, and Design.* Prentice Hall, Englewood Cliffs, NJ.

Avramovic, B. 1979. Iterative algonthms for the time scale separation of linear dynamical systems, L. H. Fink and T. A. Trygar, eds., in *Systems Engineering for Power: Organizational Forms for Large-Scale Systems,* pp. 1.10-1.12, Vol. II, US DOE, Washington, DC.

Bailey, F. N. 1966. The application of Lyapunov's second method to interconnected systems. *SIAM J. Contr.* 3:443-462.

Bailey, F. N., and Ramapriyan, H. K. 1973. Bounds on suboptimality in the control of linear dynamic systems. *IEEE Trans. Aut. Cont.* AC-18:532-534.

Barak, D.R. 1993. "Real-time fuzzy logic-based control of industrial systems," master's thesis, EE Dept., Univ. of New Mexico.

Barnett, S., and Storey, C. 1970. *Matrix Methods in Stability Analysis.* T. Nelson, London.

Barreiro, A. and J. Aracil, 1992. Stability of uncertain dynamical systems. *Proc.,*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

IFAC Symp. on AI in Real-Time Control, Delft, pp.177-182.

Bauman, E. J. 1968. Multi-level optimization techniques with application to trajcctory decomposition, in C. T. Leondes, ed., *Advances in Control Systems,* pp. 160-222.

Beck, M. B. 1974. The application of control and systems theory to problems of river pollution. Ph.D. thesis, University of Cambridge, Cambridge, England.

Bellman, R. 1970. *Introduction to Matrix Analysis*, McGraw-Hill, New York.

Benitez-Read, J. S. 1993. Advanced Control Architectures for Nuclear Reactors, Ph.D. Dissertation, CAD Laboratory, Dept. EECE, Univ. of New Mexico, Albuquerque, NM.

Benitez-Read, J. S., and Jamshidi, M. 1992. Adaptive Input-Output Linearizing Control of Nuclear Reactor, *C-TAT Journal*, Vol. 8, No. 3, pp. 525-545.

Benitez-Read, J. S., Jamshidi, M., and Kisner, R. A. 1992. Advanced Control Designs for Nuclear Reactors, *C-TAT Journal*, Vol. 8, No. 3, pp. 447-464.

Benveniste, A., Bernhard, P., and Cohen, A. 1976. On the decomposition of stochastic control problems. IRIA Research Report No. 187, France.

Berkan, R.C. and Upadhyaya, B.R. 1988. Dynamic modelling of EBR-2 for simulation and control. Research Annual Report, UTNE/BRU, Department of Nuclear Engineering, University of Tennessee, Knoxville.

Bhandarkar, M. V., and Fahmy, M. M. 1972. Controllability of tandem connected systems. *IEEE Trans. Aut. Cont.* AC-17:150-151.

Birkhoff, G., and Rota, G. C. 1962. *Ordinary Differential Equations.* Blaisdell, Waltham, MA.

Bode, H. W. 1940. Feedback amplifier design. *Bell Syst. Tech. J.* 19.

Böhm, R. 1992. Ein ansatz zur stabilitätasalyse von fuzzy-reglern. *Forschungsberichte Universität Dortmund, Fakultät fur Elektrotechnik, Band Nr. 3,2. Workshop Fuzzy Control des GMA-UA* 1.4.2. am 19/20.11.1992, pp. 24-35.

Bosley, M. J., and Lees, F. P. 1972. A survey of simple transfer function derivations from high-order state-variable models. *IFAC J. Automatica* 8:765-775.

Bouslama, F., and Ichikawa, A. 1992. Application to limit fuzzy controllers to stability analysis. *Fuzzy Sets and Systems*, 49, pp. 103-120.

Braee, M. and Rutherford, D.A. 1978. Selection of parameters for a fuzzy logic controller. *Fuzzy Set and Systems*, 2, pp. 185-199.

Braee, M. and Rutherford, D.A. 1979. Theoretical and linguistic aspects of the fuzzy logic controller. *Automatica*, 15, pp. 553-477.

Brasch, F. M., and Pearson, J. B. 1970. Pole placement using dynamic compensators. *IEEE Trans. Aut. Cont.* AC-15:34-43.

Brasch, F. M., Howze, J. W., and Pearson, J. B. 1971. On the controllability of composite systems. *IEEE Trans. Aut. Cont.* AC-16:205-206.

Bretthauer, G. and Opitz, H.-P. 1994. Stability of fuzzy systems. *Proc. EUFIT '94*. Aachen, Germany, Sept., 1994, pp. 283-290.

Brockett, R. W. 1970. *Finite Dimensional Linear Systems.* Wiley, New York.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Brogan, W. L. 1991. *Modern Control Theory,* pp. 1-7. Prentice Hall, Englewood Cliffs, NJ.

Bruinzeel, J., Jamshidi, M., and Titli, A. 1995. A sensory fusion-hierarchical real-time fuzzy control approach for complex systems. *Intelligent Control Systems Group LAAS-CNRS*. Toulouse, France.

Bühler, H. 1993. Stabilitatsuntersuchung von Fuzzy-Regelungssystemem. *Proc., 3. Workshop Fuzzy Control des GMA-UA* 1.4.1., Dortmund, pp. 1-12.

Chen, C.-L., Chen, P.-C. and Chen, C.-K. 1993. Analysis and design of a fuzzy control system. *Fuzzy Sets and Systems*, 57, pp. 125-140.

Chen, C. T. 1970. *Introduction to Linear System Theory.* Holt, Rinehart and Winston, New York.

Chen, C. T., and Desoer, C. A. 1967. Controllability and observability of composite systems. *IEEE Trans. Aut. Cont.* AC-12:402-409.

Chen, Y. Y. 1987. Stability analysis of fuzzy control—a Lyapunov approach. *IEEE Annual Conference Systems, Man, and Cybernetics*, Vol. 3, pp. 1027-1031.

Chen, Y. Y. and Tsao, T.C. 1989. A description of the dynamical behaviour of fuzzy systems. *IEEE Trans. on Systems, Man, and Cybernetics,* 19, pp. 745-755.

Cheneveaux, B. 1972. Contribution a l'optimisation hierarchisee des systemes dynamiques. Doctor Engineer Thesis, No. 4. Nantes, France.

Chidambara, M. R. 1969. Two simple techniques for simplifying large dynamic systems. *Proc. JACC* (Univ. of Colorado, Boulder, CO).

Chipman, J. S. 1976. Estimation and aggregation in econometrics: An application of the theory of generalized inverse, in *Generalized Inverse and Applications.* Academic Press, New York.

Cohen, D. S., ed. 1974. Mathematical aspects of chemical and biochemical problems and quantum chemistry. *SIAM-AMS Proceedings,* American Math. Soc., Providence, RI.

Cohen, G., Benveniste, A., and Bernhard, P. 1972. Commande hierarchisee avec coordination en ligne d'un systeme dynamique. *Renue Francaise d'Automatique, Informatique et Recherche Operationnelle.* J4:77-101.

Cohen, G., Benveniste, A., and Bernhard, P. 1974. Coordination algorithms for optimal control problems Part I. Report No. A/57, Centre d'Automatique, Ecole des Mines, Paris.

