STUDIES IN *FUZZINESS*
AND *SOFT COMPUTING*

Studies in Fuzziness and Soft Computing

Jacek Kluska

Analytical Methods in Fuzzy Modeling and Control

Springer

Jacek Kluska

Analytical Methods in Fuzzy Modeling and Control

Studies in Fuzziness and Soft Computing, Volume 241

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 225. Terry D. Clark, Jennifer M. Larson, John N. Mordeson, Joshua D. Potter, Mark J. Wierman
Applying Fuzzy Mathematics to Formal Models in Comparative Politics, 2008
ISBN 978-3-540-77460-0

Vol. 226. Bhanu Prasad (Ed.)
Soft Computing Applications in Industry, 2008
ISBN 978-3-540-77464-8

Vol. 227. Eugene Roventa, Tiberiu Spircu
Management of Knowledge Imperfection in Building Intelligent Systems, 2008
ISBN 978-3-540-77462-4

Vol. 228. Adam Kasperski
Discrete Optimization with Interval Data, 2008
ISBN 978-3-540-78483-8

Vol. 229. Sadaaki Miyamoto, Hidetomo Ichihashi, Katsuhiro Honda
Algorithms for Fuzzy Clustering, 2008
ISBN 978-3-540-78736-5

Vol. 230. Bhanu Prasad (Ed.)
Soft Computing Applications in Business, 2008
ISBN 978-3-540-79004-4

Vol. 231. Michal Baczynski, Balasubramaniam Jayaram
Fuzzy Implications, 2008
ISBN 978-3-540-69080-1

Vol. 232. Eduardo Massad, Neli Regina Siqueira Ortega, Laécio Carvalho de Barros, Claudio José Struchiner
Fuzzy Logic in Action: Applications in Epidemiology and Beyond, 2008
ISBN 978-3-540-69092-4

Vol. 233. Cengiz Kahraman (Ed.)
Fuzzy Engineering Economics with Applications, 2008
ISBN 978-3-540-70809-4

Vol. 234. Eyal Kolman, Michael Margaliot
Knowledge-Based Neurocomputing: A Fuzzy Logic Approach, 2009
ISBN 978-3-540-88076-9

Vol. 235. Kofi Kissi Dompere
Fuzzy Rationality, 2009
ISBN 978-3-540-88082-0

Vol. 236. Kofi Kissi Dompere
Epistemic Foundations of Fuzziness, 2009
ISBN 978-3-540-88084-4

Vol. 237. Kofi Kissi Dompere
Fuzziness and Approximate Reasoning, 2009
ISBN 978-3-540-88086-8

Vol. 238. Atanu Sengupta, Tapan Kumar Pal
Fuzzy Preference Ordering of Interval Numbers in Decision Problems, 2009
ISBN 978-3-540-89914-3

Vol. 239. Baoding Liu
Theory and Practice of Uncertain Programming, 2009
ISBN 978-3-540-89483-4

Vol. 240. Asli Celikyilmaz, I. Burhan Türksen
Modeling Uncertainty with Fuzzy Logic, 2009
ISBN 978-3-540-89923-5

Vol. 241. Jacek Kluska
Analytical Methods in Fuzzy Modeling and Control, 2009
ISBN 978-3-540-89926-6

Jacek Kluska

Analytical Methods in Fuzzy Modeling and Control

Springer

Author

Jacek Kluska
Rzeszow University of Technology
Department of Computer
and Control Engineering
35-959 Rzeszow
W. Pola 2
Poland
E-mail: jacklu@prz.edu.pl

ISBN 978-3-540-89926-6 e-ISBN 978-3-540-89927-3

DOI 10.1007/978-3-540-89927-3

Studies in Fuzziness and Soft Computing ISSN 1434-9922

Library of Congress Control Number: 2008942025

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Foreword

The technologies of fuzzy systems and fuzzy controllers, in particular, have been applied with great success to numerous real world applications. The number of entries in the INSPEC database with keywords "fuzzy modeling" and "fuzzy control" dated 1969–2007 is 1,541 and 9,728, respectively. In spite of the evident progress reported in terms of concepts, algorithmic developments and engineering practice, there are still a number of challenging and highly relevant problems. Unfortunately, the existing publications are rather silent when it comes to reporting comprehensive solutions to them. The two challenges become particularly apparent and have been triggered by the growing complexity of the applications. The first evident challenge we are faced with is the curse of dimensionality. Rule–based systems and fuzzy rule–based systems are quite affected by this phenomenon especially when tackling problems of high dimensionality. The second one concerns a way of constructing fuzzy models which are accurate yet highly interpretable.

The author of the monograph has focused on these two vital problems and offered an interesting, original and practically relevant insight into their solutions. When dealing with fuzzy modeling, the book focuses on a broad class of Takagi–Sugeno–Kang (TS) fuzzy models – a highly legitimate choice given a wealth of literature on these constructs and a great deal of their applications. Furthermore the TS fuzzy models have been a subject of numerous analytical studies which have resulted in a series of interesting findings. This situation stands in a deep contrast with the most studies carried out in the realm of fuzzy control where analytical methods are not very common.

The analytical methods are beneficial to the better understanding of the advantages of the technology of fuzzy systems and its usage to the fullest extent when dealing with real–world problems. The book authored by Jacek Kluska is an important endeavor along this timely line of the development of fuzzy systems. While the author relates to an interesting treatise authored by Hao Ying (*Fuzzy control and modeling. Analytical foundations and applications.* IEEE Press, New York 2000), the book brings new and very much attractive ideas and presents important findings. The author not only re–visited and cast some Ying's results in an original fashion but further developed the Takagi–Sugeno fuzzy systems endowed with polynomial membership functions.

There are new notions and interesting results. The author introduced the notions of the generator and the fundamental matrix of the rule–based system and offered a convenient matrix description of the multiple–input multiple–output fuzzy system. Next, provided was a clear mathematical relation between the system of fuzzy rules and the systems described by "classical" differential or difference equations. The new and important are recurrence theorems dealing with rule–based systems with generalized classes of membership functions. It has been shown that those functions play an essential role in battling the ubiquitous curse of dimensionality.

Through a series of theorems the author established a one–to–one correspondence between the fuzzy systems and their classical counterparts and provided a detailed solution to many practical problems of substantial dimensionality.

Numerous examples covered in the text demonstrate the usefulness of the analytical methods of the fuzzy modeling in application to physical systems. The book builds a bridge between the highly interpretable fuzzy rule–based systems, classical control methods based on Boolean logic, multivalued logic and the conventional control theory, including its classic constructs of PID controllers.

Owing to the analytical approach the author developed an algebraic theory of rule–based systems, worked out an effective identification algorithms for a certain class of nonlinear dynamical systems, and proposed an interesting new classification system involving a collection of highly interpretable fuzzy rules.

A truly outstanding feature of this book is a mathematical rigor with which the author treats the subject matter and presents the reader with carefully structured ideas and algorithmic pursuits. All in all, the book can be highly recommended to researchers and practitioners interested in exploiting analytical methods of fuzzy modeling and control, system identification and diagnostics. Definitely this well–timed volume is a testimony to the rapid progress and a significant wealth of concepts and applications of Computational Intelligence.

Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta, Edmonton
and
Systems Research Institute,
Polish Academy of Sciences
Warsaw, Poland

Preface

This book does not contain an elementary mathematics of fuzzy systems such as fuzzy sets, operations on fuzzy sets, Boolean logic, triangular norms (t-norms), t-conorms, implications, fuzzy relations, fuzzy reasoning methods, the fuzzy controller architecture, the Mamdani type fuzzy controller, etc., because of the flood of papers and books on these topics. It is assumed that the reader is familiar with the fundamentals of the fuzzy modeling and with the foundations of Boolean logic and conventional control methods, including PID control.

This book is focused on mathematical analysis and rigorous design methods for fuzzy control systems based on Takagi-Sugeno fuzzy models, sometimes called Takagi-Sugeno-Kang models. We present a rather general analytical theory of exact fuzzy modeling and control of continuous and discrete-time dynamical systems. The main attention is paid to usability of the results for the control and computer engineering community and therefore simple and easy for linguistic interpretation knowledge-bases have been used. The approach is based on the author's theorems concerning equivalence between widely used Takagi-Sugeno systems and some class of multivariate polynomials. It combines the advantages of fuzzy system theory and classical control theory. Classical control theory can be applied to modeling of dynamical plants and the controllers. They are all equivalent to the set of Takagi-Sugeno type fuzzy rules. The approach combines the best of fuzzy and conventional control theory. It enables linguistic interpretability (also called transparency) of both the plant model and the controller. In the case of linear systems and some class of nonlinear systems, the engineer can in many cases directly apply well-known classical tools from the control theory both for analysis, and the design of the closed-loop fuzzy control systems.

The main objective of this book is to establish comprehensive and unified analytical foundations for fuzzy modeling using Takagi-Sugeno rule scheme and their applications for fuzzy control, identification of some class of nonlinear dynamical processes and classification problem solver design. After an excellent book of Ying [207], this is probably the second book which attempts to rigorously show that the fuzzy control is not a collection of applications without a solid theory. We pay special attention to the use of precise language to introduce the definitions and concepts, and to prove the conclusions.

Intended Readership of the Book

This self-contained textbook is intended for anyone who is interested in analytical aspects of fuzzy modeling and control applying the widely used Takagi-Sugeno rule scheme and wants to know precisely their connections with the classical counterparts. It is a self-study book for engineering professionals in diverse technical fields and industries, especially those in the fields of control and computer science. It aims at an audience of graduate and Ph.D. students as well. We assume that the reader has elementary background corresponding to an introductory course in automatic control, linear algebra and fundamentals of switching theory and logic design.

After reading Chapters 2 and 4, it can be studied in many ways, according to the particular interests of the reader. The book can be used together with the books on fundamentals of the control theory, artificial neural networks and other methods on machine learning. If a practicing engineer wants to apply the results of this book quickly, then the proofs of the lemmas and theorems may be skipped.

Originality of the Book

This book is focused on the rigorous mathematical methods of fuzzy modeling and control systems design based on the widely used Takagi-Sugeno rule scheme (TS for short), but it is not intended as a collection of existing results on fuzzy systems or fuzzy control. We present a new analytical theory of exact fuzzy modeling of continuous and discrete-time dynamical systems and logic systems which can be applied to solve the control, identification and classification problems encountered in practice. Therefore rather simple and highly interpretable knowledge-bases are used, putting a particular emphasis on the matrix calculus, symbolic calculus and recurrence. The approach is based on the author's theorems concerning equivalence between the Takagi-Sugeno systems and some class of multivariate polynomials, which combines the advantages of fuzzy system theory and classical control theory. Among others, it enables linguistic interpretability of the plant models and the controllers. Using the results developed in this book, the engineer can in many cases directly apply well-known tools from the conventional control theory (e.g. PID control) or binary logic design theory (e.g. combinational or sequential circuits), both to the analysis and design of the linear and some class of nonlinear closed-loop fuzzy control systems.

Several notions and results are new in this book, which are unavailable in any other book. To them belong the notions of the generator and the fundamental matrix of the TS system, the matrix description of the multiple-input-multiple-output (MIMO) TS system and new results on recurrence for the rule-based systems involving the first- and the second-order polynomials as the membership functions of fuzzy sets defined for the input variables. The book contains the proofs of the results in order to maintain a rigorous approach. Many examples included in the text illustrate usefulness of the analytical methods of the fuzzy modeling to many physical systems. The results obtained in this book are compared with other ones to show the advantages of the proposed procedures.

The material contained in this book is oriented towards the algorithms that are practically useful. We use analytical and systematic approach to the synthesis and analysis of the models. Thanks to this, a comparison of the methods developed in this book with the methods obtained by other authors is straightforward. Symbolic quantities are mainly used to ensure the generality of outcomes. Seldom, if ever, will numerical data be taken, to increase transparency of the examples. The book contains many examples concerning exact fuzzy modeling and control of real systems. We show theoretically and by examples that the fuzzy rule-based systems with the linear membership functions deserve a special attention not only from the theoretical point of view, but also they should be attractive for practitioners. The analytical results reinforce our belief that many successful applications of the fuzzy control cannot be a matter of chance.

Overview of the Book

The book consists of seven chapters. Chapter 2 provides the notion of the generator and the fundamental matrix of the rule-based system which are crucial for the book. One of the theorems establishes an exact relationship between the collection of fuzzy rules and a class of functions to which they are equivalent. It plays a crucial role in the modeling of many physical systems by using highly interpretable fuzzy rules. We show that the considered fuzzy rule-based system is nothing else but a part of the well-known Kolmogorov-Gabor polynomial. We prove that by formulating the consequents of the fuzzy rules which should express a given function, the only information needed by an expert is the values of this function in all vertices of the hypercuboid. In this chapter we introduce a compact matrix description of the set of fuzzy rules. The rule-based systems which use the linear membership functions of fuzzy sets for input variables are called the P1-TS systems. They are highly interpretable and therefore they are important from the engineering point of view. Finally, we consider an equivalence problem of the rule-bases in the context of the matarules taking into account that in reality the rule-bases can be noncomplete and/or contradictory ones.

Motivated by the "curse of dimensionality problem" of the fuzzy rule-based systems, Chapter 3 provides several results to make a calculation of the crisp system output feasible. We give some features of the fundamental matrix and its inverse. Thanks to one of theorems, the fundamental matrix inverse can be found recursively using multiplication operations only, instead of using classical inversion procedures. One of the main advantages of this chapter is providing a recursive procedure to solve the problem of "How to obtain the function performed by the rule-based system containing a large number of rules". To the best of the author's knowledge this problem has not been solved in the literature as yet. We show that thanks to the recursion, the curse of dimensionality problem can be substantially reduced. The computational architecture of the recursion can be viewed as a feedforward neural network. As an example of application of the recursion, the rule-based system with 6 inputs was considered. However, it is not a big problem to consider a P1-TS system with about 10 inputs. Next we show that the P1-TS systems can be used for the exact modeling of the nonlinear continuous or discrete-time dynamical systems, where the inputs of the fuzzy rule-based system are more abstract quantities and the outputs refer to the system structure. Such approach coincides in many respects with the one described in [184, where the system inputs can contain known premise variables that are not functions of the control input, but they may be functions of the state variables, external disturbances and/or time. For every input variable we assume two complementary membership functions that cannot be monotonic or linear. The advantages of our approach are exemplified. For the inverted pendulum system we obtain a better result than in other works. By using recurrence we can easily check validity of other models of nonlinear systems in the P1-TS form, e.g. a translational oscillator with an eccentric rotational proof mass actuator, a vehicle with triple trailers and many other dynamical systems discussed in the literature. In this chapter we show that application of the Taylor series expansion can be very attractive in practice. In one example we use 4th-degree Taylor polynomials for a good approximation of nonlinear functions at the equilibrium point of the dynamical system. The result is much better than the one obtained by the linearization of differential equations around the equilibrium. By using the Taylor series expansion we obtain a small number of highly interpretable fuzzy rules. Finally, we give the best evaluation for the lower and upper bound of the function, to which the rule-based P1-TS system is equivalent.