Cohen, G., and Jolland, G. 1976. Coordination methods by the prediction principle in large dynamic constrained optimization problems. *Proc. IFAC Symposium on Large-Scale Systems Theory and Applications,* Udine, Italy.

Cook, P. A. 1974. On the stability of interconnected systems. *Int. J. Contr.* 20:407-416.

Corfmat, J. P., and Morse, A. S. 1975. Structurally controllable and structurally canonical systems. Internal Report, Dept. Engr. and Appl. Sci., Yale University, New Haven, CT.

Corfmat, J. P., and Morse, A. S. 1976a. Decentralized control of linear multi-variable

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

systems. *IFAC J. Automatica* 12:479-495.

Corfmat, J. P., and Morse, A. S. 1976b. Control of linear system through specified input channels. *SIAM J. Contr.* 14.

Cruz, J. B., Jr. 1976. Stackelberg strategies for multilevel systems, in Y. C. Ho and S. K. Mitter, eds., *Directions in Large Scale Systems.* Plenum, New York.

Cruz, J. B., Jr. 1978. Leader-follower strategies for multilevel systems. *IEEE Trans. Aut. Cont.* AC-23:244-255.

Cuk, S. M., and Šiljak, D. D. 1973. Decomposition-aggregation stability analysis of the spinning skylab. *Proc. 7th Asilomur Conf. on Circuits, Systems and Computers* (Pacific Grove, CA).

Dantzig, G., and P. Wolfe. 1960. Decomposition principle for linear programs. *Oper. Res.* 8:101-111.

Davidson, A. M., and Lucas, T. N. 1974. Linear system reduction by continued fraction expansion about a general point. *Elect. Letters* 10:271-273.

Davison, E. J. 1966. A method for simplifying linear dynamic systems. *IEEE Trans. Aut. Cont.* AC-11:93-101.

Davison, E. J. 1967. Comments on: A method for simplifying linear dynamic systems. *IEEE Trans. Aut. Cont.* AC-12:119-121.

Davison, E. J. 1968. A new method for simplifying linear dynamic systems. *IEEE Trans. Aut. Cont.* AC-13:214-215.

Davison, E. J. 1974. The decentralized stabilization and control of a class of unknown nonlinear time-varying systems. *IFAC J. Automatica* 10:309-316.

Davison, E. J. 1976. Connectability and structural controllability of composite systems. *Proc. IFAC Symposium on Large Scale Systems,* Udine, Italy.

Davison, E. J. 1976a. Decentralized stabilization and regulation in large multivariable systems, in Y. C. Ho and S. K. Mitter, Eds., *Directions in Large-Scale Systems,* pp. 303-323. Plenum Press, New York.

Davison, E. J. 1976b. Multivariable tuning regulators: The feed-forward and robust control of a general servomechanism problem. *IEEE Trans. Aut. Cont.* AC-21:35-47.

Davison, E. J. 1976c. The robust decentralized control of a general servomechanism problem. *IEEE Trans. Aut. Cont.* AC-21:14-24.

Davison, E. J. 1976d. The robust control of a servomechanism problem for linear time-invariant multivariable systems. *IEEE Trans Aut. Cont.* AC-21:25-34.

Davison, E. J. 1977. Connectability and structural controllability *of* composite systems. *IFAC J. Automatica* 13:109-123.

Davison, E. J. 1978. Decentralized robust control of unknown systems using tuning regulators. *IEEE Trans. Aut. Cont.* AC-23:276-289.

Davison, E. J. 1979. The robust decentralized control of a servomechanism problem for composite systems with input-output interconnections. *IEEE Trans. Aut. Cont.* AC-24:325-327.

Davison, E. J., and Gesing, W. 1979. Sequential stability and optimization of large scale decentralized systems. *IFAC J. Automatica* 15:307-324.

Davison, E. J., and Maki, M. C. 1973. The numerical solution of the matrix

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Riccati differential equation. *IEEE Trans. Aut. Cont.* AC-18:71-73.

Davison, E. J., and Man, F. T. 1968. The numerical solution of $ATQ + QA =$. *IEEE Trans. Aut. Cont.* AC-13:448-449.

Davison, E. J., and Ozgüner, U. 1983. Characterization of decentralized fixed modes for interconnected systems. *Automatica,* Vol. 19, No. 2, pp.169-182.

Demaya, B., Boverie, S., and Titli, A. 1994. Stability Analysis of Fuzzy controllers via cell-to-cell root locus analysis. *Proc. EVFIT '94*, Aachen, Germany, pp. 1168-1174.

Deo, N. 1974. *Craphy Theory with Applications to Engineenng and Computer Science* pp. 206-226. Prentice Hall, Englewood Cliffs, NJ.

Desoer, C. A., and Vidyasagar, M. 1975. *Feedhack Spstems: Input-Output Properties.* Academic Press, New York.

Dorato, P. 1963. On sensitivity in optimal control systems. *IEEE Trans. Aut. Cont.* AC-8:256-257.

Dreier, M.E. 1991. FULDEK-Fuzzy Logic Development Kit, TSI Press, Albuquerque, NM.

Dreier, M.E. 1994a. A fast, non-iterative method to generate fuzzy inference rules from observed data. Technical report awaiting publication. Bell Helicopter Textron, Inc., Texas, USA.

Dreier, M.E. 1994b. User's manual for fuzzy logic development kit. Bell Helicopter Textron, Inc., Texas, USA.

Dreier, M.E. and Cunningham, G.A. 1996. The Fuzzy Logic Development Kit for IBM PCs and Compatibles (Chap. 52). *The Handbook of Software for Engineers and Scientists,* ed. Paul W. Ross. CRC Publishers.

Driankov, D., Hellendoorn, H. and Reinfrank, M. 1993. *An Introduction to Fuzzy Control.* Springer-Verlag, Berlin.

Edgar, T. F. 1975. Least squares model reduction using step response. *Int. J. Contr.* 22:261-270.

Evans, W. R. 1950. Control system synthesis by root locus methods. *AIEE Trans.* 69:66-69.

Fei, J. and Isik, C. 1992. Adaptive fuzzy control via modification of linguistic variables. *IEEE Int. Conf. Fuzzy Systems*, San Diego, pp. 399-406.

Fletcher, R., and Powell, M. J. D. 1963. A rapidly convergent descend method for minimization. *Computer J.* 6:163-168.

Franke, D. 1993. Fuzzy control with Lyapunov Stability. *Proc. European Control Conference,* Groningen.

Franklin, G.F. and J. D. Powell 1980. *Digital Control of Dynamic Systems.* Addison-Wesley, Reading, MA.

Frobenius, G. 1912. Uber matrizen mit nicht negativen elementen. *Berlin Akad.* 23:456-477.

Galiana, F. D. 1973. On the approximation of multiple-input multiple-output constant linear systems. *Int. J. Contr.* 17:1313-1324.

Garcia, G. and Bernussou, J. 1995. Pole assignment for uncertain systems in a specified disk by state feedback. *IEEE Trans. Atom. Cont. 40.*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Garrard, W. L. 1969. Additional results on suboptimal feedback control on nonlinear system. *Int. J. Cont.* 10:657-663.

Garrard, W. L. 1972. Suboptimal feedback control for nonlinear system. *IFAC J. Automatica* 8:219-221.

Garrard, W. L., and Jordan, J. M. 1977. Design of nonlinear automatic flight control systems. *IFAC J. Automatica* 13:497-505.

Garrard, W. L., McClamroch, N. H., and Clark, L. G. 1967. An approach to suboptimal feedback control of nonlinear systems. *Int. J. Cont.* 5:425 435.

Gelb, A., ed. 1974. *Applied Optimal Estimation.* MIT Press, Cambridge, MA.

Genesio, R., and Milanese, M. 1976. A note on the derivation and use of reducedorder models. *IEEE Trans. Aut. Cont.* AC-21:118-122.

Geoffrion, A. M. 1970. Elements of large-scale mathematical programming: Part I, Concepts, Part II, Synthesis of Algorithms and Bibliography. *Management Sci.* 16.

Geoffrion, A. M. 1971a. Duality in non-linear programming. *SIAM Review* 13:1-37.

Geoffrion, A. M. 1971b. Large-scale linear and nonlinear programming, in D. A. Wismer, ed., *Optimization Methods for Large Scale Systems.* McGraw-Hill, New York.

Geromel, J. C., and Bernussou, J. 1979. An algorithm for optimal decentralized regulation of linear quadratic interconnected systems. *IFAC J. Automatica* 15:489-491.

Gertler, J. and Chang, H. W. 1986. An instability indicator for expert control. *IEEE Trans on Control Systems,* Heft, pp. 14-17.

Gilbert, E. G. 1963. Controllability and observability in multivariable control systems. *SIAM J. Contr.* 2:128-151.

Glover, K., and Silverman, L. M. 1975. Characterization of structural controllability Internal Report, Dept. Elec. Engr., Univ. S. California, Los Angeles, CA.

Godbout, L. F. 1974. Pole placement algorithms for multivariable systems. M.S. thesis., Dept. Elect. Engr., Univ. of Conn., Tech. Report 74-3.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley.

Grasselli, O. M. 1972. Controllability and observability of series connections of systems. *Ricerche di Automatica* 3:44 53.

Grujić, L. T. and Šiljak, D. D. 1973a. On the stability of discrete composite systems. *IEEE Truns. Aut. Cont.* AC-18:522-524.

Grujić, L. T., and Šiljak, D. D. 1973b. Asymptotic stability and unstability of large-scale system. *IEEE Trans. Aut. Cont.* AC-18:636-645.

Haimes, Y. Y. 1977. *Hierarchical Analyses of Water Resources Systems.* McGraw-Hill, New York.

Hammarling, S.J. 1982. "Numerical Solution of the Stable Non-Negative Definite Lyapunov Equation," *IMAJ. Num. Analysis*, Vol. 2, pp. 303-323.

Han, H. -S. and Lee, J.-G. 1994. Necessary and sufficient conditions for stability of time-varying discrete interval matrices. *Int. Journal of Control,* Vol. 59, pp.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

1021-1029.

Hassan M. F., and Singh, M. G. 1976. A two-level costate prediction algorithm for nonlinear systems. Cambridge University Engineering Dept. Report No. CUEDF-CAMS/TR(124).

Hassan, M. F., and Singh, M. G. 1976. The optimization of nonlinear systems using a new two level method. *IFAC J. Automatica* 12:359-363.

Hassan, M. F., and Singh, M. G. 1981. Hierarchical successive approximation algorithms for non-linear systems. Part II. Algorithms based on costate coordination. *J. Large Scale Systems.* 2:81-95.

Hassan, M. F., Singh, M. G., and Titli, A. 1979. Near optimal decentralized control with a pre-specified degree of stability. *IFAC J. Automatica.* 15:483-488.

Hautus, M. L. J 1975. Input regularity of cascaded systems. *IEEE Trans. Aut. Cont.* Vol. 26, pp. 1-18

Heineken, F. G., Tsuchiya, H. M., and Aris, R. 1967. On the mathematical status of the pseudo-steady-state hypothesis of biochemical kenetics. *Math. Biosciences* 1:95.

Hewlett-Packard Company, 1979. *Interpolation: Chebyschev polynomial.* System 9845 Numerical Analysis Library Part No. 9282-0563, pp. 247-256.

Hickin, J., and Sinha, N. K. 1980. Model reduction for linear multivariable systems. *IEEE Trans. Aut. Cont.* AC-25:1121-1127.

Hirzinger, G., and Kreisselmeier, G. 1975. On optimal approximation of high-order linear systems by low-order models. *Int. J. Contr.* 22:399-408.

Ho, Y. C., and S. K. Mitter, eds. 1976. *Directions in Large-Scale Systems, pp. v-x.* Plenum, New York.

Hojo, T., Terano, T., and Masui, S. 1991. Stability analysis of fuzzy control systems. Proc. IFSA '91, Vol. Engineering, Brussels, pp. 44-49.

Homaifar, A. 1994. Genetic algorithms. Tutorial No. 1, World Automation Congress, Maui, HI.

Huang, P., and Sundareshan, M. K. 1980. A new approach to the design of reliable decentralized control schemes for large-scale systems. *Proc. IEEE Int. Conf. Circuits und Systems* (Houston, TX), pp. 678-680.

Huang, S. -N. and H. -H. Shao 1994a. Heirarchical control in Taylor-based variables parameterization. *IEEE Trans. on Aut. Control*, submitted.

Huang, S. -N. and H. -H. Shao 1994b. New method to heirarchical control of large-scale systems, *IFAC J. Automatica*, submitted.

Hush, D. and Home, B.. 1990. *An Introduction to the Theory of Neural Networks. Report No. EECE 90-005.* Dept. EECE, University of New Mexico Press, Albuquerque, NM.

Hwang, G.-C. and Liu, S.C. 1992. A stability approach to fuzzy control design for nonlinear systems. *Fuzzy Sets and Systems*, 48, pp. 279-287.

IEEE Control Systems Magazine, Letters to the Editor, 1993. IEEE, Vol.13.

Ikeda, M., and Šiljak, D. D. 1980a. Decentralized stabilization of linear time-varying systems. *IEEE Trans. Aut. Cont.* AC-25:106-107.

Ikeda, M., and Šiljak, D. D. 1980b. On decentrally stabilizable large-scale sys-

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

tems. *IFAC J. Automatica* 16:331-334.

Ioannou and Kokotović. 1983. Adaptive systems with reduced models. Springer-Verlag, Berlin.

Iyer, S. N., and Cory, B. J. 1971. Optimization of turbo-generator transient performance by differential dynamic programming. *IEEE Trans. Power. Appar. Syst.* PAS-90:2149-2157.

Jamshidi, M. 1969. A near-optimal controller for nonlinear systems. *Proc. 7th Allerton Conf. on Circuits and Systems* (Monticello, IL), pp. 169-180.

Jamshidi, M. 1972. A near-optimum controller for cold-rolling mills. *Int. J. Contr.* 16:1137-1154.

Jamshidi, M. 1974. Three-stage near-optimum design of nonlinear control processes. *Proc. IEE* 121:886-892.