In order to obtain a richer class of functions to which the fuzzy rule-based system is equivalent, in Chapter 4 we use polynomials of the degree higher than one, as the membership functions of fuzzy sets. A special attention is paid to the TS systems which use the second degree polynomials. We show that it is not possible to obtain any second degree polynomial function, to which a TS rule-based system is equivalent, on the assumption that only two complementary membership functions as the second degree polynomials are defined for the input variables. However, three quadratic membership functions suffice to model every second degree polynomial function. For the zero-order TS system, we define for every input variable the set of three highly interpretable normalized membership functions as the second degree polynomials. The TS systems that use such fuzzy sets we call P2-TS systems. Such systems are thoroughly investigated. One of theorems says that the crisp output of the MISO P2-TS system in the vertex of the hypercuboid is exactly the same as the appropriate conclusion of the fuzzy rule contained in the rule-base. For the P2-TS systems both the generator and the fundamental matrix are defined. The fundamental matrix and its inverse are very important for the considered systems, since they enable one to establish an exact relationship between the consequents of the "If-then" rules and the parameters that define the crisp function, to which the rule-based system is equivalent. Therefore the procedures of how to compute the fundamental matrix and its inverse are given. The examples show that P2-TS systems have highly interpretable rule-bases when we use individual fuzzy rules or the metarules. The curse of dimensionality problem is much more serious for the P2-TS systems than the one for the P1-TS systems. Therefore, we develope the recursive procedures for the computation of both the inverse of the fundamental matrix and the crisp output of the P2-TS systems. The theorems say that we do not need to inverse large matrices to obtain the crisp output of the P2-TS systems. As a result of these theorems, the curse of dimensionality in P2-TS systems is substantially weakened. The results of this chapter can be easily generalized for the MIMO case. After this chapter we are able to thoroughly generalize the results for the TS systems with the membership functions that are polynomials of the degree $d \geq 3$. However, we should realize that the number of complete and noncontradictory rules will rapidly grow and the analysis will become more and more complicated. Both P1- and P2-TS systems are able to model a large class of real nonlinear processes. Therefore, if it is not necessary, we should not complicate our models in the engineering practice.

Chapter 5 mainly focuses on the P1-TS systems as the simplest and the most transparent among fuzzy rule-based systems with polynomial membership functions. In order to show that there are quite a lot of applications of P1-TS systems, many examples of exact modeling of conventional systems are given, especially in relation to nonlinear dynamical processes modeling and control. The P1-TS systems with two and more inputs are comprehensively investigated in the subsequent sections of Chapter 5, considering interpretability issue. It is exemplified that by using a multi-valued logic for highly nonlinear dynamical process, one can design an acceptable control algorithm expressed by the P1-TS system fuzzy rules. We show a connection between P1-TS systems and classical combinational logic systems. The fuzzy rule-based systems with inputs and outputs from the unity intervals are discussed in the context of generalized operators such as triangular norms, t-conorms, implications, etc. In this way, an unavoidable connection between fuzzy rule-based systems and Boolean algebra becomes apparent. We exemplify that the theory of P1-TS systems can be used to transform some control algorithms, formerly obtained with the use of Boolean logic, into the fuzzy domain. The highly interpretable rule-bases are constructed for the systems with three and more inputs not only for abstract processes, but also for real dynamical plants, e.g. a NARX model, fuzzy J-K flip-flop, Euler equations for a rigid body, Chen's attractor, the human immunodeficiency virus, magnetic suspension system, low order atmospheric circulation process and induction motor. The theory of P1-TS systems is also used for optimal analytical design of the well-known PID controller, working in the closed-loop control system for some class of the linear and nonlinear second order plants. Such a controller in the form of P1-TS system is optimal with respect to typical requirements for automatic control systems. After studying analytical results it is clear why the fuzzy PID controller as a P1-TS system can be better than the conventional PID algorithm. Next, we show that using our systematic approach, the so called "controller with variable gains" introduced by Ying [205], [206] can be easily obtained. In the last sections of Chapter 5 exact modeling of single input dynamical systems is investigated. Similarly as in the preceding sections we assume that nonlinear dynamical system is a collection of linear dynamical subprocesses. However, in contrast to the previous approach, where the inference was concerned with the structure parameters represented by matrices describing local linear models, the nonlinear model of the whole system is now inferred according to the original Takagi-Sugeno inference method. Based on this inference, we identify the class of dynamical systems to which the rule-based system is equivalent. Theoretical results are exemplified by exact fuzzy modeling of the van de Vusse reaction and Rössler chaotic system. Next we describe the architecture of the P1-TS system as the fuzzy model of conventional MIMO linear dynamical system. In Section 5.8 we show that the idea of TS systems with two linear membership functions of fuzzy sets can be easily extended to the systems with triangular fuzzy partition. The triangular membership functions can be substituted by other nonlinear membership functions which have the same support and the same monotonicity intervals. As a practical example of using the systems with triangular fuzzy partition, we present a sensor-based navigation system for a mobile robot. Chapter 5 ends with supplementary results for P1-TS systems. The outcomes concern the necessary and sufficient condition of linearity for such rule-based systems, the first-order P1-TS systems and the zero-order systems with contradictory rule-base. In the last section we show that the system without contradictions is a special case of the rule-based system with contradictions. For such systems we introduce a generalized fundamental matrix of the P1-TS, which can be easily extended to the P2-TS systems.

In Chapter 6 we investigate the identification problem of multilinear dynamical systems from observation data. Based on analytical results concerning exact fuzzy modeling of multilinear dynamical systems which provide necessary and sufficient conditions for transformation of fuzzy rules into crisp model, we prove the theorem on existence of the solution in the form of the P1-TS system. To get a solution we propose to use a batch procedure or recursive least squares method. The methodology preserves the interpretability of the fuzzy models, which is a key property of the considered rule-based systems and can be applied to continuous or discrete-time multilinear systems.

The proposed computation method can be viewed as a supervised learning algorithm for the adaptive linear neural network.

Chapter 7 provides a method for obtaining a set of highly interpretable "If-then" rules for the P1-TS system as an optimal (in the sense of a good generalization ability) binary classification problem solver. We use the results developed in the previous sections, especially related to modeling of the rule-based system from the input-output data, and the contradictory rules. The idea of constructing the classifier involves the theory of generators, fundamental matrices and support vector machines.

The bibliography at the end of the book lists the publications cited in the text as well as other relevant items that are not cited. Given a vast amount of papers and books, it is inevitable that the bibliography is still incomplete.

Of course, it is impossible to cover the entire spectrum of topic areas in one volume. A connection between the highly interpretable fuzzy "If-then" rules and some methods of artificial intelligence such as neural networks or kernel-based methods, was only signalled in this volume. Many of the results contained in this book establish a good starting point for stability and robustness analysis of fuzzy control systems and developing new learning and adaptation tools for intelligent control and diagnostic systems, which could be included in the future edition.

Jacek Kluska
Department of Computer and Control Engineering
Rzeszow University of Technology

Acknowledgements

It is my pleasant duty to thank my wife, Urszula, for her patience, understanding and support. My sincere thanks go to Professor Lotfi A. Zadeh of the University of California at Berkeley, the founder of fuzzy set theory and fuzzy logic, for his frequent words of personal encouragement, interest and inspiring advice which I always appreciate enormously.

I would like to express my most sincere gratitude to Professor Janusz Kacprzyk, the series editor, for his continuous encouragement, support, cooperation and giving me the opportunity to prepare the text.

I am deeply indebted to Professors Henryk Górecki from AGH University of Science and Technology in Krakow and Andrzej P. Wierzbicki from National Institute of Telecomunications in Warsaw, and to my Colleagues, Professors Leszek Trybus and Andrzej Sobkowiak from Rzeszow University of Technology, for their friendship, encouragement and understanding.

Jacek Kluska
Department of Computer and Control Engineering
Rzeszow University of Technology

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Systems can be represented by mathematical models of many different forms, such as algebraic equations, differential or integral equations, finite state machines, Petri nets, rules, etc. They are used particularly in the natural sciences and engineering disciplines such as physics, biology, electrical and computer engineering, in the social sciences such as economics or sociology. Engineers, computer scientists, physicists and economists use mathematical models most extensively. A mathematical model should be a representation of the essential aspects of an existing system (or a system to be constructed). This model should express the knowledge of that system in usable form [45].

Fuzzy systems theory enables us to utilize qualitative, linguistic information about a system to construct a mathematical model for it [132]. For many real-life systems, which are highly complex and inherently nonlinear, conventional approaches to modeling are not easy to apply, whereas the fuzzy approach might be a very helpful alternative. The modeling framework considered in this book is based on the models which describe relationships between variables by means of fuzzy "If-then" rules. Such models have one of two general structures: Mamdani or Takagi-Sugeno (TS). The difference between them is the construction of the rule consequents. In the former one, the consequents are linguistic (fuzzy sets), whereas the latter one employs crisp functions (or simply constants). Our considerations will be restricted to the Takagi-Sugeno models with the simplest fuzzy sets for the input variables.

Fuzzy models can be seen as rule-based systems suitable for formalizing the knowledge of experts. At the same time they are flexible mathematical structures which can represent complex nonlinear mappings. They integrate the logical processing of information with function approximation. Rule-based systems are not restricted to areas requiring human expertise and knowledge; they can be obtained from empirical data, as well. Methods for constructing fuzzy models from input-output data should not be limited to the best approximation of the data set only, but also and more importantly, to extract knowledge from training data in the form of the fuzzy rules. The rules should be easily understood and interpreted (see e.g. [12]). However, the

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 1–2.
springerlink.com

interpretability of fuzzy systems has not received much attention in the field of fuzzy modeling until now.

Fuzzy control is easy to learn and easy to apply, since it is close to human intuition. For this reason, it has been successfully applied to a variety of industrial processes and consumer products such as chemical reactors, cement kilns, vacuum cleaners, washing machines, autofocusing cameras, air conditioners, robots, voice-controlled robot helicopters, elevator systems and so on. However, we still need efficient analytical analysis and design methods to enable our deep understanding of fuzzy systems in the context of conventional modeling methods and control tools. Furthermore, we need systematic and unified approaches to design highly interpretable fuzzy models for the dynamical plants, the fuzzy controllers and other systems which are used in the engineering practice. Unfortunately, despite much research, such approaches seem to be only beginning to emerge. The main difficulty in the mathematical analysis of fuzzy models is that they are inherently nonlinear and, therefore, classical control theory with its emphasis on linear systems is difficult to apply or cannot be applied at all.

It should be added that the existing fuzzy models in the form of fuzzy "If-then" rules are not free from drawbacks. The curse of dimensionality problem of the rule-based systems is one of them. What is more, the fuzzy systems are mostly treated as magic black boxes with little analytical understanding and explanation [206]. Furthermore, there are no analytical results concerning quality of the closed-loop fuzzy control systems; practically all 'proofs' from the field of fuzzy control have been made by simulations, which is not always accepted by the scientific community. Finally, engineers need sufficiently clear and well justified methods for modeling and control which can be directly applied in practice. Such opinion and the above mentioned questions were the main motivation for writing this book.

Chapter 2
MISO Takagi-Sugeno Fuzzy System with Linear Membership Functions

Although we will be especially interested in Takagi-Sugeno models [180] called TS models for short which use linear or polynomial membership functions, we begin our considerations with the single-input and single-output system (SISO TS) which uses nonlinear membership functions. The problem involves determining the fuzzy rules which exactly model a nonlinear function belonging to some class of functions.

2.1 Perfect Approximation of Nonlinear Functions Using the Simplest Takagi-Sugeno Model

Below we will consider the problem of perfect approximation of nonlinear functions using the simplest Takagi-Sugeno model in the context of interpretability of fuzzy sets.

Suppose the input variable of a TS system is $z \in [-\alpha, \beta]$ and its output is S as shown in Fig. 2.1.

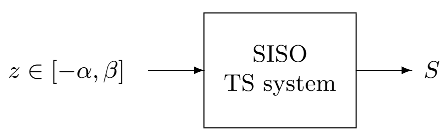


Fig. 2.1 Single-input-single-output TS system defined by the rules (2.1)

We assume that $\alpha + \beta \neq 0$. By N and P we denote two fuzzy sets which will be identified both with their linguistic labels and membership functions: $N(z)$ and $P(z)$, respectively. Thus, $N, P : [-\alpha, \beta] \to [0, 1]$. The TS system is defined by two fuzzy rules

$$\left.\begin{array}{l} R_1 : \text{If } z \text{ is } N, \text{ then } S = q_1, \\ R_2 : \text{If } z \text{ is } P, \text{ then } S = q_2. \end{array}\right\} \tag{2.1}$$

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 3–24.
springerlink.com

The natural requirements concerning the fuzzy sets are as follows

1. $N(z)$ is a continuous, nonincreasing function of z,
2. $N(-\alpha) = 1$ and $N(\beta) = 0$,
3. $P(z) = 1 - N(z)$.

Observe that P is a continuous, increasing function of z which satisfies boundary conditions: $P(\beta) = 1$ and $P(-\alpha) = 0$. Continuity, monotonicity and preservation of boundary conditions ensure a clear linguistic interpretation of both membership functions.

Suppose some continuous and monotonic function $f(z) : [-\alpha, \beta] \to \mathbb{R}$ is given. The problem is "How to obtain membership functions for the fuzzy rule-based TS system, such that its output is exactly the same, i.e. $S(z) = f(z)$ for any $z \in [-\alpha, \beta]$?" First of all the following conditions

$$q_1 = f(-\alpha), \qquad q_2 = f(\beta), \tag{2.2}$$

$$N(z) = \frac{f(z) - f(\beta)}{f(-\alpha) - f(\beta)}, \tag{2.3}$$

must be satisfied, since the output of the TS system is computed as follows [180]

$$S(z) = \frac{q_1 N(z) + q_2 P(z)}{N(z) + P(z)} = f(z), \qquad \text{for} \quad z \in [-\alpha, \beta]. \tag{2.4}$$

Example 2.1. The model (2.1) exactly approximates the following monotonic and continuous function (see Fig. 2.2)

$$f(z) = -\frac{\cos z}{z + \pi/4}, \qquad \text{for} \quad z \in \left[-\frac{\pi}{6}, \frac{\pi}{2}\right]. \tag{2.5}$$

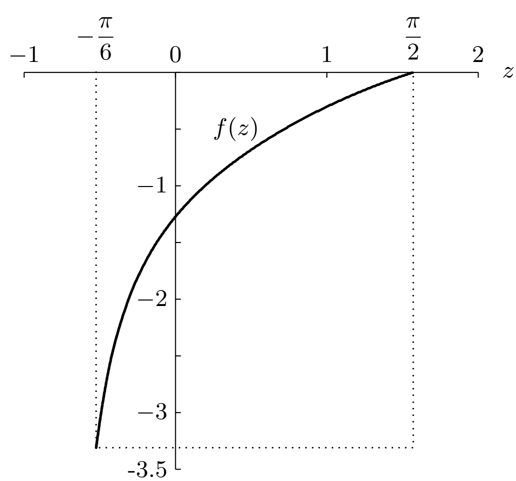


Fig. 2.2 Plot of the monotonic function (2.5) which can be exactly expressed by a TS system

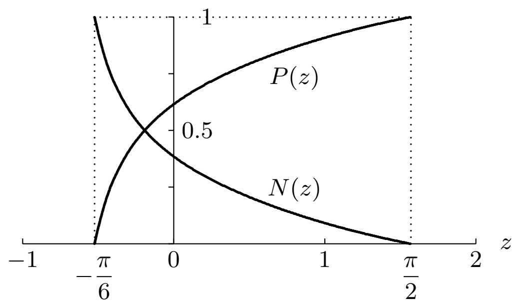


Fig. 2.3 Plot of the membership function $N(z)$ defined by (2.6) and its complement $P(z) = 1 - N(z)$

This is true if, and only if the membership function $N(z)$ from the class of functions defined above is given by the function shown in Fig. 2.3

$$N(z) = -\frac{\pi\sqrt{3}}{18} f(z), \qquad \text{for} \quad z \in \left[-\frac{\pi}{6}, \frac{\pi}{2}\right]. \tag{2.6}$$

Monotonicity of the membership functions of fuzzy sets is an important requirement. The question arises whether this requirement can be substituted by a local or global sector nonlinearity condition as suggested in [184] (p. 10)?