Jamshidi, M. 1975. Optimal control of nonlinear power systems by an imbedding method. *IFAC J. Automatica.* 11:633-636.

Jamshidi, M. 1976. A feedback near-optimum control for nonlinear systems. Information Control, Vol. 32(1).

Jamshidi, M. 1980. An overview on the solutions of the algebraic matrix Riccati equation and related problems. *J. Large Scale Systems* 1:167-192.

Jamshidi, M. 1983. *Large-Scale Systems -Modeling and Control.* Elsevier North-Holland, New York.

Jamshidi, M. 1984. "An overview on the aggregation of Large-Scale Systems," *Proc. VIII IFAC Congress*, Kyoto, Japan.

Jamshidi, M. 1989. "Introduction to Large-Scale Systems," Chapter 6 in *Simulation of Dynamic Systems*. N. Kheir (ed.) Marcel Dekker, New York.

Jamshidi, M. 1994. On software and hardware application of fuzzy logic (Chap. 20). *Fuzzy Sets, Neural Networks, and Soft Computing.* ed. R.R. Yager and L.A. Zadeh. Van Nostrand Reinhold, NY, pp. 396-430.

Jamshidi, M., and Malek-Zavarei 1986. *Linear Control System: A Computer-aided Approach.* Pergamon Press, Oxford, England.

Jamshidi, M., Nguyen, C.C., Lumia, R. and Yuh, J. 1994. *Robotics and Manufacturing—Recent Trends on Environment, Intelligence, and Applications.* ASME Press, New York.

Jamshidi, M., Tarokh, M., and Shafai, B.1992. *Computer-aided Analysis and Design of Linear Control Systems*, Prentice Hall, Englewood Cliffs, N.J.

Jamshidi, M., Vadiee, N. and Ross, T. J. (eds.). 1993. *Fuzzy Logic and Control: Software and Hardware Applications.* Vol.2, Prentice Hall Series on Environmental and Intelligent Manufacturing Systems, (M. Jamshidi, ed.). Prentice Hall, Englewood Cliffs, NJ.

Jang, J.S.R. 1993. Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man. Cyber.,* Vol. 23, No. 3, pp. 665-685.

Javdan, M.R. 1976a. On the use of lagrange duality in multi-level optimal control. *Proc. IFAC Symposium on Large Scale Systems Theory and Appl.,* Udine, Italy.

Javdan, M.R. 1976b. Extension of dual coordination to a class of nonlinear

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

systems. *Int. J. Contr.* 24, pp. 551-571.

Kalman, R. E. 1960. Contribution to the theory of optimal control. *Bol. Soc. Math Mexicana* 5:102-119.

Kalman, R. E. 1962. Canonical structure of linear dynamical systems. *Proc. Nat. Acad. Sci.* 48:596-600.

Kang, H. 1993. Stability and control of fuzzy dynamic systems via cell-state transitions in fuzzy hypercubes. *IEEE Trans. on Fuzzy Systems,* 1, pp. 267-279.

Kelley, H. J., and Edelbaum, T. N. 1970. Energy climbs, energy turns and asymptotic expansions. *J. Aircraft* 7:93-95.

Kelly, J. J. 1964. An optimal guidance approximation theory. *IEEE Trans. Aut. Cont.* 9:375.

Kickert, W. J. and Mamdani, E.H. 1978. Analysis of a fuzzy logic controller. *Fuzzy Sets and Systems*, 1, pp. 29-44.

Kitamori, T. 1979. A method of control system design based upon partial knowledge about controlled processes. *Trans. SICE, Japan,* Vol. 15, pp. 549-555.

Kiszka, J.B., Gupta, M.M., and Nikiforuk, P.N. 1985. Energistic Stability of Fuzzy Dynamic Systems. *IEEE Trans on Systems, Man, and Cybernetics,* 15, pp. 783-792.

Klamka, J. 1972. Uncontrollability and unobservability of multivariable systems. *IEEE Trans. Aut. Cont.* AC-17:725-726.

Klamka, J. 1974. Uncontrollability of composite systems. *IEEE Trans. Aut. Cont.* AC-19:280-281.

Kokotović, P. V. 1972. Feedback design of large linear systems, in J. B. Cruz, Jr., ed., *Feedback Systems,* pp. 99-137. McGraw-Hill, New York.

Kokotović, P. V. 1979. Overview of multimodeling by singular perturbations, in L. H. Fink and T. A. Trygar, eds., *Systems Engineering for Power: Organizational Forms for Large-Scale Systems,* pp. 1.3-1.4. US DOE, Washington, DC.

Kokotović, P. V., Allemong, J. J., Winkelman, J. R., and Chow, J. H. 1980. Singular perturbation and iterative separation of time scales. *IFAC J. Automatica* 16:23-34.

Kokotović, P. V., and Cruz, J. B., Jr. 1969. An approximate theorem for linear optimal regulator. *J. Math. Anal. Appl.* 27:249-252.

Kokotović P. V., Cruz, J. B., Jr., Heller, J. E.: and Sannuti, P. 1969a. Synthesis of optimally sensitive systems. *Proc. IEEE* 56:13-18.

Kokotović, P. V., and D'Ans, G. 1969. Parameter imbedding design of linear optimal regulators. *Proc. 3rd Princeton Conf.* (Princeton, NJ), pp. 378-379.

Kokotović, P. V., and Haddad, A. H. 1975. Controllability and time-optimal control of systems with slow and fast modes. *IEEE Trans. Aut. Cont. AC-20*:111-113.

Kokotović, P. V., O'Malley, R. E., Jr., and Sannuti, P. 1976. Singular perturbations and order reduction in control theory—An overview. *IFAC J. Automatica* 12:123-132.

Kokotović, P. V., Perkins, W. R., Cruz, J. B., Jr., and D'Ans, G. 1969b. e-coupling

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

method for near optimum dosing of large scale linear systems. *Proc. IEE* 116:889-892.

König, D. 1931. Graphak es matrixok. *Mater Fiziol Lapok.* 38:116-119.

Kosko, B. 1992. *Neural Networks and Fuzzy Systems: a Dynamical Approach to Machine Intelligence.* Prentice Hall, Englewood Cliffs, NJ.

Kruse, R., Gebhardt, J., and Klawonn, F. 1994. Foundations of fuzzy systems. John Wiley & Sons, Chichester, West Sussex, U.K.

Kumbla, K.K., and Jamshidi, M. 1995. Hierarchical fuzzy control of robot manipulator. *Journal of Intelligent and Fuzzy Systems*, Vol. 3, pp. 21-29.

Kwakernaak, H. W., and Sivan, R. 1972. *Linear Optimal Control Systems.* Wiley, New York.

Lamba, S. S., and Rao, S. V. 1978. Aggregation matrix for the reduced-order continued fraction expansion model of Chen and Shieh. *IEEE Trans. Aut. Cont.* AC-23:81-83.

Langari, G., and Tomizuka, M. 1990. Stability of fuzzy linguistic control systems. *Proc. IEEE Conf. Decision and Control,* Hawaii, pp. 2185-2190.