Example 2.2. Let us consider the function depicted in Fig. 2.4

$$f(z) = z(\sin z + 2), \qquad \text{for} \quad z \in [-1, 5]. \tag{2.7}$$

This function is a sector bounded nonlinearity. It is clear that for $z \in [-1, 5]$ the equation

$$N(z) = \frac{z(\sin z + 2) - 5\sin 5 - 10}{\sin 1 - 5\sin 5 - 12} \tag{2.8}$$

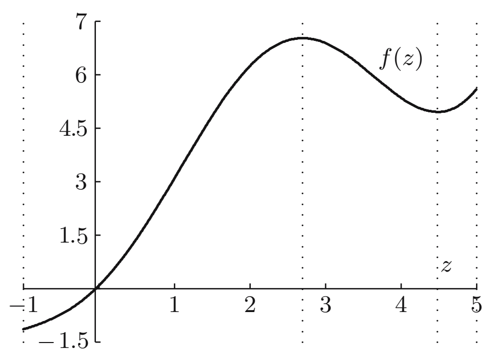


Fig. 2.4 Plot of a sector bounded function (2.7) which cannot be exactly expressed by a single TS system

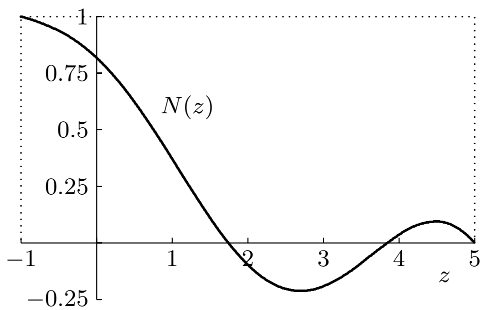


Fig. 2.5 Plot of the function (2.8)

must be satisfied, but the condition $N(z) \in [0,1]$ is not true for all $z \in [-1,5]$. Therefore $N(z)$ cannot be viewed as a membership function of some fuzzy set defined on the universe of discourse $[-1,5]$, (see Fig. 2.5). Of course, the function (2.7) can be exactly expressed in the form of three rule-based TS systems, where every system is designed in the monotonicity region of the original function $f(z)$ (see Fig.2.4).

Monotonicity of the membership functions of fuzzy sets is very important requirement from the interpretability point of view.

Example 2.3. For the continuous, smooth and highly nonlinear function

$$f(z) = e^{\pi} - (e^{\pi} - \pi^{e}) \sin^2(5\pi z/2) \exp\left(-\sin^2(9\pi z)\right), \quad z \in [0,1], \qquad (2.9)$$

one can find the fuzzy rules in the form of (2.1) and the fuzzy sets, such that $S(z) = f(z)$ for $z \in [0,1]$. The consequents of the fuzzy rules are constants $q_1 = e^{\pi}$, $q_2 = \pi^{e}$ and the membership functions of fuzzy sets N and P satisfy the boundary conditions ($P(0) = 0$, $P(1) = 1$, $N(0) = 1$ and $N(1) = 0$). The membership functions are as follows (see Fig. 2.6)

$$N(z) = 1 - P(z), \quad P(z) = \sin^2(5\pi z/2) \exp\left(-\sin^2(9\pi z)\right), \quad z \in [0,1]. \qquad (2.10)$$

Even though the output S of the TS system is exactly the same as the function (2.9) for all points from the universe of discourse and the membership functions satisfy the boundary conditions, the fuzzy sets are not easy for interpretation.

In the fuzzy modeling we should rather avoid nonmonotonic membership functions. Similar investigation to the one in the above section can be made for exact modeling of nonlinear systems with many input variables. Some ideas on this subject are included in [184], where however, there is no systematic procedure for converting a general nonlinear system to the TS form, even for nonlinear systems with nonlinearities that are polynomials of input variables [31].

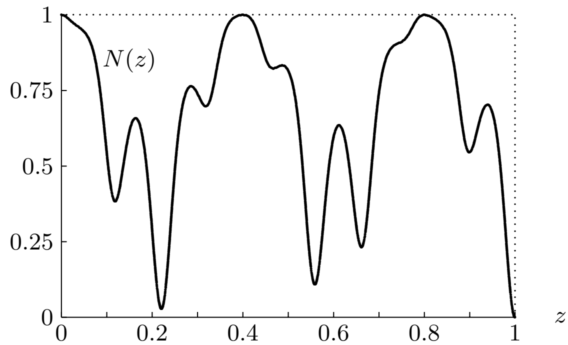


Fig. 2.6 Plot of the membership function $N(z)$ defined by (2.10)

2.2 Assumptions and Linguistic Interpretation of Linear Membership Functions

We will mainly use linear membership functions for input variables. They are conceptually the simplest, have a clear interpretation and play a crucial role in many applications in the fuzzy modeling and control. We will show further on mathematically and by examples that they are sufficient for modeling complex highly nonlinear static or dynamic, continuous or discrete-time systems.

Let us consider a multiple-input and single-output rule-based system (MISO system for short) with *input variables* $z_1, z_2, \ldots, z_n$. For every input $z_k \in [-\alpha_k, \beta_k]$ we require that there is no interval degenerated to a single point, i.e. we assume $\alpha_k + \beta_k \neq 0$ for $k = 1, 2, \ldots, n$, throughout the book. For any z_k, we define two fuzzy sets with *linear membership functions* $N_k(z_k)$, and $P_k(z_k)$, where P_k is an *algebraic complement* to N_k (see Fig. 2.7)

$$N_k(z_k) = \frac{\beta_k - z_k}{\alpha_k + \beta_k}, \tag{2.11}$$

$$P_k(z_k) = 1 - N_k(z_k), \qquad k = 1, 2, \ldots, n. \tag{2.12}$$

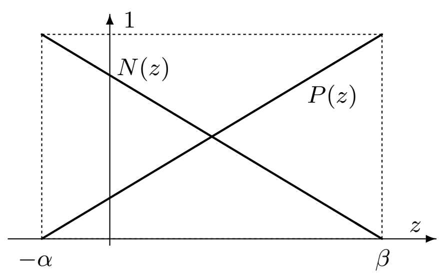


Fig. 2.7 Linear membership functions of two fuzzy sets

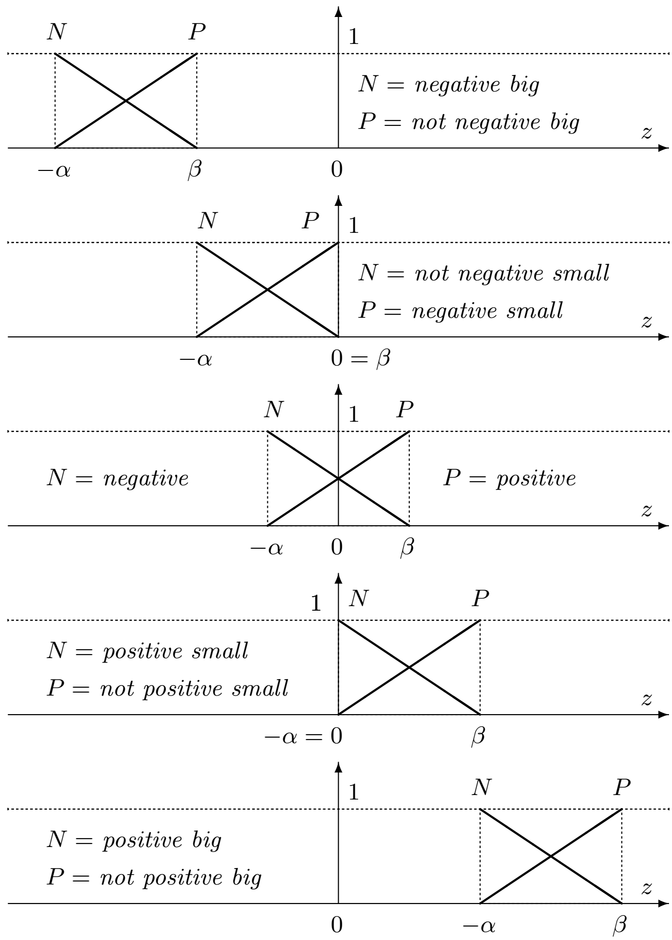


Fig. 2.8 Examples of linguistic interpretation of the fuzzy sets $N = N(z)$ and $P = P(z)$ for $z \in [-\alpha, \beta]$

It should be noted that using some linear transformation, the intervals $[-\alpha_k, \beta_k]$ could be replaced by different "standardized intervals". The unity interval $[0, 1]$ or symmetric around zero interval $[-1, 1]$ and many others belong to them. Such substitution would greatly simplify all mathematical descriptions and proofs. However, we will mainly use intervals $[-\alpha_k, \beta_k]$ further on, because for them it is possible to distinguish five cases, in which the terms N_k and P_k have different linguistic interpretations (see Fig. 2.8):

1. If $-\alpha_k < \beta_k < 0$, then N_k can be interpreted as *negative big*, and P_k - *not negative big*,

2. If $-\alpha_k < \beta_k = 0$, then N_k can be interpreted as *not negative small*, and P_k - *negative small*,
3. If $\alpha_k \approx \beta_k > 0$, then N_k can be interpreted as *negative*, and P_k - *positive*,
4. If $0 = -\alpha_k < \beta_k$, then N_k can be interpreted as *positive small*, and P_k - as *not positive small*,
5. If $0 < -\alpha_k < \beta_k$, then N_k can be interpreted as *positive big*, and P_k - as *not positive big*.

Obviously, depending on the context or specific application, the linguistic terms can be substituted by more suitable, adequate for the considered problem. For example the term *positive* can be replaced by *positive small* or *positive big*. We will use symbolic intervals $[-\alpha_k, \beta_k]$, where $-\alpha_k < \beta_k$. Thanks to this our analytical results will be more general than those obtained in other works, e.g. [168], [207].

Observe that for the functions (2.11)-(2.12) the inequalities

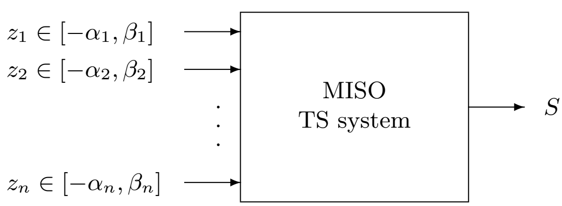
$$\frac{dN_k}{dz_k} < 0 \qquad \text{and} \qquad \frac{dP_k}{dz_k} > 0,$$

are satisfied, since $\alpha_k + \beta_k > 0$ for $k = 1, \ldots, n$. Therefore the symbol N_k refers to the membership function with *negative slope* and analogously P_k refers to the function with *positive slope*.

2.3 Compact Description of the MISO TS System

In order to allow the numbering of fuzzy rules by natural numbers, and to give more compact descriptions, we introduce a convenient indexing. Let us consider a MISO TS system with the inputs $z_1, \ldots, z_n$ and the output S (see Fig. 2.9). This system is defined by 2^n rules in the form of implications

Fig. 2.9 The inputs and the output of MISO TS system



$$\text{If } P_{(i_1,\ldots,i_n)}, \text{ then } S = q_{(i_1,\ldots,i_n)}, \tag{2.13}$$

where $(i_1, \ldots, i_n) \in \{0, 1\}^n$ and each *antecedent* $P_{(i_1,\ldots,i_n)}$ of an implication is the statement of the form

$$P_{(i_1,\ldots,i_n)} = \text{“}z_1 \text{ is } A_{i_1} \text{ and ... and } z_n \text{ is } A_{i_n}\text{”}, \tag{2.14}$$

and

$$A_{i_k} = \begin{cases} N_k, & \text{for} \quad i_k = 0 \\ P_k, & \text{for} \quad i_k = 1 \end{cases}, \quad k = 1, \ldots, n. \tag{2.15}$$

If it is not stated differently, we assume that the *consequents* $q_{(i_1,\ldots,i_n)}$ of the rules in (2.13) do not depend on the input variables, i.e. we will consider a *zero-order Takagi-Sugeno model* [180]. In more general TS systems, the consequents are polynomials of the first or higher order or more complicated functions of input variables.

The rule-based system (2.13)-(2.15) we will call *P1-TS system* to emphasize that membership functions of fuzzy sets for input variables are polynomials of the first order.

Now we introduce indexing which allows the ordering of the fuzzy rules. For any n-tuple of indices $(i_1, \ldots, i_n) \in \{0,1\}^n$ we define the corresponding index v, which is formally a function of the sequence of indices $(i_1, \ldots, i_n)$:

$$v = 1 + \sum_{k=1}^{n} i_k 2^{n-k}, \qquad i_k \in \{0,1\}, \qquad k = 1, \ldots, n. \tag{2.16}$$

Any v from the set $\{1, 2, \ldots, 2^n\}$ corresponds to only one antecedent of the fuzzy "If–then" rule. When the bijection (2.16) holds we will simply write $v \leftrightarrow (i_1, \ldots, i_n)$, e.g. $182 \leftrightarrow (1,0,1,1,0,1,0,1)$.

The rules (2.13) can be rewritten as

$$\text{If } P_v, \text{ then } S = q_v, \tag{2.17}$$

where $v \leftrightarrow (i_1, \ldots, i_n)$. For the inputs $z_1, \ldots, z_n$, the output is S and it is defined by the formula [180]

$$S(z_1, \ldots, z_n) = \frac{\sum_{v=1}^{2^n} q_v h_v(z_1, \ldots, z_n)}{\sum_{v=1}^{2^n} h_v(z_1, \ldots, z_n)}, \tag{2.18}$$

where

$$h_v(z_1, \ldots, z_n) = \top \left(A_{i_1}(z_1), \ldots, A_{i_n}(z_n)\right)_v, \tag{2.19}$$

the operator $\top$ denotes an algebraic t-norm: $\top(x, y) = xy$ [202], the indices v and $(i_1, \ldots, i_n)$ are in the one-to-one correspondence (2.16), and $A_{i_k}(z)$ are membership functions of the fuzzy sets, i.e. $A_{i_k} \in \{N_k, P_k\}$ for $i_k \in \{0,1\}$ and $k = 1, \ldots, n$. The value h_v can be interpreted as a *degree of fulfilment* (or *degree of firing level*) of the vth rule by the given inputs $z_1, \ldots, z_n$. One can check that

$$\sum_{v=1}^{2^n} h_v(z_1, \ldots, z_n) = \prod_{i=1}^{n} \left(N_i(z_i) + P_i(z_i)\right), \tag{2.20}$$

and therefore, if the complementary property (2.12) is satisfied, then (2.18) reduces to

$$S = \sum_{v=1}^{2^n} q_v h_v(z_1, \ldots, z_n). \tag{2.21}$$

The function $h_v(z_1, \ldots, z_n)$ can be viewed as a *normalized* membership function of many variables or as a fuzzy relation.

The set

$$D^n = [-\alpha_1, \beta_1] \times \ldots \times [-\alpha_n, \beta_n], \tag{2.22}$$

where $\times$ denotes the Cartesian product, we will call a *hypercuboid*. Its vertices are the vectors

$$\boldsymbol{\gamma}_v = [\gamma_1, \ldots, \gamma_n]^T \in \{-\alpha_1, \beta_1\} \times \ldots \times \{-\alpha_n, \beta_n\}, \tag{2.23}$$

where $v \leftrightarrow (i_1, \ldots, i_n) \in \{0,1\}^n$, and they can be ordered according to (2.16) as shown in Fig. 2.10. The length L_k of the interval $[-\alpha_k, \beta_k]$ and the volume V_k of the hypercuboid D^k are defined by

$$L_k = \alpha_k + \beta_k \ , \qquad k = 1, 2, ..., n, \tag{2.24}$$

$$V_k = \prod_{i=1}^{k} L_i \ , \qquad k = 1, 2, ..., n. \tag{2.25}$$

They will be helpful in the future for the interpretation of some results.

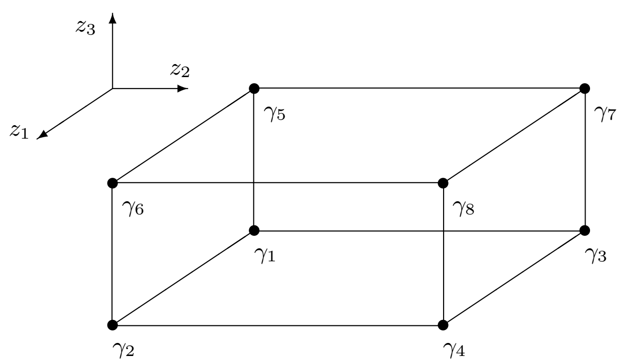


Fig. 2.10 Vertices of the hypercuboid D^n for $n = 3$

2.4 Crisp Output of the Zero-Order MISO P1-TS System

In this section we prove the main theorem concerning modeling of systems using the Takagi-Sugeno rule scheme, which uses two complementary linear membership functions for each input variable.