Lasley, E. L., and Michel, A. N. 1976a. Input-Output stability of interconnected systems. *IEEE Trans. Aut. Cont.* AC-21:84-89.

Lasley, E. L., and Michel, A. N. 1976b. Input-Output stability of interconnected systems. *IEEE Trans. Cir. Syst.* CAS-23:261-270.

Laub, A. J. 1979. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Aut. Cont.* AC-24:913-921.

Laub, A. J. 1980. Computation of balancing transformations. *Proc. JAAC.*

Laub, A. J., and Bailey, F. N. 1976. Suboptimality bounds and stability in the control of nonlinear dynamic systems. *IEEE Trans. Aut. Cont.* AC-21:396-399.

Lee, E. B., and Markus, L. 1967. *Foundations of Optimal Control Theory.* Wiley, New York.

Lee, G, Jordan, D., and Sohrwardy, M. 1979. A pole assignment algorithm for multivariable control systems. *IEEE Trans Aut. Cont.* AC-24:357-362.

Lee, G., Jordan, D., and Sohrwardy, M. 1980. Author's reply. *IEEE Trans. Aut. Cont.* AC-25:140.

Lefschetz, S. 1965. *Stability of Nonlinear Control Systems.* Academic Press, New York.

Lin, C. T. 1974. Structural controllability. *IEEE Trans. Aut. Cont.* AC-19:201 208.

Lin, Y. and Cunningham, G.A.1994. Building a fuzzy system from input-output data. *J. of Intelligent and Fuzzy Systems,* Vol. 2, No. 3, pp. 243-250.

Liu, C. L. 1968. *Introduction to Combinatorial Mathematics,* chapter 11. McGraw-Hill, New York.

Ljung, L. 1987. *Systern Identification Theory for the User.* Prentice Hall, Englewood Cliffs, NJ.

Ljung, L. 1988. *Systern Identification Toolbox.* Mathworks, Inc. South Natick, MA.

Luenberger, D. G. 1977. Dynamic equations in descriptor form. *IEEE Trans. Aut.*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Cont. AC-22:312-322.

Luenberger, D. G. 1978. Time-invariance descriptor systems. *IFAC J. Automatica* 14:473-480.

Luenberger, D. G. 1984. *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA.

Luger, G.F., and W. A. Stubblefield. 1989. *Artificial Intelligence and the Design of Expert Systems.* Benjamin/Cummings, Redwood City, CA.

Mahmoud, M. S. 1977. Multilevel systems control and applications: A survey. *IEEE Trans. Sys. Man. Cyb.,* SMC-7:125-143.

Mahmoud, M. S., and Singh, M. G. 1981. *Large-Scale Systems Modelling.* pp. 156 166. Pergamon Press, Oxford, England.

Malek-Zavarei, M., and Jamshidi, M. 1975. Sensitivity of linear time-delay systems to parameter variations. *IFAC J. Automatica* 11:315-319.

Malinvaud, E. 1956. L'Aggregation dans les models economiques. *Cahiers du Semnaire d'Econometrive* 4:69-146.

Mamdani, E.H. 1974. Applications of fuzzy algorithms for simple dynamic plant. *Proc. IEE,* 121, No. 12, pp. 1585-1588.

Mamdani, E.H. and Assilian, S. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Machine Studies,* 7, No. 1, pp. 1-13.

March, J. G., and Simon, H. A. 1958. *Organizations.* Wiley, New York.

Matrosov, V. M. 1972. Method of Lyapunov vector functions in feedback systems. *Aut. Remote Contr.* 33:1458-1469.

Mayne, D. Q. 1976. Decentralized control of large-scale systems, in Y. C. Ho and S. K. Mitter, eds., *Directions in Large Scale Systems,* pp. 1723. Plenum, New York.

McClamroch, N. H., Clark, L. G., and Aggarwal, J. K. 1969. Sensitivity of linear control systems to large parameter variations. *IFAC J. Automatica* 5:257-263.

Meier, L. and Luenberger, D. G. 1967. Approximation of linear constant systems. *IEEE Trans. Aut. Cont.* AC-12:585-588.

Mesarovic, M. D., Macko, D., and Takahara, Y. 1970. *Theory of Hierarchical Multilevel Systems.* Academic Press, New York.

Mesarovic, M. D., Macko, D., and Takahara, Y. 1969. Two coordination principles and their applications in large-scale systems control. *Proc. IFAC Congr.* Warsaw, Poland.

Michailesco, G., and J. M. Siret. 1980. Comments on "Aggregation matrix for the reduced-order continued fraction expansion model of Chen and Shieh" and authors' reply, *IEEE Trans. Aut. Cont.* AC-25:133-134.

Michel, A. N. 1975a. Stability and trajectory behavior of composite systems. *IEEE Trans.Cir. Syst.* CAS-22:305-312.

Michel, A. N. 1975b. Stability analysis of stochastic large-scale systems. Z.*Angew. Math. Mech.* 55:93-105.

Michel, A. N., and Porter, D. W. 1972. Stability analysis of composite systems. *IEEE Trans. Aut. Cont.* AC-17:111-116.

Microsoft, Inc. 1990. Microsoft-C Optimizing Compiler. Redmond, WA.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Milne, R. D. 1965. The analysis of weakly coupled dynamic systems. *Int. J. Contr.* 2:171-199.

Mishra, R. N., and Wilson, D. A. 1980. A new algorithm for optimal reduction of multivariable systems. *Int. J. Contr.* 31:443-466.

Moler, C. 1980. *MATLAB User's Guide.* Department of Computer Science, University of New Mexico, Albuquerque, NM.

Monopoli, R.V. 1974. Model reference adaptive control with an augmented error signal. *IEEE Trans. on Automatic Control,* AC-19:474-484.

Moore, B. C. 1981 Principle component analysis in linear systems: Controllability, observability, and model reduction, *IEEE Trans. Aut. Contr.* AC-26.

Mora-Camino, F. 1979. Comments on Optimization of non-linear systems using a new two level method. *IFAC J. Automatica* 15:125-126.

Morari, M., Stephanopoulos, G., Shields, R. W., and Pearson, J. B. 1978. Comments on finding the generic rank of a structured matrix, and authors' response. *IEEE Trans. Aut. Cont.* AC-23:509-510.

Moylan, P. J. and Hill, D. J. 1978. Stability criteria for large-scale systems. *IEEE Trans. Aut. Cont.* AC-23:143-149.

Mukhopadhyay, B. K., and Malik, O. P. 1973. Solution of nonlinear optimization problem in power systems. *Int. J. Contr.* 17:1041-1058.

Nagaragan, R. 1971. Optimum reduction of large dynamic systems. *Int. J. Contr.* 14:1164-1174.

Narendra, K.S. and Parthasarathy, K. 1990. Identification and Control of Dynamic System Using Neural Networks, *IEEE Transactions on Neural Networks,* Vol. 1, No.1., pp. 4-27.

Narendra, K.S. and Annaswamy, A.M. 1989. Stable adaptive systems. Prentice Hall, Englewood Cliffs, NJ.

Newell, A. and H. A. Simon. 1976. Computer Science as Empirical Inquiry: Symbols and Search. *Comm. of the ACM,* 19:113-126.