Theorem 2.4. *Define for the vector variable* $\mathbf{z} = [z_1, \ldots, z_n]^T$, *the following multilinear function* $f_0 : D^n \to \mathbb{R}$,

$$f_0(\mathbf{z}) = \sum_{(p_1, p_2, \ldots, p_n) \in \{0,1\}^n} \theta_{p_1, p_2, \ldots, p_n} z_1^{p_1} z_2^{p_2} \cdots z_n^{p_n} \ , \tag{2.26}$$

where 2^n coefficients $\theta_{00\ldots0}$, $\theta_{10\ldots0}$, $\theta_{01\ldots0}$, ..., $\theta_{11\ldots1}$, are real numbers. For every function of the type (2.26) there exists a zero-order MISO P1-TS system such that $S(\mathbf{z}) = f_0(\mathbf{z})$ for all $\mathbf{z} \in D^n$ and

(i) the inputs of the system are components of $\mathbf{z} \in D^n$ and the output is S (see Fig. 2.9),
(ii) two linear membership functions defined by (2.11)-(2.12) are assigned to each component of the vector $\mathbf{z}$,
(iii) the system is defined by 2^n fuzzy rules in the form of (2.13)-(2.15).

One can find all consequents q_1, q_2, ..., q_{2^n} of the fuzzy rules by solving 2^n linear equations. For a nonzero volume of the hypercuboid D^n, the unique solution always exists.

Proof. First we identify the class of functions performed by the TS system. The t-norm in (2.18) is an algebraic product and any function h_v in (2.19) is a product of the first order polynomials. Thus,

$$S = \sum_{i_n=1}^{2} \cdots \sum_{i_1=1}^{2} \prod_{k=1}^{n} (a_{i_k} z_k + b_{i_k})\, q_{(i_1,\ldots,i_n)} \ ,$$

where a_{i_k}, b_{i_k}, and $q_{(i_1,\ldots,i_n)}$ are real numbers. This means that $S(\mathbf{z})$ is a multilinear function which can be written in the form of (2.26).

Now assume that some function f_0 in the form of (2.26) is given. Our goal is to express all consequents of the rules for the fixed in advance collection of coefficients $\theta_{00\ldots0}$, $\theta_{10\ldots0}$, $\theta_{01\ldots0}$, ..., $\theta_{11\ldots1}$. The function f_0 is the scalar product $f_0(\mathbf{z}) = \boldsymbol{\theta}^T \mathbf{g}(\mathbf{z})$, where

$$\boldsymbol{\theta} = [\theta_{00\ldots0}, \theta_{10\ldots0}, \theta_{01\ldots0}, ..., \theta_{p_1\ldots p_n}, ..., \theta_{11\ldots1}]^T \in \mathbb{R}^{2^n}, \qquad (2.27)$$

$$\mathbf{g}(\mathbf{z}) = [1, \ldots, (z_1^{p_1} \cdots z_n^{p_n}), \ldots, (z_1 \cdots z_n)]^T, \qquad (2.28)$$

with $p_k \in \{0,1\}$ for $k = 1, \ldots, n$. For a given $\mathbf{z}$, the vector $\mathbf{g}(\mathbf{z})$ we will call a *generator*. It is continuous nonlinear mapping, which transforms the points $\mathbf{z} \in D^n$ into 2^n-dimensional space, whereas the function f_0 is a *linear function with respect to parameters* $\theta_{00\ldots0}$, $\theta_{10\ldots0}$, $\theta_{01\ldots0}$, $\ldots$, $\theta_{p_1,p_2,\ldots,p_n}$, $\ldots$, $\theta_{11\ldots1}$. The generator $\mathbf{g} = \mathbf{g}(z_1, z_2, \ldots, z_n)$ contains 2^n components of the form "$z_{i_1} z_{i_2} \ldots z_{i_k}$" being elements of the polynomial "$(1+z_1) \times \cdots \times (1+z_n)$" written in the expanded additive form, when substituting in the *monomials* of this polynomial all coefficients by "1".

The equation $f_0(z_1, \ldots, z_n) = \boldsymbol{\theta}^T \mathbf{g}(z_1, \ldots, z_n)$ must be satisfied for all points in the hypercuboid D^n, especially in its vertices. Thus, the following 2^n linear equations

$$\boldsymbol{\theta}^T \mathbf{g}(\boldsymbol{\gamma}_v) = q_v, \qquad v = 1, 2, \ldots, 2^n, \qquad (2.29)$$

must be satisfied, or equivalently

$$\mathbf{q} = \boldsymbol{\Omega}^T \boldsymbol{\theta}, \quad \boldsymbol{\Omega} = \left[\mathbf{g}(\boldsymbol{\gamma}_1), \ldots, \mathbf{g}(\boldsymbol{\gamma}_{2^n}) \right]_{2^n \times 2^n}, \tag{2.30}$$

where the consequents of the rules constitute the vector $\mathbf{q} = [q_1, \ldots, q_{2^n}]^T$. Thus,

$$\mathbf{q} = \begin{bmatrix} \boldsymbol{\theta}^T \mathbf{g}(\boldsymbol{\gamma}_1) \\ \vdots \\ \boldsymbol{\theta}^T \mathbf{g}(\boldsymbol{\gamma}_{2^n}) \end{bmatrix} = \begin{bmatrix} f_0(\boldsymbol{\gamma}_1) \\ \vdots \\ f_0(\boldsymbol{\gamma}_{2^n}) \end{bmatrix}. \tag{2.31}$$

The equations (2.30)-(2.31) formulate necessary conditions, under which the system of fuzzy rules is equivalent to (2.26). Now we prove that they are sufficient as well. Sufficiency requires that the $2^n \times 2^n$ matrix $\boldsymbol{\Omega}$ containing the columns $\mathbf{g}(-\alpha_1, -\alpha_2, \ldots, -\alpha_n)$, ..., $\mathbf{g}(\beta_1, \beta_2, \ldots, \beta_n)$ is a nonsingular one. Observe that the output S of the TS system with n inputs z_1, z_2, ..., z_n can be defined by the following rule base, which is equivalent to (2.13)-(2.15):

$$\left.\begin{array}{l} R_1: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_1, \\ R_2: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_2, \\ R_3: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_3, \\ R_4: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_4, \\ \qquad\qquad\qquad\vdots \\ R_{2^n}: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{2^n}. \end{array}\right\} \tag{2.32}$$

The matrix $\boldsymbol{\Omega}$ we will call the *fundamental matrix* throughout the book. It contains as elements the values of the generator $\mathbf{g}$ in such vertices of the hypercuboid D^n that exactly correspond to the labels used in the antecedents of the rules. This means that "$-\alpha_k$" in $\mathbf{g}$ corresponds to N_k, and "β_k" in $\mathbf{g}$ corresponds to P_k, where N_k and/or P_k are used in the antecedent of the rule. The vector $\mathbf{q} = [q_1, q_2, \ldots, q_{2^n}]^T$ contains successive consequents of the rules. Thus, both the order of vertices $\boldsymbol{\gamma}_v$ used for computing $\boldsymbol{\Omega}$, and the order of elements of $\mathbf{q}$ are strictly defined. Now we prove inductively that $\boldsymbol{\Omega}$ is nonsingular if, and only if $\alpha_k + \beta_k \neq 0$ for $k = 1, 2, \ldots, n$. In the case of n input variables, the generator $\mathbf{g}$ and the matrix $\boldsymbol{\Omega}$ will have a subscript, i.e. $\mathbf{g} = \mathbf{g}_n$ and $\boldsymbol{\Omega} = \boldsymbol{\Omega}_n$. Formally, we define an artificial generator $\mathbf{g}_0 = 1$ and the corresponding artificial matrix $\boldsymbol{\Omega}_0 = 1$. First, consider the case with $n = 1$. The rule base structure is as follows

$$\left.\begin{array}{l} R_1: \text{If } z_1 \text{ is } N_1, \text{ then } S = q_1, \\ R_2: \text{If } z_1 \text{ is } P_1, \text{ then } S = q_2, \end{array}\right\} \tag{2.33}$$

and the corresponding generator $\mathbf{g}_1 = \mathbf{g}_1(z_1)$ is given by

$$\mathbf{g}_1 = \begin{bmatrix} \mathbf{g}_0 \\ z_1 \mathbf{g}_0 \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \end{bmatrix}. \tag{2.34}$$

The fundamental matrix $\mathbf{\Omega}_1$ is a concatenation of 2 columns $\mathbf{g}_1$. It is generated as follows

$$\mathbf{\Omega}_1 = [\mathbf{g}_1(-\alpha_1), \mathbf{g}_1(\beta_1)] = \begin{bmatrix} 1 & 1 \\ -\alpha_1 & \beta_1 \end{bmatrix}. \tag{2.35}$$

For $n = 2$ the rule-base structure is the following

$$\left.\begin{aligned} R_1 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_1, \\ R_2 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_2, \\ R_3 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_3, \\ R_4 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_4, \end{aligned}\right\} \tag{2.36}$$

and the corresponding generator $\mathbf{g}_2 = \mathbf{g}_2(z_1, z_2)$ is given by

$$\mathbf{g}_2 = \begin{bmatrix} \mathbf{g}_1 \\ z_2\mathbf{g}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \end{bmatrix}. \tag{2.37}$$

The fundamental matrix $\mathbf{\Omega}_2$ is a concatenation of 4 columns

$$\begin{aligned} \mathbf{\Omega}_2 &= [\mathbf{g}_2(-\alpha_1, -\alpha_2), \mathbf{g}_2(\beta_1, -\alpha_2), \mathbf{g}_2(-\alpha_1, \beta_2), \mathbf{g}_2(\beta_1, \beta_2)] \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \end{bmatrix}. \end{aligned} \tag{2.38}$$

For $n = 3$ the rule base structure is

$$\left.\begin{aligned} R_1 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_3 \text{ is } N_3, \text{ then } S = q_1, \\ R_2 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_3 \text{ is } N_3, \text{ then } S = q_2, \\ R_3 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2 \text{ and } z_3 \text{ is } N_3, \text{ then } S = q_3, \\ R_4 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } z_3 \text{ is } N_3, \text{ then } S = q_4, \\ R_5 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_3 \text{ is } P_3, \text{ then } S = q_5, \\ R_6 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_3 \text{ is } P_3, \text{ then } S = q_6, \\ R_7 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2 \text{ and } z_3 \text{ is } P_3, \text{ then } S = q_7, \\ R_8 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } z_3 \text{ is } P_3, \text{ then } S = q_8, \end{aligned}\right\} \tag{2.39}$$

and the corresponding generator $\mathbf{g}_3 = \mathbf{g}_3(z_1, z_2, z_3)$ is given by

$$\mathbf{g}_3(z_1, z_2, z_3) = \begin{bmatrix} \mathbf{g}_2 \\ z_3\mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \\ z_3 \\ z_1 z_3 \\ z_2 z_3 \\ z_1 z_2 z_3 \end{bmatrix}. \tag{2.40}$$

The fundamental matrix $\mathbf{\Omega}_3$ is a concatenation of $2^3 = 8$ columns $\mathbf{g}_3$ and is generated as

$$\begin{aligned}\mathbf{\Omega}_3 = [&\mathbf{g}_3(-\alpha_1, -\alpha_2, -\alpha_3),\ \mathbf{g}_3(\beta_1, -\alpha_2, -\alpha_3),\ \mathbf{g}_3(-\alpha_1, \beta_2, -\alpha_3),\\ &\mathbf{g}_3(\beta_1, \beta_2, -\alpha_3),\ \mathbf{g}_3(-\alpha_1, -\alpha_2, \beta_3),\ \mathbf{g}_3(\beta_1, -\alpha_2, \beta_3),\\ &\mathbf{g}_3(-\alpha_1, \beta_2, \beta_3),\ \mathbf{g}_3(\beta_1, \beta_2, \beta_3)]\\ = &\begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & -\alpha_3 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & -\alpha_3 & -\beta_1\alpha_3 & \alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & -\alpha_3 & \alpha_1\alpha_3 & -\alpha_3\beta_2 & \alpha_1\alpha_3\beta_2 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & -\alpha_3 & -\beta_1\alpha_3 & -\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & \beta_3 & -\alpha_1\beta_3 & -\alpha_2\beta_3 & \alpha_1\alpha_2\beta_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & \beta_3 & \beta_1\beta_3 & -\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & \beta_3 & -\alpha_1\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_2\beta_3 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & \beta_3 & \beta_1\beta_3 & \beta_2\beta_3 & \beta_1\beta_2\beta_3 \end{bmatrix}^T,\end{aligned} \tag{2.41}$$

and so forth. In general,

$$\begin{aligned}\mathbf{g}_0 &= 1, \qquad \mathbf{\Omega}_0 = 1,\\ \mathbf{g}_{k+1} &= \begin{bmatrix} \mathbf{g}_k \\ z_{k+1}\mathbf{g}_k \end{bmatrix} = \begin{bmatrix} 1 \\ z_{k+1} \end{bmatrix} \otimes \mathbf{g}_k \in \mathbb{R}^{2^{k+1}}, \qquad k = 0, 1, 2, \ldots, n-1,\end{aligned} \tag{2.42}$$

where the symbol "$\otimes$" denotes the Kronecker product (see Appendix A or [43], [83]). One can easily check that

$$\mathbf{\Omega}_{k+1} = \begin{bmatrix} \mathbf{\Omega}_k & \mathbf{\Omega}_k \\ -\alpha_{k+1}\mathbf{\Omega}_k & \beta_{k+1}\mathbf{\Omega}_k \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix} \otimes \mathbf{\Omega}_k \in \mathbb{R}^{2^{k+1} \times 2^{k+1}}, \tag{2.43}$$

for $k = 0, 1, 2, \ldots, n-1$. From (A.6a) given in Appendix A we immediately obtain

$$\det \mathbf{\Omega}_{k+1} = (\beta_{k+1} + \alpha_{k+1})^{2^k} (\det \mathbf{\Omega}_k)^2 .$$

Taking into account $\mathbf{\Omega}_0 = 1$ already defined, we obtain

$$\det \mathbf{\Omega}_n = \prod_{i=1}^{n} (\beta_i + \alpha_i)^{2^{n-1}} = (V_n)^{2^{n-1}} .$$

Thus, $\det \mathbf{\Omega}_n \neq 0$ if, and only if the volume of the heperrectangle D^n in the space $\mathbb{R}^n$ is not zero or, equivalently, for every input variable z_k, $(k = 1, 2, \ldots, n)$, the interval $[-\alpha_k, \beta_k]$ is not degenerated to a single point. Finally, from (2.30) we obtain the vector of coefficients of the function (2.26)

$$\boldsymbol{\theta} = \left(\mathbf{\Omega}^T\right)^{-1} \mathbf{q}. \tag{2.44}$$

Thus, the crisp output of the P1-TS system is given by

$$S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\mathbf{\Omega}^T\right)^{-1} \mathbf{q} = f_0(\mathbf{z}). \tag{2.45}$$

This ends the proof of Theorem 2.4. □

Remark 2.5. The components of the vector $\boldsymbol{\theta}$ in (2.26) depend on $2^n + 2n$ parameters, i.e. on 2^n coefficients q_v and $2n$ boundaries of intervals $[-\alpha_k, \beta_k]$, $(k = 1, 2, \ldots, n)$.