Newhouse, A. 1962. Chebyschev curve fit. *Comm. ACM* 5:281.

Nishikawa, Y., Sannomiya, N., and Tokura, H. L. 1971. Suboptimal design of a nonlinear feedback system. *IFAC J. Automatica* 7:703-712.

Nomura, H., Hayashi, I., and Wakami, N. 1992. A learning method of fuzzy inference rules by descent method. IEEE Int. Conf. Fuzzy Systems, San Diego, pp. 203-210.

Nyquist, H. 1932. Regeneration theory. *Bell Syst. Tech. J.* 2.

O'Malley, R. E. 1974. *Introduction to Singular Perturbations.* Academic Prcss, New York.

Opitz, H.P. 1993. Fuzzy control. Teil 6: stabilitat von fuzzy-regelungen. *Automatisierungstechnik*, 41, A21-24.

Opitz, H.P. 1994. Stability analysis and fuzzy control. *Proc. Fuzzy Duisburg '94, Internation Workshop on Fuzzy Technologies in Automation and Intelligent Systems,* Duisburg.

Ortega, J. M., and Rheinboldt, W. C. 1970. *Iterative Solution of Nonlinear Equa-*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

tions in Several Variables, pp. 230-235. Academic Press, New York.

Ortega, R. and Tang, Y. 1989. Robustness of adaptive controllers-a survey, *IFAC J. Automatica*, Vol. 25, pp. 651-677.

Pearson, J. D. 1971. Dynamic decomposition techniques, in Wismer, D. A., ed., *Optimization Methods for Large-Scale Systems.* McGraw-Hill, New York.

Pedrycz, W. 1993. *Fuzzy Control and Fuzzy Systems.* 2nd Extended Edition. Wiley, New York.

Penrose, R. 1955. A generalized inverse for matrices. *Proc. Cambridge Phil. Soc.* 51:406-413.

Perez-Arriaga, I. J., Verghese, G. C., and Schweppe, F. C. 1981. Determination of relevant state variables for selective modal analysis. *Proc. JACC.* Charlottesville, VA. paper TA-4F.

Perkins, W. R. and Cruz, J. B., Jr. 1969. Engineering of Dynamic Systems, pp. 418-421. Wiley, New York.

Peterson, B. B. and Narendra, K. S.1982. Bounded error adaptive control*, IEEE Trans Automat. Contr.* Vol. AC-27, pp. 1161-1168.

Phillips, R. G. 1979. A two-stage design of linear feedback controls, in L. H. Fink and T. A. Trygar, eds., *Systems Engineering for Power: Organizational Forms for Large-Scale Systems,* pp. 1.18-1.21, Vol. II. US DOE, Washington, DC.

Piontkovskii, A. A., and Ruthovskaya, L. D. 1967. Investigation of certain stabilitytheory problems by the vector Lyapunov function method. *Aut. Remote Contr.* 28:1422-1429.

Popov, V. M. 1960. Criterion of quality for nonlinear controlled systems. *Proc. Ist IFAC Congress* (Moscow, USSR), pp. 173-176.

Porter, B. 1976. Necessary and sufficient conditions for the controllability and observability of general composite systems. *Proc. IFAC Symposium on Large Scale Systems* (Udine, Italy), pp. 265-269.

Porter, B. 1980. Comments on A pole assignment algorithm for multivariable control system. *IEEE Trans. Aut. Cont.* AC-25:139-140.

Porter, B., and Bradshaw, A. 1978. Design of linear multivariable continuous-time output feedback regulators. *Int. J. Syst. Sci.* 9:445-450.

Porter, B., Bradshaw, A., and Daintith, D. 1979. EIGENFRTRAC: A software package for the design of multivariable digital control systems. Report USAME/DC/102/79, Univ. of Salford, Salford, England.

Porter, D. W., and Michel, A. N. 1974. Input-output stability of time-varying nonlinear multiloop feedback systems. *IEEE Trans. Aut. Cont.* AC-19:422-427.

Praly, L. 1983. Robustness of model reference adaptive control, *Proc. 3rd Yale Workshop on Appl. of Adaptive Syst. Theory*, Yale Univ., New Haven, CT.

Prescott, R., and Pearson, J. B. 1981. Private communication, Rice University, Houston, TX.

Ragazzini, J. R., and Franklin, G. F. 1958. *Sampled-Data Control Systems.* McGraw-Hill, New York.

Raju, G.U., and Zhou, J. 1993. Adaptive hierarchical fuzzy controller. *IEEE Trans. on SMC,* Vol. 23, No. 4, pp. 973-980.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Raju, G.U., Zhou, J. and Kisner, R.A. 1991. Hierarchical fuzzy control. *Int. Journal of Control,* pp. 1201-1216.

Ramapriyan, H. K. 1970. A study of coupling in interconnected systems. Ph.D. dissertation, Univ. of Minnesota, Minneapolis, MN.

Rao, S. V., and Lamba, S. S. 1974. A new frequency domain technique for the simplification of linear dynamic systems. *Int. J. Contr.* 20:71-79.

Rasmussen, R. D., and Michel, A. N. 1976a. On vector Lyapunov function for stochastic dynamical systems. *IEEE Trans. Aut. Cont.* AC-21:250-254.

Rasmussen, R. D., and Michel, A. N. 1976b. Stability of interconnected dynamical systems described on Banach spaces. *IEEE Trans. Aut. Cont.* AC-21:464-471.

Ray, K.S., Ananda, S.G., and Majumder, D.D. L-stability and the related resign concept for SISO linear systems associated with fuzzy logic controller. *IEEE Trans. on Systems, Man and Cybernetics,* 14, pp. 932-939.

Ray, K.S. and Majumder, D.D. 1984. Application of circle criteria for stability analysis associated with fuzzy logic controller. *IEEE Trans on Systems, Man, and Cybernetics,* 14, pp. 345-349.

Razzaghi, M. and A. Arabshahi 1989. Optimatal control of linear distributed-parameter systems via polynomial series, *Int. J. of Systems Sci.*, Vol. 20, pp. 1141-1148.

Retallack, D. G., and MacFarlane, A. G. J. 1970. Pole-shifting techniques for multivariable feedback systems. *Proc. IEE* 117:1037-1038.

Riggs, J. B., and Edgar, T. F. 1974. Least squares reduction of linear systems impulse response. *Int. J. Contr.* 20:213-223.

Rissanen, J. J. 1966. Performance deterioration of optimum systems. *IEEE Trans. Aut. Cont.* AC-11:530-532.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386-408.

Rosenbrock, H. H. 1960. An automatic method for finding the greatest or least value of a function. *Computer J.* 3: 175-184.

Rosenbrock, H. H. 1970. *State Space and Multivariahle Theory.* Nelson and Sons, London.

Ross, T.J. 1995. *Fuzzy Logic with Engineering Application.* McGraw-Hill, New York.

Rovere, L.A. 1989. A multimodular reactor models for supervisory control studies. ORNL-ACTO Program, Oak Ridge Nat'l Lab., Oak Ridge, TN.

Saeks, R. 1979. On the decentralized control of interconnected dynamical systems. *IEEE Trans. Aut. Cont.* AC-24:269-271.

Saeks, R. and R. A. DeCarlo. 1981. *Interconnected Dynamical Systems.* Marcel Dekker, New York.

Sage, A. P. 1968. *Optimum Systems Control.* Prentice Hall, Englewood Cliffs, NJ.

Sage, A. P. 1977. *Methodologies for Large-Scale Systems.* McGraw-Hill, New York.

Sandell, N. R., Jr., Varaiya, P., Athans, M. J. and Safonov, M. G. 1978. *Survey of*

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

decentralized control methods for large scale systems, IEEE Trans. Aut. Cont. AC-23:108-128 (special issue on large-scale systems).

Santiago, J. J. and Jamshidi, M. 1986. On the extensions of the balanced approach for model reduction, *Int. J. Cont. Theory Adv. Tech.,* Vol. 2, No. 2, pp. 207-226

Sastry, S. and Bodson, M. 1989. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice Hall, Englewood Cliffs, NJ.

Sayyarrodsari, B. and Homaifar, A. 1995. The role of hierarchy in the design of fuzzy logic controllers. Submitted to the *IEEE Trans. on Fuzzy Systems.*

Schoeffler, J. D. 1971. Static multilevel systems, in D. A. Wismer, ed., *Optimization Methods for Large-Scale Systems.* McGraw-Hill, New York.

Schoeffler, J. D., and Lasdon, L. S. 1966. Decentralized plant control. *ISA Trans.* 5:175- 183.

Sezar, E., and Hüseyin, O. 1977. A counter-example on the controllability of composite systems. *IEEE Trans. Aut. Cont.* AC-22:683-684.

Sezer, E., and Hüseyin, O. 1979. On the controllability of composite systems. *IEEE Trans. Aut. Cont.* AC-24:327-329.

Sezer, M. E., and Šiljak, D. D. 1981b. On decentralized stabilization and structure of linear large scale systems. *IFAC J. Automatica* 17:641-644.

Sheikholeslam, -F. 1994. Stability analysis of nonlinear and fuzzy systems. M. SC. Thesis, Department of EECS Isfahan University of Technology, Isfahan, IRAN.

Shields, R. W., and Pearson, J. B. 1976. Structural controllability of multiinput linear systems. *IEEE Trans. Aut. Cont.* AC-21:203-212.

Šiljak, D. D. 1972a. Stability of large-scale systems, *Proc. 5th IFAC World Congress,* Paris, France.

Šiljak, D. D. 1972b. Stability of large-scale systems under structural perturbations. *IEEE Trans. Syst. Man. Cyber.* SMC-2:657-663.

Šiljak, D. D. 1974a. Large-scale systems: Complexity, stability, reliability. *Proc. Utah St. Univ.* (Ames Research Ctr. Seminar Workshop on Large Scale Dynamic Systems, Utah State Univ., Logan, UT).

Šiljak, D. D. 1974b. On the connective stability and instability of competitive equilibrium. *Proc. 1974 JACC* (Austin, TX).

Šiljak, D. D. 1975. When is a complex system stable? *Math. Biosci.* 25:25-50.

Šiljak, D. D. 1976. Large-scale systems: Complexity, stability, reliability. *J. Franklin Inst.* 30:49-69.

Šiljak, D. D. 1978. *Large-Scale Dynamic Systems: Stability and Structure,* pp. 68-74. Elsevier North Holland, New York.

Šiljak, D. D. 1978a. On decentralized control of large scale systems. *Proc. IFAC 7th World Congress* (Helsinki, Finland), pp. 1849-1856.

Šiljak, D. D. 1978b. *Large Scale Dynamic Systems.* Elsevier North Holland, New York.

Šiljak, D. D. 1991. *Decentralized Control of Complex Systems.* Academic Press, Boston.

controlengineers.ir

controlengineers.ir
@controlengineers

Šiljak, D. D., and Sundareshan, M. K. 1976. A multilevel optimization of large-scale dynamic systems. *IEEE Trans. Aut. Cont.* AC-21:79-84.

Simmons, M. 1975. The decentralized profit maximisation of inter-connected production systems. Report CUED/F Control/TR101, Cambridge University Engineering Dept., Cambridge, England.

Simon, J. D. and Mitter, S. K. 1968. A theory of modal control. *Inform. Contr.* 13:316-353.

Singh, M. G. 1975. River pollution control. *Int. J. Svst. Sci.* 6:9-21.

Singh, M. 1980. *Dynamical Hierarchical Control.* Rev. Ed. North Holland, Amsterdam.

Singh, M. G., and Hassan, M. 1976. A comparison of two hierarchical optimisation methods. *Int. J. Syst. Sci.* 7:603-611.

Singh, M. G., and Hassan, M. F. 1978. Hierarchical optimization for non-linear dynamical systems with non-separable cost functions. *1FAC J. Automatica* 14:99-101.

Singh, M. G., and Hassan, M. F. 1979. Author's reply. *IFAC J. Automatica* 15:126-128.

Singh, M. G., and Titli, A. 1978. *Systems: Decomposition, Optimization and Control.* Pergamon Press, Oxford, England.

Sinha, N. K., and Berezani, G. T. 1971b. Optimum approximation of high-order systems by low-order models. *Int. J. Contr.* 14:951-959.

Sinha, N. K., and Pille, W. 1971a. A new method for reduction of dynamic systems. *Int. J. Cont.* 14:111-118.

Siret, J. M., Michailesco, G., and Bertrand, P. 1977b. Optimal approximation of high-order systems subject to polynomial inputs. *Int. J. Contr.* 26:963-971.

Slotine, J.E. and Li, W. 1991. *Applied Nonlinear Control.* Prentice Hall, Englewood Cliffs, NJ.

Smith, N. H., and Sage, A. P. 1973. An introduction to hierarchical systems theory. *Computers and Electrical Engineering* 1:55-71.

Strang, G. 1986. *Introduction to Applied Mathematics.* Wellesley, Cambridge, MA.

Stengel, D. N., Luenberger, D. G., Larson, R. E., and Cline, T. S. 1979. A descriptor variable approach to modeling and optimization of large-scale systems, Report No. CONS-2858-TI. US DOE, Oak Ridge, TN.

Stewart, G. W. 1976. Simultaneous iteration method for computing invariant subspaces of non-Hamiltonian matrices. *Num. Math.* 25:123-136.

Suda, N. 1973. Analysis of large-scale systems by decomposition. *Aut. Cont. Technique* 15:3-18 (in Japanese).

Sugeno, M. and Murakami, K. 1985. An experimental study on fuzzy parking control using a model car. *Industrial Applications of Fuzzy Control,* ed. M. Sugeno. North-Holland, Amsterdam.

Sundareshan, M. K. 1977a. Exponential stabilization of large-scale systems: Decentralized and multilevel schemes. *IEEE Trans. Syst. Man. Cyber.* SMC 7:478-483.

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Sundareshan, M. K. 1977b. Generation of multilevel control and estimation schemes for large-scale systems: A perturbation approach. *IEEE Trans. Syst. Man. Cyber.* SMC-7:144-152.