Remark 2.6. Suppose the hypercuboid $D^n \subset \mathbb{R}^n$ is established. The function

$$f_1(\mathbf{z}) = \prod_{k=1}^{n} (r_k z_k + s_k), \tag{2.46}$$

where r_k, s_k are real numbers, is a special case of the function (2.26) for $n > 2$, since it contains $2n$ parameters r_k and s_k, whereas the function (2.26) contains 2^n coefficients $\theta_{00\ldots0}$, $\theta_{10\ldots0}$, $\theta_{01\ldots0}$, ..., $\theta_{11\ldots1}$.

As a conclusion of Theorem 2.4, whose interpretation is important, we obtain

Corollary 2.7. *Suppose a function $f : D^n \to \mathbb{R}$ is known and it belongs to the class of functions (2.26). In other words $f(\mathbf{z}) = f_0(\mathbf{z})$, where $\mathbf{z} \in D^n$ for some collection of coefficients of the vector $\boldsymbol{\theta}$ as in (2.27). A necessary and sufficient condition under which the considered TS system is equivalent to $f(\mathbf{z})$ for any $\mathbf{z} \in D^n$, is as follows*

$$q_v = f(\boldsymbol{\gamma}_v), \quad \textit{for} \quad v = 1, 2, \ldots, 2^n. \tag{2.47}$$

This means that by formulating the consequents of the fuzzy rules, the only information needed by an expert are values of the function f in all vertices of the hypercuboid D^n.

What is more, Theorem 2.4 says that we can always obtain an equivalent TS system to the given function (2.26).

2.5 Completeness and Noncontradiction in Rule-Based Systems Defined by Metarules

The rule-base is usually assumed to have the form of (2.32). Such system contains *complete* and *noncontradictory* rules [92]. The system of "If-then" fuzzy rules will be called *complete* if every rule contains all possible antecedents in its “If” part, which results in 2^n rules as in (2.13)-(2.15). The system of rules is a *contradictory* one if there are at least two rules which have the same antecedent but different consequents. By such definitions, the system (2.32) is both complete and noncontradictory. The same notions can be defined in the fuzzy sense, i.e. the rules can be viewed as complete or noncontradictory to some degree. However, we will consider them as bivalent notions, i.e. the systems of rules will be treated as complete (contradictory) or not, throughout the book.

When the number of inputs is large, we can use the *metarules*, i.e. the rules which are equivalent to some subset of the rules, where each single rule is in the form of (2.13)-(2.15). Most frequently we have to do with the metarule if some “If-then” rule in its “If” part contains the word ANY_k (or ANY without a subscript). By the term ANY_k we mean any label from the bivalent set $\{N_k, P_k\}$, $(k = 1, \ldots, n)$. Sometimes the set of the rules may be generated by a metarule for other reasons.

Remark 2.8. The fragments “z_k is ANY” in the antecedents of the rules will be sometimes omitted. For example, we can simplify the fuzzy rule “If z_1 is N_1 and z_2 is ANY_2, then $S = q_1$” into the shorter one “If z_1 is N_1, then $S = q_1$”.

Example 2.9. Let us consider three P1-TS systems with two inputs and one output, which are equivalently presented in Tables 2.1 a) – c).

a) The system of rules is defined by (2.36) and shown in Table 2.1 a). It is complete and noncontradictory. This case is simple and does not need a comment.
b) The system of rules:

$$\left.\begin{array}{l} R_1: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_1, \\ R_2: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_2, \\ R_4: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_4, \end{array}\right\}$$

is shown in Table 2.1 b). Observe that there is no consequent for the antecedent “z_1 is N_1 and z_2 is P_2”. This system is noncomplete and noncontradictory.

Table 2.1 Look-up-tables for the P1-TS system from Example 2.9: a) Complete and noncontradictory rules, b) Noncomplete and noncontradictory rules, c) Noncomplete and contradictory rules

a)

z_1, $z_2 \rightarrow$ ↓	N_2	P_2
N_1	q_1	q_3
P_1	q_2	q_4

b)

z_1, $z_2 \rightarrow$ ↓	N_2	P_2
N_1	q_1	–
P_1	q_2	q_4

c)

z_1, $z_2 \rightarrow$ ↓	N_2	P_2
N_1	q_1	q_1, q_3
P_1	q_2	–

c) The system of rules:

$$\left.\begin{array}{l} R_1 : \text{If } z_1 \text{ is } N_1, \text{ then } S = q_1, \\ R_2 : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_2, \\ R_3 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_3, \end{array}\right\}$$

is equivalent to (see Remark 2.8)

$$\left.\begin{array}{l} R_1' : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_1, \\ R_1'' : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_1, \\ R_2 : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_2, \\ R_3 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_3. \end{array}\right\}$$

Although there are four single rules, there is no consequent for the antecedent "z_1 is P_1 and z_2 is P_2". For $q_1 \neq q_3$, the metarule R_1'' contradicts (more or less) the rule R_3. Thus, this system is both contradictory and noncomplete.

The above example shows that detection of completeness or noncontradiction in the rule-base is a very simple task if we use the look-up tables.

2.6 Matrix Description of the MIMO Fuzzy Rule-Based System

In this section we generalize the concept of MISO fuzzy rule-based systems into the multiple-input and multiple-output (MIMO) systems. In the systems with many outputs there are no cross-feedback loops. Therefore the procedure of computing a single output is the same as for the MISO TS systems.

Our goal in this section is to develop yet another compact and convenient description of the rule-based system, i.e. the *model in the matrix form*. Let us consider a TS system with the inputs $z_1, \ldots, z_n$ and the outputs $S_1, \ldots, S_m$, as shown in Fig. 2.11. By a MIMO P1-TS system we mean the system with $m \geq 2$ outputs, in which the membership functions of fuzzy sets for all inputs are linear as defined in (2.11)-(2.12). Such a system is described by the following 2^n fuzzy rules:

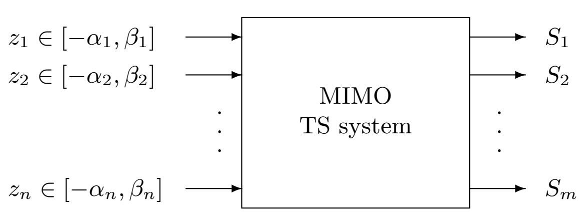


Fig. 2.11 The inputs and the outputs of a MIMO TS system given in the matrix form (2.48)-(2.53)

R_1 : If z_1 is N_1 and z_2 is N_2 and ... and z_n is N_n,
then $S_1 = q_{1,1}$, ..., $S_m = q_{1,m}$,

$\vdots$

R_v : If z_1 is A_{i_1} and z_2 is A_{i_2} and ... and z_n is A_{i_n},
then $S_1 = q_{v,1}$, ..., $S_m = q_{v,m}$,

$\vdots$

R_{2^n} : If z_1 is P_1 and z_2 is P_2 and ... and z_n is P_n,
then $S_1 = q_{2^n,1}$, ..., $S_m = q_{2^n,m}$,

where $A_{i_k} \in \{N_k, P_k\}$, ($k = 1, 2, \ldots, n$ and $i_k \in \{0, 1\}$), as defined in (2.15). Equivalently, this system can be described by the following single *"If-then" rule in the matrix form*

$$\text{If } [z_1, \ldots, z_n] \text{ is } \mathbf{M}, \text{ then } [S_1, \ldots, S_m] \text{ is } \mathbf{Q}, \tag{2.48}$$

where we assume that

- the *antecedents matrix* $\mathbf{M}$ contains the labels of fuzzy sets and has 2^n rows and n columns
$$\mathbf{M} = \begin{bmatrix} N_1 & \cdots & N_{n-1} & N_n \\ \vdots & \ddots & \vdots & \vdots \\ A_{i_1} & \cdots & A_{i_{n-1}} & A_{i_n} \\ \vdots & \ddots & \vdots & \vdots \\ P_1 & \cdots & P_{n-1} & P_n \end{bmatrix}, \tag{2.49}$$
where $\left(A_{i_1}, \ldots, A_{i_{n-1}}, A_{i_n}\right) \in \{N_1, P_1\} \times \ldots \times \{N_{n-1}, P_{n-1}\} \times \{N_n, P_n\}$,
- the *consequents matrix* $\mathbf{Q}$ contains m columns, and every column $\mathbf{q}_j$ corresponds to the output S_j of the rule-based system

$$\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_j, \cdots, \mathbf{q}_m] = \begin{bmatrix} q_{1,1} & \cdots & q_{1,j} & \cdots & q_{1,m} \\ q_{2,1} & \cdots & q_{2,j} & \cdots & q_{2,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{2^n,1} & \cdots & q_{2^n,j} & \cdots & q_{2^n,m} \end{bmatrix} \in \mathbb{R}^{2^n \times m}. \tag{2.50}$$

For such systems we formulate the following

Theorem 2.10. *Suppose the MIMO P1-TS system with the inputs constituting the vector* $[z_1, \ldots, z_n]^T = \mathbf{z} \in D^n$ *and the outputs* $S_1, \ldots, S_m$*, is defined by* 2^n *fuzzy "If-then" rules or equivalently – by a single rule in the matrix form (2.48)-(2.50). The row vector of crisp outputs* $\mathbf{S}(\mathbf{z}) = [S_1, \ldots, S_m]$ *can be computed by the formula*

$$\mathbf{S}(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\mathbf{\Omega}^T\right)^{-1} \mathbf{Q}, \tag{2.51}$$

where the consequents matrix

$$\mathbf{Q} = \mathbf{\Omega}^T \mathbf{\Theta}, \tag{2.52}$$

and

$$\begin{aligned} &\mathbf{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m] \in \mathbb{R}^{2^n \times m}, \\ &\boldsymbol{\theta}_j = [\theta_{j,00\ldots0}, \theta_{j,10\ldots0}, \theta_{j,01\ldots0}, ..., \theta_{j,11\ldots1}]^T \in \mathbb{R}^{2^n}, \qquad j = 1, \ldots, m. \end{aligned} \tag{2.53}$$

Every column $\boldsymbol{\theta}_j$ is assigned to a single system output S_j, ($j = 1, \ldots, m$). The successive components of the generator $\mathbf{g}(\mathbf{z})$ are in accordance with the rows of the antecedents matrix $\mathbf{M}$ defined by (2.49). The fundamental matrix $\mathbf{\Omega}$ of the system is a concatenation of 2^n columns which are values of the generator $\mathbf{g}$ for the vertices of the hypercuboid D^n, where every vertex corresponds to the appropriate antecedent of the rule.

Proof. The proof is straightforward and will be omitted, since the procedure for computing every output S_j applies in the same manner as in the proof of Theorem 2.4. The formal proof of Theorem 2.10 comes down to substituting the vector of consequents of the rules $\mathbf{q}$ by the matrix $\mathbf{Q}$ in the equations (2.45) and (2.30). □

2.7 Equivalence Problem in the Rule-Based Systems

The problem of equivalence between the systems of fuzzy "If-then" rules is important especially when one compares the outcomes obtained by various experts or designers, and the number of inputs is greater than two. In the systems with n inputs, there are 2^n fuzzy rules. Thus, the number of ways of ordering "If" parts is $(2^n)!$ and the number of generators and fundamental matrices is $(2^n)!$ as well. For example, for $n = 2$ we have $\left(2^2\right)! = 24$, but for $n = 3$ there are 40 320 possibilities of writing the rules. The systems of rules can be equivalent or not.

We call two rule-based systems *equivalent* if, and only if, their crisp outputs are the same for the same inputs from the universe of discourse D^n. In order to avoid mistakes in computations, which may occur especially for systems with $n \geq 3$ inputs, the designer must know exactly the relationship between the fuzzy "If-then" rules containing antecedents and consequents, and their algebraic counterparts in the form of generators, fundamental matrices, and consequents of the rules. We will show that the relationship between elements of generators and the particular consequents of the rules plays a key role; they must correspond to each other.

More precisely, our goal is to explain why the results formulated by Theorems 2.4 and 2.10 are valid independently of the order of fuzzy “If-then” rules. Without loss of generality we consider two MISO systems, which are defined by two pairs of matrices describing antecedents and consequents of the rules: $\left(\mathbf{M}_{[1]}, \mathbf{Q}_{[1]}\right)$ and $\left(\mathbf{M}_{[2]}, \mathbf{Q}_{[2]}\right)$, respectively. The systems differ from each other in two rows, namely, the rth row in matrices $\mathbf{M}_{[1]}$ and $\mathbf{Q}_{[1]}$ is the same as the sth row of $\mathbf{M}_{[2]}$ and $\mathbf{Q}_{[2]}$, respectively, and vice-versa:

$$\mathbf{M}_{[1]} = \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_r \\ \vdots \\ \mathbf{m}_s \\ \vdots \\ \mathbf{m}_{2^n} \end{bmatrix}, \quad \mathbf{Q}_{[1]} = \begin{bmatrix} q_1 \\ \vdots \\ q_r \\ \vdots \\ q_s \\ \vdots \\ q_{2^n} \end{bmatrix}, \quad \mathbf{M}_{[2]} = \begin{bmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_s \\ \vdots \\ \mathbf{m}_r \\ \vdots \\ \mathbf{m}_{2^n} \end{bmatrix}, \quad \mathbf{Q}_{[2]} = \begin{bmatrix} q_1 \\ \vdots \\ q_s \\ \vdots \\ q_r \\ \vdots \\ q_{2^n} \end{bmatrix},$$

where $\mathbf{m}_1, \ldots, \mathbf{m}_{2^n} \in \{N_1, P_1\} \times \cdots \times \{N_n, P_n\}$, and $q_1, \ldots, q_{2^n} \in \mathbb{R}$.

From the proof of Theorem 2.4 (equations (2.29)-(2.31)), we immediately obtain that the outputs of such systems are the same. Of course, the above systems have different generators

$$\mathbf{g}_{[1]}\left(z_1, \ldots, z_n\right) = \begin{bmatrix} g_0 \\ \vdots \\ z_1^{r_1} \cdots z_n^{r_n} \\ \vdots \\ z_1^{s_1} \cdots z_n^{s_n} \\ \vdots \\ g_{2^n} \end{bmatrix}, \qquad \mathbf{g}_{[2]}\left(z_1, \ldots, z_n\right) = \begin{bmatrix} g_0 \\ \vdots \\ z_1^{s_1} \cdots z_n^{s_n} \\ \vdots \\ z_1^{r_1} \cdots z_n^{r_n} \\ \vdots \\ g_{2^n} \end{bmatrix},$$

where the powers $r_1, \ldots, r_n$ and $s_1, \ldots, s_n$ are from the set $\{0, 1\}$. The power r_k or s_k is 0, if kth element of the row $\mathbf{m}_r$ contains the label N_k, or 1, if kth element of the row $\mathbf{m}_r$ contains the label P_k. Consequently, the fundamental matrices $\mathbf{\Omega}_{[1]}$ and $\mathbf{\Omega}_{[2]}$ are different, according to their own generators. In spite of this, the above systems of rules are equivalent, i.e. they generate the same output, which can be expressed using formerly proved Theorems 2.4 and 2.10. We omit the formal proof, because it is a simple consequence of Theorem 2.4. Instead of this let us consider an example.

Example 2.11. Consider two rule bases for the systems with $n = 3$ inputs and $m = 1$ output, using the matrix description of the rules. Let the generator of the first system $\mathbf{g}_{[1]}\left(z_1, z_2, z_3\right)$ be the same as in (2.40). Thus,

$$\mathbf{M}_{[1]} = \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ P_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix}, \ \mathbf{Q}_{[1]} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix}, \ \mathbf{g}_{[1]}(z_1, z_2, z_3) = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \\ z_3 \\ z_1 z_3 \\ z_2 z_3 \\ z_1 z_2 z_3 \end{bmatrix}, \tag{2.54}$$

and the corresponding fundamental matrix is given by (2.41). The second system differs from the first one in the replacement of the 4th row in the matrix $\mathbf{M}_{[1]}$ with the 5th one, i.e. we replace the row (P_1, P_2, N_3) for (N_1, N_2, P_3) and vice-versa. Thus,

$$\mathbf{M}_{[2]} = \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & P_2 & N_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix}, \ \mathbf{Q}_{[2]} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_5 \\ q_4 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix}, \ \mathbf{g}_{[2]}(z_1, z_2, z_3) = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_3 \\ z_1 z_2 \\ z_1 z_3 \\ z_2 z_3 \\ z_1 z_2 z_3 \end{bmatrix}. \tag{2.55}$$

By generating the fundamental matrix for the second system we take the following concatenation of the columns

$$\begin{aligned} \mathbf{\Omega}_{[2]} = [& \mathbf{g}_{[2]}(-\alpha_1, -\alpha_2, -\alpha_3), \mathbf{g}_{[2]}(\beta_1, -\alpha_2, -\alpha_3), \mathbf{g}_{[2]}(-\alpha_1, \beta_2, -\alpha_3), \\ & \mathbf{g}_{[2]}(-\alpha_1, -\alpha_2, \beta_3), \mathbf{g}_{[2]}(\beta_1, \beta_2, -\alpha_3), \mathbf{g}_{[2]}(\beta_1, -\alpha_2, \beta_3), \\ & \mathbf{g}_{[2]}(-\alpha_1, \beta_2, \beta_3), \mathbf{g}_{[2]}(\beta_1, \beta_2, \beta_3)] \end{aligned}$$

$$= \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & -\alpha_3 & \alpha_1\alpha_2 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_3 & -\alpha_2\beta_1 & -\beta_1\alpha_3 & \alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_3 & -\alpha_1\beta_2 & \alpha_1\alpha_3 & -\alpha_3\beta_2 & \alpha_1\alpha_3\beta_2 \\ 1 & -\alpha_1 & -\alpha_2 & \beta_3 & \alpha_1\alpha_2 & -\alpha_1\beta_3 & -\alpha_2\beta_3 & \alpha_1\alpha_2\beta_3 \\ 1 & \beta_1 & \beta_2 & -\alpha_3 & \beta_1\beta_2 & -\beta_1\alpha_3 & -\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ 1 & \beta_1 & -\alpha_2 & \beta_3 & -\alpha_2\beta_1 & \beta_1\beta_3 & -\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 \\ 1 & -\alpha_1 & \beta_2 & \beta_3 & -\alpha_1\beta_2 & -\alpha_1\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_2\beta_3 \\ 1 & \beta_1 & \beta_2 & \beta_3 & \beta_1\beta_2 & \beta_1\beta_3 & \beta_2\beta_3 & \beta_1\beta_2\beta_3 \end{bmatrix}^T . \tag{2.56}$$

As one can see $\mathbf{\Omega}_{[1]} \neq \mathbf{\Omega}_{[2]}$. From the logical point of view both systems describe the same mapping $D^3 \rightarrow \mathbb{R}$, but we want to prove this using formerly formulated theorems. From (2.51) we have $\mathbf{S}(\mathbf{z}) = \mathbf{Q}^T \mathbf{\Omega}^{-1} \mathbf{g}(\mathbf{z})$ and therefore

$$\mathbf{S}_{[1]}(\mathbf{z}) = \mathbf{Q}_{[1]}^T \mathbf{\Omega}_{[1]}^{-1} \mathbf{g}_{[1]}(\mathbf{z}), \qquad \mathbf{S}_{[2]}(\mathbf{z}) = \mathbf{Q}_{[2]}^T \mathbf{\Omega}_{[2]}^{-1} \mathbf{g}_{[2]}(\mathbf{z}). \tag{2.57}$$

After computing $\mathbf{\Omega}_{[1]}^{-1}$ and $\mathbf{\Omega}_{[2]}^{-1}$, one can easily check that using equations (2.54), (2.41), (2.55) and (2.56) for $\mathbf{Q}_{[i]}$, $\mathbf{\Omega}_{[i]}$, and $\mathbf{g}_{[i]}(\mathbf{z})$, $(i = 1, 2)$, we obtain

$$\mathbf{S}_{[1]}(\mathbf{z}) = \mathbf{S}_{[2]}(\mathbf{z}),$$

for all $\mathbf{z} \in D^3$. Thus, both systems of rules describe exactly the same system, indeed. □

In this way we can apply the results of Theorems 2.4 and 2.10 independently of the order of "If-then" rules provided that every element "$z_1^{i_1} \cdots z_{n-1}^{i_{n-1}} z_n^{i_n}$" of the generator corresponds both to the appropriate sequence of labels

$$\left(A_{i_1}, \ldots, A_{i_{n-1}}, A_{i_n}\right) \in \{N_1, P_1\} \times \ldots \times \{N_{n-1}, P_{n-1}\} \times \{N_n, P_n\},$$

which occurs in the antecedent of the rule, and to the appropriate element q_v in the consequents vector $\mathbf{q}$.

An extension of the above result for MIMO systems is straightforward and will be omitted.

2.8 Summary

The simplest TS system with one input and output and two fuzzy sets for the input variable is capable of expressing exactly any nonlinear, continuous and monotonic function of one variable. The system of fuzzy rules of such TS system has clear linguistic interpretation. If the TS system approximates a nonmonotonic function, the fuzzy sets may be very difficult for interpretation, even if they satisfy boundary conditions. Therefore in the fuzzy modeling we should rather avoid nonmonotonic membership functions.

By proving Theorem 2.4 we established an exact relationship between the P1-TS systems and a class of functions to which they are equivalent. It plays a crucial role in modeling, synthesis and analysis of many physical systems by using highly interpretable fuzzy rules. The notion of the *generator* and the *fundamental matrix* of the rule-based system belong to the most important ones, both for the theory and applications. We showed that the P1-TS system is nothing else than a *multi-linear* (or *multi-affine*) *polynomial* as stated in (2.26). It is worth adding that every *Boolean* (or *switching*) *function* $\{0,1\}^n \rightarrow \{0,1\}$ has a unique representation as a multi-linear polynomial. Such representation has been originally introduced by Zhegalkin [219] and was called *canonical polynomial form* of a Boolean function and plays an important role in many applications [6], [46], [48], [118].

The question arises: "What is the class of polynomials of the form (2.26)?" We can say informally that two multivariate polynomials are structurally the same if they differ in nonzero coefficients. Thus, the number of structurally different functions of n variables performed by the considered TS systems is 2^{2^n}. Observe that (2.26) is a part of the well-known *Kolmogorov-Gabor polynomial* (KGP for short) [49], [68]. More precisely, a *zero-order TS model* with *two linear* membership functions is equivalent to the KGP minus all the components of the type $z_i^m \cdots z_j^l \cdots z_k^r$ with the powers $\max\{m, l, \ldots, r\} \geq 2$ for $n > 1$. This observation seems to be worthy of discussion. The output of the zero-order TS system with n inputs will be denoted by S and the Kolmogorov-Gabor polynomial by KGP_n. We will say that two polynomials $p_1(\mathbf{z})$ and $p_2(\mathbf{z})$ are *equally powerful* and write $p_1 \equiv p_2$, if they are the same with the exception of nonzero coefficients, e.g. $1+2z_1+z_1^2 z_2 \equiv 3+5z_1-4z_1^2 z_2$. Furthermore, we will say that $p_1(\mathbf{z})$ is more *powerful* then $p_2(\mathbf{z})$, and write $p_1(\mathbf{z}) \supset p_2(\mathbf{z})$, if all monomials from $p_2(\mathbf{z})$ are included in $p_1(\mathbf{z})$ and at least one monomial (with nonzero coefficient) is included in $p_1(\mathbf{z})$, but not in $p_2(\mathbf{z})$, e.g. $1 + 2z_1 + z_1 z_2 + z_1^2 \supset 3 + 5z_1 + z_1 z_2$. One can prove that $KGP_n \supset S$ for all $n > 1$. For example, in the case of the system with $n = 4$ variables, the KGP has exactly 70 coefficients that uniquely define KGP_4, whereas a zero-order TS system has 16 coefficients only. A different situation occurs when we allow the rules in which the consequents are polynomials or the membership functions of fuzzy sets are polynomials of the degree $n > 1$.

One of the most important interpretations of Theorem 2.4 says that by formulating the consequents of the fuzzy rules which should express a given function f, the only information needed by an expert are the values of this function in all vertices of the hypercuboid D^n.

We introduced a compact matrix description of the MIMO P1-TS model. Observe that we can always set up a sequence of the antecedents of the rules e.g. by ordering the vertices of the hypercuboid D^n as shown in Section 2.3. In such case we can obtain an unambiguous model of the rule-based system in the matrix form (2.48) by establishing only the matrix of consequents of the rules. This fact can be used for the preservation of the computer memory needed to store the expert knowledge about the process modeled by a TS model.

Finally, we considered an equivalence problem of the rule-bases in the context of the matarules taking into account that in reality the rule-bases can be noncomplete and/or contradictory ones. The theorems proved in this chapter are valid independently of the sequences of the rules of a TS model.

Chapter 3
Recursion in TS Systems with Two Fuzzy Sets for Every Input

The fuzzy rule-based systems exhibit the "curse of dimensionality" [14], because they grow exponentially with the number of inputs. By adding an extra dimension to the input space we observe a twofold increase in the number of fuzzy "If-then" rules in the MISO P1-TS system and a rapid increase in the volume occupied by the matrices and vectors which represent this system. Namely, in the case of MISO P1-TS systems with n inputs, the dimension of the generator is 2^n, the fundamental matrix contains 4^n elements and the number of parameters of the function to which this system is equivalent, is 2^n.

One of the main questions in this chapter is to consider how the problem of computing the crisp output of the P1-TS system can be reduced to smaller problems of the same type, and how the solutions to these smaller problems can be used to construct a solution for the original one. In other words we want to develop a *recursive procedure* to solve the problem of "How to obtain the function performed by the P1-TS system containing a large number of rules?" To the best of the author's knowledge this problem has not been solved in the literature as yet. We will show that thanks to recursion, the problem of the curse of dimensionality in the rule-based systems can be substantially reduced.

3.1 Some Features of the Fundamental Matrix and Its Inverse

The fundamental matrix is important for P1-TS systems analysis and synthesis and therefore we show some of its features. In many cases we should compute its inverse (as in Example 2.11) and therefore we give some results concerning this problem. As presented before, the subscripts in the matrix or vector name will be used to indicate the number of system inputs, if necessary.

Theorem 3.1. *Let us consider a P1-TS system with inputs z_1, z_2, ..., z_k, ($1 \leq k \leq n$). One can compute recursively the inverse of the fundamental matrix $\mathbf{\Omega}_k$ as follows*

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 25–59.
springerlink.com

$$\mathbf{\Omega}_0 = 1,$$

$$\mathbf{\Omega}_{k+1}^{-1} = \frac{1}{L_{k+1}} \begin{bmatrix} \beta_{k+1} & -1 \\ \alpha_{k+1} & 1 \end{bmatrix} \otimes \mathbf{\Omega}_k^{-1}, \qquad k = 0, 1, 2, \ldots, n-1, \tag{3.1}$$

where $L_k = \alpha_k + \beta_k$ is the interval length and $\otimes$ is the Kronecker product.

Proof. According to (2.43) from Section 2.4 and (A.4) from Appendix A we obtain

$$\mathbf{\Omega}_{k+1}^{-1} = \left(\begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix} \otimes \mathbf{\Omega}_k \right)^{-1} = \frac{1}{\alpha_{k+1} + \beta_{k+1}} \begin{bmatrix} \beta_{k+1} & -1 \\ \alpha_{k+1} & 1 \end{bmatrix} \otimes \mathbf{\Omega}_k^{-1},$$

for $k = 0, 1, 2, \ldots, n-1$. This ends the proof of Theorem 3.1. □

Thanks to the above result, the computation of the inverse of the fundamental matrix does not need classical matrix inversion procedures. The matrix inverse can be found recursively using multiplication operations only.

There are many interesting features of the fundamental matrix and its inverse. To them belongs the matrix orthogonality. A matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ containing the nonzero rows $\mathbf{a}_1^T, \ldots, \mathbf{a}_m^T$ is thought to be orthogonal, if the scalar product $\mathbf{a}_i^T \mathbf{a}_j = 0$ for $i \neq j$, $(i, j = 1, \ldots, m)$. If, additionally, the scalar product is $\mathbf{a}_i^T \mathbf{a}_i = 1$ for $i = 1, \ldots, m$, then we call the matrix $\mathbf{A}$ orthonormal one.

Remark 3.2. Let us consider a P1-TS system with inputs $z_1, z_2, \ldots, z_n$, where $z_k \in [-\alpha_k, \beta_k]$ for $k = 1, \ldots, n$.

1. The following equality

$$\mathbf{\Omega}_{k+1} \mathbf{\Omega}_{k+1}^T = \begin{bmatrix} 2 & \beta_{k+1} - \alpha_{k+1} \\ \beta_{k+1} - \alpha_{k+1} & \alpha_{k+1}^2 + \beta_{k+1}^2 \end{bmatrix} \otimes \mathbf{\Omega}_k \mathbf{\Omega}_k^T, \tag{3.2}$$

   is satisfied for $k = 0, 1, 2, \ldots, n-1$.
2. The fundamental matrix $\mathbf{\Omega}$ has orthogonal rows if, and only if $\beta_k = \alpha_k$ for $k = 1, \ldots, n$.

   Proof is given in Appendix C.1.

Remark 3.3. For a P1-TS system with the inputs $z_1, z_2, \ldots, z_n$, where $z_k \in [-\alpha_k, \alpha_k]$ for $k = 1, \ldots, n$, the inverse of the fundamental matrix is given by

$$\mathbf{\Omega}^{-1} = 2^{-n} \mathbf{\Omega}^T \mathbf{\Lambda}_n^{-2}, \tag{3.3}$$

where $\mathbf{\Lambda}_n^{-2} = \mathbf{\Lambda}_n^{-1} \mathbf{\Lambda}_n^{-1}$ and the diagonal matrix $\mathbf{\Lambda}_n$ is given by a simple recurrence

$$\mathbf{\Lambda}_0 = 1,$$
$$\mathbf{\Lambda}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_{k+1} \end{bmatrix} \otimes \mathbf{\Lambda}_k, \qquad k = 0, 1, 2, \ldots, n-1. \tag{3.4}$$

Proof of this Remark is given in Appendix C.2. Thus, the rows of the fundamental matrix are orthogonal if, and only if the universe of discourse of the rule-based system is a special hypercuboid: $D^n = [-\alpha_1, \alpha_1] \times \ldots \times [-\alpha_n, \alpha_n]$, ($\alpha_k > 0$ for $k = 1, \ldots, n$).

Remark 3.4. For P1-TS system with inputs z_1, z_2, $\ldots$, z_n, where $z_k \in [-1, 1]$ for $k = 1, \ldots, n$, the inverse of the fundamental matrix is given by

$$\mathbf{\Omega}^{-1} = 2^{-n} \mathbf{\Omega}^T \ . \tag{3.5}$$

Proof of Remark 3.4 is a simple consequence of the equations (3.3)-(3.4), since $\mathbf{\Lambda}_n$ is the unity matrix for all n. Thus, the formula (3.5) is valid for P1-TS systems, in which the universe of discourse is the hypercube $D^n = [-1, 1]^n$. It is extremely simple and enables one to compute the inverse of the fundamental matrix by using its transpose, which is divided by the volume of the hypercube $[-1, 1]^n$.

3.2 Theorem on Recursion for P1-TS Systems

We begin our considerations with a simple example.

Example 3.5. The problem lies in obtaining the simplest function $S(\mathbf{z}) = c = \mathit{const}$ from the fuzzy rules which define a P1-TS system. From the logical point of view it is clear that one metarule in the form

If z_1 is ANY and z_2 is ANY and $\ldots$ and z_n is ANY, then $S = c$,

is an adequate model. Our goal is to prove that $S(\mathbf{z}) = c$ for all $\mathbf{z} \in D^n$.