Sundareshan, M. K., and Vidyasagar, M. 1975. L2-stability of large-scale dynamical systems criteria via positive operator theory. Technical Report, Faculty of Engr., Concordia Univ., Canada.

Tahani, V. and Sheikholeslam, F. 1994. Extension of new results on nonlinear systems stability to fuzzy systems. *Proc. EUFIT'94*, Aachen, Germany, pp. 683-686.

Takahara, Y. 1965. A multi-level structure for a class of dynamical optimization problems. M.S. thesis, Case Western Reserve University, Cleveland, OH.

Tanaka, K. and Sugeno, M. 1992. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems,* 45, pp. 135-156.

Tarokh, M. and Jamshidi, M. 1987. Elimination of decentralized fixed modes with minimum number of interconnection gains. *J. of Large Scale Systems,* Vol. II, pp. 207-215.

Thompson, W. E. 1970. Exponential stability of interconnected systems. *IEEE Trans. Aut. Cont.* AC-15:504-506.

Titli, A. 1972. Contribution a l'etude des structures de commande hierarchisees en vue de l'optimization des processes complexes, *These d'Etat.* No. 495, Toulouse, France.

Titli, A., Galy, J.and Singh, M. G. 1975. Methodes de decompositon-coordination en calcul des variations et couplage par variables des etats. *Revue Francaise d' Automatique. Informatique et Recherche Operationnelle.* J4

Togai InfraLogic, Inc. 1991. *Fuzzy-C Development System User's Manual.* Irvine, CA.

Tunstel, E., Akbarzadeh-T, M.R., Kumbla, K., and Jamshidi, M. 1995. Hybrid fuzzy control schemes for robotic systems. *10th IEEE Intl. Symp. on Intelligent Control,* Monterey CA, August 27-29, 1995, pp. 171-176.

Tunstel, E. and Jamshidi, M. 1996. On genetic programming of fuzzy rule-based systems for intelligent control. To appear in *Intl. Journal of Intelligent Automation and Soft Computing,* Vol. 2, No. 2.

Varaiya, P. 1972. Book Review of Mesarovic *et al.* Theory of hierarchical multilevel systems. *IEEE Trans. Aut. Cont.* AC-17:280-281.

Vidyasagar, M. and Viswanadham, N. 1982. Algebraic design techniques for reliable stabilization. *IEEE Trans. Auto. Control,* AC-17:1085-1095.

Wang, B.H. and Vachtsenavos, G. 1992. Learning fuzzy logic control: An indirect control approach. *IEEE Int. Conf. Fuzzy Systems,* San Diego, pp. 297-304.

Wang, L.-X. 1993. Stable adaptivc fuzzy control of nonlinear systems. *IEEE Trans. on Fuzzy Systems,* 1, pp. 146-155.

Wang, L.-X., 1994a. A supervisory controller for fuzzy control systems that guarantees stability. *Proc. IEEE Conf. SMC*, pp. 1035-1039.

Wang, L.-X., 1994b. *Adaptive Fuzzy Systems and Control.* Prentice Hall,

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Engelwood Cliffs, NJ.

Wang, L.X. and Mendel, J.M. 1990. "Generating Fuzzy Rules from Numerical Data with Applications," *Technical Report*, University of Southern California, Los Angeles, CA.

Wang, P.-Z., Zhang, H.-M., and Xu, W. 1990. Pad-analysis of fuzzy control stability. *Fuzzy Sets and Systems*, 38, pp. 27-42.

Wang, S. H., and Davison, E. J. 1973a. On the controllability and observability of composlte systems. *IEEE Trans. Aut. Cont.* AC-18:74-75.

Wang, S. H., and Davison, E. J. 1973b. On the stabilization of decentralized control systems. *IEEE Trans. Aut. Cont.* AC-18:473-478.

Weissenberger, S. 1974. Tolerance of decentrally optimal controllers to nonlinearity and coupling. *Proc. 12th Allerton Conf. Circuits and Systems* (Monticello, IL), pp.87-95.

Werner, R.A. and Cruz, J.B. Jr. 1968. Feedback control which preserves optimality for systems with unknown parameters. *IEEE Trans. Aut. Cont.* AC-13, No. 6, pp. 621-629.

White, R.E. and Cook, G. 1973. Use of piecewise linearization for suboptimal control of non-linear systems. *Int. J. Control,* 18.

Whittle, P. 1971. *Optimization Under Constraints.* Wiley-Interscience, London.

Willems, J. C. 1970. Stability Theory of Dynamical Systems. Nelson, London.

Wilson, D. A. 1970. Optimum solution of model reduction problem. *Proc. IEE* 117:1161-1165.

Wilson, D. A. 1974. Model reduction for multivariable systems. *Int. J. Contr.* 20:57-64.

Wilson, D. A., and Mishra, R. N. 1979. Optimal reduction of multivariable systems. *Int. J. Contr.* 29:267-278.

Wolovich, W. A., and Huang, H. L. 1974. Composite system controllability and observability. *IFAC J. Automatica* 10:209-212.

Xue, H., and N. Chong, 1993. Fuzzy Rule Base Generation Based on Numerical Data, Project Report, CAD Laboratory for Intelligent and Robotic Systems, Department EECE, University of New Mexico, Albuquerque, NM.

Yaekel, R. A., and Kokotović, P. V. 1973. A boundary layer method for the matrix I Riccati equation. *IEEE Trans. Aut. Cont.* AC-18:17-24.

Yamakawa, T. 1991. "Wine-glass Balancing Problem," videotape. Kyushu Institute of Technology. Iizuka, Fukuoka, Japan.

Yamakawa, T. 1994. Private communication. Iizuka, Japan.

Yamakawa, T. and Furukawa, M. 1992. A design algorithm of membership functions for a fuzzy neuron using example-based learning. *IEEE Int. Conf. Fuzzy Systems,* San Diego, pp. 75-82.

Yin, T.K. and George Lee, C.S. 1995. Fuzzy model-reference adaptive control. *IEEE Trans. on Sys., Man, and Cybernetics,* Vol. 25, No. 12, pp. 1606-1615.

Yonemura, Y., and Ito, M. 1972. Controllability of composite systems in tandem connection. *IEEE Trans. Aut. Cont.* AC-17:722-724.

Zadeh, L.A. 1965. Fuzzy sets. *Information and Control,* 8:338-353.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Zames, G. 1966. On input-output stability of time-varying nonlinear systems. Parts I and II. *IEEE Trans. Aut. Cont.* AC-11:228-238, 465-476.

Ziegler, J.G. and N.B. Nichols, 1942. 1942. Optimum settings for automatic controllers. *Trans. ASME*, Vol. 64, pp. 759-768.

Zhang, B.S. and Edmunds, J.M. 1992. Self-organizing fuzzy logic controller. *IEE Proceedings - D,* Vol. 139, No. 5, pp. 460-464.

Zhao, Z. Y., Tomizuka, M., and Isaka, S. 1993. Fuzzy gain scheduling of PID Controllers. *IEEE Trans. on SMC*, Vol. 23, No. 5, pp. 1392-1398.

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

Index

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

D

E

F

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir

اختصاصی مهندسی کنترل
controlengineers.ir
@controlengineers

controlengineers.ir