Proof. (by induction). The above metarule is equivalent to the set of 2^n complete and noncontradictory fuzzy "If-then" rules, which contain the consequent "c" in every "then" part. For the MISO P1-TS system with n-inputs constituting the vector $\mathbf{z} = [z_1, \ldots, z_n]^T$, let us denote by $S_n(\mathbf{z})$ its output, and by $\mathbf{g}_n(\mathbf{z})$ and $\mathbf{\Omega}_n$ - its generator and fundamental matrix, respectively. From (2.45) we have

$$S_n(\mathbf{z}) = \mathbf{q}_n^T \mathbf{\Omega}_n^{-1} \mathbf{g}_n(\mathbf{z}), \qquad \mathbf{q}_n = [c, \ldots, c]^T = \mathbf{c} \in \mathbb{R}^{2^n}. \tag{3.6}$$

For $n = 1$, the input $z_1 \in [-\alpha_1, \beta_1]$, and there are two fuzzy rules: "If z_1 is N_1, then $S_1 = c$" and "If z_1 is P_1, then $S = c$", the generator $\mathbf{g}_1(z_1)$ is given by (2.34), and the fundamental matrix by (2.35). Thus,

$$S_1(z_1) = [c, c] \begin{bmatrix} 1 & 1 \\ -\alpha_1 & \beta_1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ z_1 \end{bmatrix} = c.$$

Assume that (3.6) is true for n, i.e. $S_n(\mathbf{z}) = c$. For the system with the inputs which constitute the vector $[\mathbf{z}, z_{n+1}]^T$, from (2.42)-(2.43) and Theorem 3.1 we obtain

$$\begin{aligned} S_{n+1}(\mathbf{z}, z_{n+1}) &= \left[\mathbf{c}^T, \mathbf{c}^T\right] \mathbf{\Omega}_{n+1}^{-1} \mathbf{g}_{n+1}(\mathbf{z}, z_{n+1}) \\ &= \left[\mathbf{c}^T, \mathbf{c}^T\right] \frac{1}{L_{n+1}} \begin{bmatrix} \beta_{n+1}\mathbf{\Omega}_n^{-1} & -\mathbf{\Omega}_n^{-1} \\ \alpha_{n+1}\mathbf{\Omega}_n^{-1} & \mathbf{\Omega}_n^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{g}_n(\mathbf{z}) \\ z_{n+1}\mathbf{g}_n(\mathbf{z}) \end{bmatrix} \\ &= \frac{1}{L_{n+1}} \left[\mathbf{c}^T \beta_{n+1}\mathbf{\Omega}_n^{-1} + \mathbf{c}^T \alpha_{n+1}\mathbf{\Omega}_n^{-1}, \quad \mathbf{0}\right] \begin{bmatrix} \mathbf{g}_n(\mathbf{z}) \\ z_{n+1}\mathbf{g}_n(\mathbf{z}) \end{bmatrix} \\ &= \mathbf{c}^T \mathbf{\Omega}_n^{-1} \mathbf{g}_n(\mathbf{z}) = S_n(\mathbf{z}) = c. \end{aligned}$$

This ends the proof that the crisp system output is $S(\mathbf{z}) = c$ for all $\mathbf{z} \in D^n$. □

In the above example one can see that coefficients of the function in (2.26) constitute the vector $\boldsymbol{\theta} = [c, 0, \ldots, 0]^T$. The output of the rule-based system does not depend on inputs and this fact logically follows from the knowledge expressed by the rules.

3.2.1 Rule-Base Decomposition

The former computing methods to evaluate the crisp output of the P1-TS systems in a general case seem to be not very convenient. Therefore we aspire to give a more suitable algorithm to compute the system output.

Without loss of generality we will consider P1-TS system with one output. Suppose a MISO P1-TS system with n inputs constituting the vector $\mathbf{z} = [z_1, \ldots, z_n]^T \in D^n$, $(n = 2, 3, \ldots)$ is given by 2^n complete and noncontradictory fuzzy rules as in (2.32). Observe that one can always decompose this system into the following two subsystems:

$$\left.\begin{array}{l} \left.\begin{array}{ll} R_1: & \text{If } \mathcal{P}_1 \text{ and } z_n \text{ is } N_n, \text{ then } S = q_1, \\ \vdots & \\ R_{2^{n-1}}: & \text{If } \mathcal{P}_{2^{n-1}} \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{2^{n-1}}, \end{array}\right\} \\ \left.\begin{array}{ll} R_{2^{n-1}+1}: & \text{If } \mathcal{P}_1 \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{2^{n-1}+1}, \\ \vdots & \\ R_{2^n}: & \text{If } \mathcal{P}_{2^{n-1}} \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{2^n}, \end{array}\right\} \end{array}\right\} \tag{3.7}$$

where $\mathcal{P}_1$, $\mathcal{P}_2$, ..., $\mathcal{P}_{2^{n-1}}$ are "If" parts of the P1-TS system with $(n-1)$ inputs $[z_1, z_2, \ldots, z_{n-1}]^T \in D^{n-1}$, $(n = 2, 3, \ldots)$, i.e.

$$\left.\begin{array}{ll} R_1': & \underbrace{\text{If } z_1 \text{ is } N_1 \text{ and } \ldots \text{ and } z_{n-1} \text{ is } N_{n-1}}_{\mathcal{P}_1}, \text{ then } S = q_1, \\ \vdots & \\ R_{2^{n-1}}': & \underbrace{\text{If } z_1 \text{ is } P_1 \text{ and } \ldots \text{ and } z_{n-1} \text{ is } P_{n-1}}_{\mathcal{P}_{2^{n-1}}}, \text{ then } S = q_{2^{n-1}}. \end{array}\right\} \tag{3.8}$$

For the sake of simplicity we assume the following notation for the P1-TS systems:

- $S(\mathbf{z} \mid \mathbf{q}_n)$ is system output S by the input variables $z_1, \ldots, z_n$ as components of the vector $\mathbf{z} \in D^n$ provided that the consequents of the rules q_1, $\ldots$, q_{2^n} are components of the vector $\mathbf{q}_n$.
- $\mathbf{g}_n(\mathbf{z})$ is the generator and $\mathbf{\Omega}_n$ is the fundamental matrix of the system by the input vector $\mathbf{z} \in D^n$.

Thus, we can rewrite (2.45) as follows

$$S(\mathbf{z} \mid \mathbf{q}_n) = \mathbf{g}_n^T(\mathbf{z}) \left(\mathbf{\Omega}_n^T\right)^{-1} \mathbf{q}_n , \tag{3.9}$$

which will be used further on.

3.2.2 Crisp Output Calculation for P1-TS System Using Recursion

Now we prove the following

Theorem 3.6. *For any natural $n \geq 2$ the recursive formula that enables one to compute the crisp output of any P1-TS system with the inputs $z_1 \in [-\alpha_1, \beta_1]$, $\ldots$, $z_n \in [-\alpha_n, \beta_n]$, is as follows*

$$\begin{aligned} S_n(z_1, \ldots, z_n \mid q_1, \ldots, q_{2^n}) = {} & N_n(z_n)\, S_{n-1}(z_1, \ldots, z_{n-1} \mid q_1, \ldots, q_{2^{n-1}}) \\ & + P_n(z_n)\, S_{n-1}(z_1, \ldots, z_{n-1} \mid q_{2^{n-1}+1}, \ldots, q_{2^n}), \end{aligned} \tag{3.10}$$

where

- $S_n(z_1, \ldots, z_n \mid q_1, \ldots, q_{2^n})$ *is the crisp output of the P1-TS system described by the fuzzy rules (3.7), with input variables $(z_1, \ldots, z_n) \in D^n$ and the consequents of the rules constituting the vector $[q_1, \cdots, q_{2^n}]^T$,*
- $N_n(z_n)$ *and* $P_n(z_n)$ *are membership functions for the input variable $z_n \in [-\alpha_n, \beta_n]$ defined by (2.11)-(2.12),*
- $S_{n-1}(z_1, \ldots, z_{n-1} \mid q_1, \ldots, q_{2^{n-1}})$ *is the crisp output of the P1-TS system described by the fuzzy rules (3.8), with the inputs $(z_1, \ldots, z_{n-1}) \in D^{n-1}$ and the consequents of the rules constituting the vector $[q_1, \cdots, q_{2^{n-1}}]^T$,*

- $S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n})$ *is the crisp output of the P1-TS system described by the fuzzy rules (3.8), with input variables* $(z_1,\ldots,z_{n-1}) \in D^{n-1}$, *replacing its consequents with* $[q_{2^{n-1}+1},\ldots,q_{2^n}]^T$.

Proof. For $n=1$ the system is defined by the rules (2.33) and the membership functions of fuzzy sets for the input z_1 are as in (2.11)-(2.12) by $k=1$. Thus, the system output is the following

$$S_1(z_1 \mid q_1,q_2) = \frac{\beta_1 - z_1}{\alpha_1+\beta_1}q_1 + \frac{z_1+\alpha_1}{\alpha_1+\beta_1}q_2. \tag{3.11}$$

For $n=2$ the system is defined by the rules (2.36) and the membership functions for the input z_2 are $N_2(z_2) = (\alpha_2+\beta_2)^{-1}(\beta_2 - z_2)$ and $P_2(z_2) = 1 - N_2(z_2)$. Thus, from (3.10) and (3.11) we obtain

$$\begin{aligned} S_2(z_1,z_2 \mid q_1,q_2,q_3,q_4) &= N_2(z_2)\,S_1(z_1 \mid q_1,q_2) + P_2(z_2)\,S_1(z_1 \mid q_3,q_4) \\ &= \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \theta_3 z_1 z_2\ , \end{aligned}$$

where

$$\begin{aligned} \theta_0 &= \frac{\alpha_1\beta_2 q_2 + \beta_1\beta_2 q_1 + \alpha_1\alpha_2 q_4 + \alpha_2\beta_1 q_3}{(\alpha_2+\beta_2)(\alpha_1+\beta_1)}, \\ \theta_1 &= \frac{\beta_2 q_2 + \alpha_2 q_4 - \beta_2 q_1 - \alpha_2 q_3}{(\alpha_2+\beta_2)(\alpha_1+\beta_1)}, \\ \theta_2 &= \frac{-\alpha_1 q_2 - \beta_1 q_1 + \alpha_1 q_4 + \beta_1 q_3}{(\alpha_2+\beta_2)(\alpha_1+\beta_1)}, \\ \theta_3 &= \frac{q_1 - q_2 - q_3 + q_4}{(\alpha_2+\beta_2)(\alpha_1+\beta_1)}. \end{aligned}$$

One can check that the same result is obtained using the formula (2.45), i.e.

$$S_2(z_1,z_2) = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \end{bmatrix},$$

and this completes the proof for $n=2$.

Now assume that (3.10) is correct for n. From Theorem 3.1, the equations (3.9) and (2.42), the system output can be expressed as follows

$$\begin{aligned} S_n(\mathbf{z} \mid \mathbf{q}_n) &= \mathbf{q}_n^T \mathbf{\Omega}_n^{-1} \mathbf{g}_n(\mathbf{z}) \\ &= \frac{1}{L_n}\left[\mathbf{a}^T, \mathbf{b}^T\right] \begin{bmatrix} \beta_n \mathbf{\Omega}_{n-1}^{-1} & -\mathbf{\Omega}_{n-1}^{-1} \\ \alpha_n \mathbf{\Omega}_{n-1}^{-1} & \mathbf{\Omega}_{n-1}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) \\ z_n \mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) \end{bmatrix}, \end{aligned}$$

where $\mathbf{a}^T = [q_1,\cdots,q_{2^{n-1}}]$, and $\mathbf{b}^T = [q_{2^{n-1}+1},\cdots,q_{2^n}]$. Next we obtain

$$S_n(\mathbf{z} \mid \mathbf{q}_n) = \frac{1}{L_n}\left[\mathbf{a}^T\beta_n + \mathbf{b}^T\alpha_n \ , \ \mathbf{b}^T - \mathbf{a}^T\right]\boldsymbol{\Omega}_{n-1}^{-1}\begin{bmatrix} \mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) \\ z_n\mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) \end{bmatrix}$$
$$= \frac{(\beta_n - z_n)\,\mathbf{a}^T\boldsymbol{\Omega}_{n-1}^{-1}\mathbf{g}_{n-1}(z_1,\ldots,z_{n-1})}{\alpha_n+\beta_n} + \frac{(\alpha_n + z_n)\,\mathbf{b}^T\boldsymbol{\Omega}_{n-1}^{-1}\mathbf{g}_{n-1}(z_1,\ldots,z_{n-1})}{\alpha_n+\beta_n}.$$

But

$$\mathbf{a}^T\boldsymbol{\Omega}_{n-1}^{-1}\mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) = S_{n-1}(z_1,\ldots,z_{n-1} \mid q_1,\ldots,q_{2^{n-1}}),$$

and

$$\mathbf{b}^T\boldsymbol{\Omega}_{n-1}^{-1}\mathbf{g}_{n-1}(z_1,\ldots,z_{n-1}) = S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n}).$$

Thus,

$$\begin{aligned} S_n(\mathbf{z} \mid \mathbf{q}_n) &= \frac{\beta_n - z_n}{\alpha_n+\beta_n}S_{n-1}(z_1,\ldots,z_{n-1} \mid q_1,\ldots,q_{2^{n-1}}) \\ &\quad + \frac{\alpha_n + z_n}{\alpha_n+\beta_n}S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n}) \\ &= N_n(z_n)\,S_{n-1}(z_1,\ldots,z_{n-1} \mid q_1,\ldots,q_{2^{n-1}}) \\ &\quad + P_n(z_n)\,S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n}). \end{aligned} \tag{3.12}$$

This completes the proof of Theorem 3.6. □

The above result is important especially for the rule-based systems with three or more inputs. We do not need to inverse large matrices to obtain the crisp system output; the curse of dimensionality in such systems is going to disappear. A generalization of (3.10) for MIMO systems is straightforward and will be omitted.

3.3 Recursion in More General TS Systems with Two Fuzzy Sets for Every Input

Theorem 3.6 has been proved using conception of the fundamental matrix for P1-TS systems, since this matrix is important for many applications. However, we will show below that the same theorem is valid for more general class of fuzzy rule-based TS systems, i.e. the systems with two fuzzy sets for every input, where no more assumptions on membership functions such as linearity or monotonicity are necessary. Our goal in this section is to prove the following generalization of Theorem 3.6.

Theorem 3.7. *Theorem 3.6 is valid for any TS system described by the fuzzy rules (3.7), with the inputs* $z_1 \in [-\alpha_1, \beta_1]$, $\ldots$, $z_n \in [-\alpha_n, \beta_n]$, *where for any input z_k there are assigned two fuzzy sets with the normalized membership functions, i.e. $N_k : [-\alpha_k, \beta_k] \to [0,1]$ and $P_k : [-\alpha_k, \beta_k] \to [0,1]$ and $N_k(z_k) + P_k(z_k) = 1$ for $k = 1, \ldots, n$. This means that if $S_n(z_1, \ldots, z_n \mid q_1, \ldots, q_{2^n})$ denotes the crisp output of the TS system with the consequents of the rules constituting the vector $[q_1, \cdots, q_{2^n}]^T$, then for any natural $n \geq 2$ the recursive formula that enables one to compute the crisp system output is the same as (3.10).*

Proof. For $n = 1$ the system is defined by the rules (2.33). Thus, the system output is the following

$$\begin{aligned} S_1(z_1 \mid q_1, q_2) &= \frac{N_1(z_1)}{N_1(z_1) + P_1(z_1)} q_1 + \frac{P_1(z_1)}{N_1(z_1) + P_1(z_1)} q_2 \\ &= N_1(z_1) q_1 + P_1(z_1) q_2. \end{aligned} \tag{3.13}$$

As one can see we used the fact that the membership functions are normalized. For $n = 2$ we have

$$\begin{aligned} S_2(z_1, z_2 \mid q_1, q_2, q_3, q_4) &= N_1(z_1) N_2(z_2) q_1 / D_2 + P_1(z_1) N_2(z_2) q_2 / D_2 \\ &+ N_1(z_1) P_2(z_2) q_3 / D_2 + P_1(z_1) P_2(z_2) q_4 / D_2 \\ &= N_2(z_2)(N_1(z_1) q_1 + P_1(z_1) q_2) / D_2 \\ &+ P_2(z_2)(N_1(z_1) q_3 + P_1(z_1) q_4) / D_2 \\ &= N_2(z_2) S_1(z_1 \mid q_1, q_2) + P_2(z_2) S_1(z_1 \mid q_3, q_4), \end{aligned}$$

since $D_2 = \prod_{k=1}^{2} (N_k(z_k) + P_k(z_k)) = 1$. Thus, for $n = 2$ the theorem is true. Let us use simplified notation: $N_k = N_k(z_k)$, $P_k = P_k(z_k)$ and S_{k+1} instead of $S_{k+1}(z_1, \ldots, z_{k+1} \mid q_1, \ldots, q_{2^{k+1}})$. For $n = k + 1 \geq 3$ we obtain (see the rule-base decomposition (3.7))

$$\begin{aligned} S_{k+1} = &(N_1 N_2 \ldots N_k N_{k+1} q_1 / D_{k+1} + P_1 N_2 \ldots N_k N_{k+1} q_2 / D_{k+1} \\ &+ \ldots + P_1 P_2 \ldots P_k N_{k+1} q_{2^k} / D_{k+1}) \\ &+ (N_1 N_2 \ldots N_k P_{k+1} q_{2^k+1} / D_{k+1} + P_1 N_2 \ldots N_k P_{k+1} q_{2^k+2} / D_{k+1} \\ &+ \ldots + P_1 P_2 \ldots P_k P_{k+1} q_{2^{k+1}} / D_{k+1}) \\ &= N_{k+1}(N_1 N_2 \ldots N_k q_1 + \ldots + P_1 P_2 \ldots P_k q_{2^k}) / D_{k+1} \\ &+ P_{k+1}(N_1 N_2 \ldots N_k q_{2^k+1} + \ldots + P_1 P_2 \ldots P_k q_{2^{k+1}}) / D_{k+1}, \end{aligned} \tag{3.14}$$

where the denominator $D_{k+1} = \prod_{i=1}^{k+1} (N_i(z_i) + P_i(z_i)) = 1$. Knowing that $D_k = 1$ for $k = 1, 2, \ldots$ we have

$$\begin{aligned} S_k(z_1, \ldots, z_k \mid q_1, \ldots, q_{2^k}) &= N_1 N_2 \ldots N_k q_1 + P_1 N_2 \ldots N_k q_2 \\ &+ \ldots + P_1 P_2 \ldots P_k q_{2^k} \end{aligned} \tag{3.15}$$

and

$$S_k\left(z_1,\ldots,z_k \mid q_{2^k+1},\ldots,q_{2^{k+1}}\right) = N_1N_2\ldots N_k q_{2^k+1} + P_1N_2\ldots N_k q_{2^k+2} + \ldots + P_1P_2\ldots P_k q_{2^{k+1}} \; . \tag{3.16}$$

Using the above equation to (3.14) we finally obtain

$$S_{k+1}\left(z_1,\ldots,z_{k+1} \mid q_1,\ldots,q_{2^{k+1}}\right) = N_{k+1}S_k\left(z_1,\ldots,z_k \mid q_1,\ldots,q_{2^k}\right) + P_{k+1}S_k\left(z_1,\ldots,z_k \mid q_{2^k+1},\ldots,q_{2^{k+1}}\right). \tag{3.17}$$

This finishes the proof of Theorem 3.7. □

The above Theorem can be used for large rule-bases, where the membership functions of fuzzy sets cannot be linear or monotonic. According to (3.17) it can be graphically interpreted as shown in Fig. 3.1. In the case of the TS system with n inputs, the architecture can be viewed as n-layer neural network [58], [87] with linear activation functions f for all neurons, where $f\,(input) = input$. In the layer number k, the network contains exactly the same neurons S_k and every neuron has two inputs and the same weights, namely $N_k\,(z_k)$ and $P_k\,(z_k)$ for $k = 1,\ldots,n$.

A generalization of the Theorem 3.7 for MIMO systems is straightforward and will be omitted. Instead of the formal proof, the MIMO case will be exemplified further on. A computational architecture of the recursion (3.17) for MIMO systems can be easily drawn analogously to the one of Fig. 3.1, as well.

Example 3.8. Let us consider a P1-TS system with 6 inputs and one output. The generator is given by

$$\mathbf{g}_6\left(z_1,z_2,z_3,z_4,z_5,z_6\right) = \begin{bmatrix} \mathbf{g}_4 \\ z_5\mathbf{g}_4 \\ z_6\mathbf{g}_4 \\ z_5z_6\mathbf{g}_4 \end{bmatrix},$$

where $\mathbf{g}_4\left(z_1,z_2,z_3,z_4\right)$ is given by (B.15) in Appendix B. After calculations we obtain

$$\begin{aligned}\mathbf{g}_6 = [&1, z_1, z_2, z_1z_2, z_3, z_1z_3, z_2z_3, z_1z_2z_3, z_4, z_1z_4, z_2z_4, z_1z_2z_4,\\ &z_3z_4, z_1z_3z_4, z_2z_3z_4, z_1z_2z_3z_4, z_5, z_1z_5, z_2z_5, z_1z_2z_5, z_3z_5,\\ &z_1z_3z_5, z_2z_3z_5, z_1z_2z_3z_5, z_4z_5, z_1z_4z_5, z_2z_4z_5, z_1z_2z_4z_5,\\ &z_3z_4z_5, z_1z_3z_4z_5, z_2z_3z_4z_5, z_1z_2z_3z_4z_5, z_6, z_1z_6, z_2z_6,\\ &z_1z_2z_6, z_3z_6, z_1z_3z_6, z_2z_3z_6, z_1z_2z_3z_6, z_4z_6, z_1z_4z_6, z_2z_4z_6,\\ &z_1z_2z_4z_6, z_3z_4z_6, z_1z_3z_4z_6, z_2z_3z_4z_6, z_1z_2z_3z_4z_6, z_5z_6,\\ &z_1z_5z_6, z_2z_5z_6, z_1z_2z_5z_6, z_3z_5z_6, z_1z_3z_5z_6, z_2z_3z_5z_6,\\ &z_1z_2z_3z_5z_6, z_4z_5z_6, z_1z_4z_5z_6, z_2z_4z_5z_6, z_1z_2z_4z_5z_6,\\ &z_3z_4z_5z_6, z_1z_3z_4z_5z_6, z_2z_3z_4z_5z_6, z_1z_2z_3z_4z_5z_6]^T.\end{aligned}$$

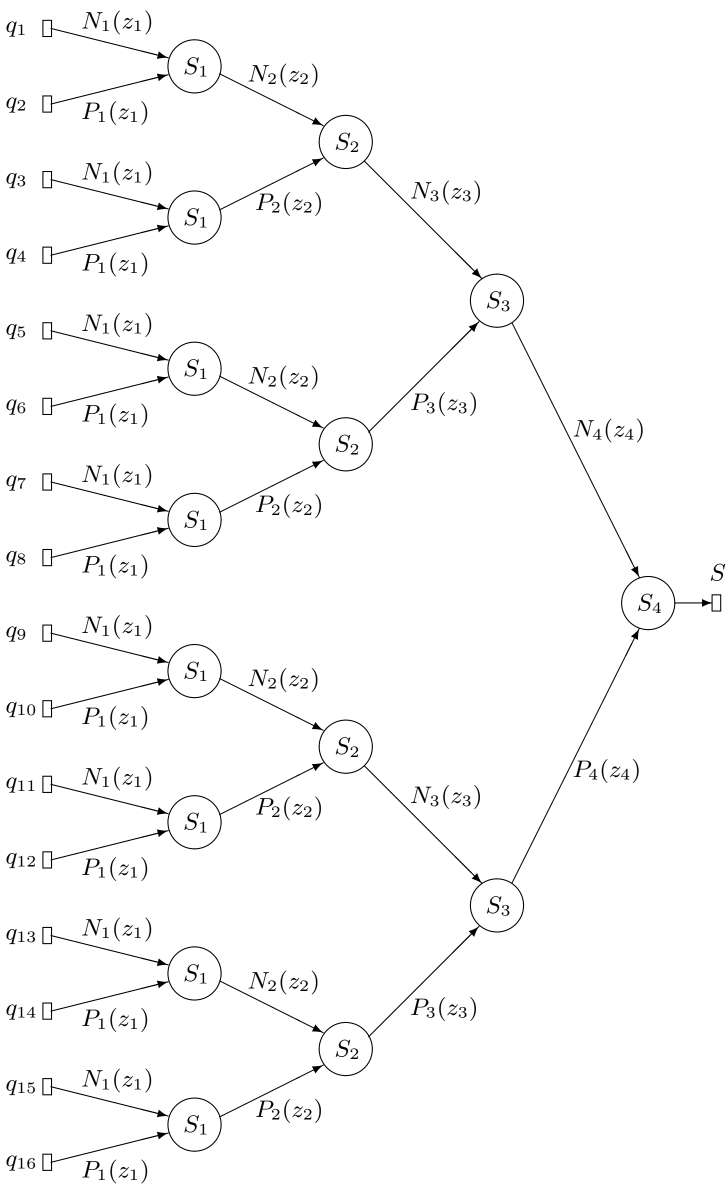


Fig. 3.1 Graphic interpretation of Theorems 3.6 and 3.7 for a TS system with $n = 4$ inputs and the output $S = S_4\left(z_1, z_2, z_3, z_4 | q_1, \ldots, q_{16}\right)$

The look-up-table for this system in a general case is given in Table 3.1. The rows and columns of this table are described by elements of the sets $\{N_1, P_1\} \times \{N_2, P_2\} \times \{N_3, P_3\}$ and $\{N_4, P_4\} \times \{N_5, P_5\} \times \{N_6, P_6\}$, respectively (the subscript k in N_k and P_k was neglected for short). The consequents q_v of

Table 3.1 Look-up-table for the P1-TS system with $n = 6$ input variables in general case

$z_1z_2z_3 \setminus z_4z_5z_6$	NNN	NNP	NPP	NPN	PPN	PPP	PNP	PNN
NNN	q_1	q_{33}	q_{49}	q_{17}	q_{25}	q_{57}	q_{41}	q_9
NNP	q_5	q_{37}	q_{53}	q_{21}	q_{29}	q_{61}	q_{45}	q_{13}
NPP	q_7	q_{39}	q_{55}	q_{23}	q_{31}	q_{63}	q_{47}	q_{15}
NPN	q_3	q_{35}	q_{51}	q_{19}	q_{27}	q_{59}	q_{43}	q_{11}
PPN	q_4	q_{36}	q_{52}	q_{20}	q_{28}	q_{60}	q_{44}	q_{12}
PPP	q_8	q_{40}	q_{56}	q_{24}	q_{32}	q_{64}	q_{48}	q_{16}
PNP	q_6	q_{38}	q_{54}	q_{22}	q_{30}	q_{62}	q_{46}	q_{14}
PNN	q_2	q_{34}	q_{50}	q_{18}	q_{26}	q_{58}	q_{42}	q_{10}

the rules containing antecedents $A_{i_1}A_{i_2}A_{i_3}A_{i_4}A_{i_5}A_{i_6} \in \times_{i=1}^{6}\{N_i, P_i\}$ have indices $v \leftrightarrow (i_1, \ldots, i_6)$ according to (2.16).

Suppose an expert formulated the following 8 metarules:

M_1 : If z_1 is N_1 and z_3 is P_3 and z_4 is N_4, then $S = a$,

M_2 : If z_3 is P_3 and z_4 is P_4 and z_6 is N_6, then $S = b$,

M_3 : If z_1 is P_1 and z_2 is N_2 and z_4 is N_4 and z_5 is N_5, then $S = c$,

M_4 : If z_1 is N_1 and z_3 is N_3, then $S = 0$,

M_5 : If z_4 is P_4 and z_6 is P_6, then $S = 0$,

M_6 : If z_1 is P_1 and z_4 is N_4 and z_5 is P_5, then $S = 0$,

M_7 : If z_1 is P_1 and z_3 is N_3 and z_4 is P_4 and z_6 is N_6, then $S = 0$,

M_8 : If z_1 is P_1 and z_2 is P_2 and z_4 is N_4, then $S = 0$,

which generate 64 fuzzy "If-then" rules. The rows and columns of Table 3.1 are described by the Gray code [85], [158], [115]. Owing to this we can easily explain and interpret all metarules. For example, the metarules M_1, M_2 and M_3 are graphically shown in Tables 3.2, 3.3 and 3.4, respectively. Analogously, we can justify the remaining metarules - they all have a clear interpretation.

One can check, that the system of rules is complete and a noncontradictory one. According to the generator $\mathbf{g}_6$ and the metarules, we obtain the consecutive elements of the vector $\mathbf{q}$ (see (2.30)), as follows:

$$\begin{array}{lllllll}
q_1 = 0, & q_2 = c, & q_3 = 0, & q_4 = 0, & q_5 = a, & q_6 = c, & q_7 = a, \\
q_8 = 0, & q_9 = 0, & q_{10} = 0, & q_{11} = 0, & q_{12} = 0, & q_{13} = b, & q_{14} = b, \\
q_{15} = b, & q_{16} = b, & q_{17} = 0, & q_{18} = 0, & q_{19} = 0, & q_{20} = 0, & q_{21} = a, \\
q_{22} = 0, & q_{23} = a, & q_{24} = 0, & q_{25} = 0, & q_{26} = 0, & q_{27} = 0, & q_{28} = 0, \\
q_{29} = b, & q_{30} = b, & q_{31} = b, & q_{32} = b, & q_{33} = 0, & q_{34} = c, & q_{35} = 0, \\
q_{36} = 0, & q_{37} = a, & q_{38} = c, & q_{39} = a, & q_{40} = 0, & q_{41} = 0, & q_{42} = 0,
\end{array}$$

$$q_{43} = 0, \quad q_{44} = 0, \quad q_{45} = 0, \quad q_{46} = 0, \quad q_{47} = 0, \quad q_{48} = 0, \quad q_{49} = 0,$$
$$q_{50} = 0, \quad q_{51} = 0, \quad q_{52} = 0, \quad q_{53} = a, \quad q_{54} = 0, \quad q_{55} = a, \quad q_{56} = 0,$$
$$q_{57} = 0, \quad q_{58} = 0, \quad q_{59} = 0, \quad q_{60} = 0, \quad q_{61} = 0, \quad q_{62} = 0, \quad q_{63} = 0,$$
$$q_{64} = 0,$$

Table 3.2 Graphical explanation of the metarule M_1 from Example 3.8

$z_1z_2z_3 \setminus z_4z_5z_6 \longrightarrow$ ↓	$\mathbf{N}NN$	$\mathbf{N}NP$	$\mathbf{N}PP$	$\mathbf{N}PN$	PPN	PPP	PNP	PNN
NNN	0	0	0	0	0	0	0	0
$\mathbf{N}N\mathbf{P}$	a	a	a	a	b	0	0	b
$\mathbf{N}P\mathbf{P}$	a	a	a	a	b	0	0	b
NPN	0	0	0	0	0	0	0	0
PPN	0	0	0	0	0	0	0	0
PPP	0	0	0	0	b	0	0	b
PNP	c	c	0	0	b	0	0	b
PNN	c	c	0	0	0	0	0	0

Table 3.3 Graphical explanation of the metarule M_2 from Example 3.8

$z_1z_2z_3 \setminus z_4z_5z_6 \longrightarrow$ ↓	NNN	NNP	NPP	NPN	$\mathbf{P}P\mathbf{N}$	PPP	PNP	$\mathbf{P}N\mathbf{N}$
NNN	0	0	0	0	0	0	0	0
$NN\mathbf{P}$	a	a	a	a	b	0	0	b
$NP\mathbf{P}$	a	a	a	a	b	0	0	b
NPN	0	0	0	0	0	0	0	0
PPN	0	0	0	0	0	0	0	0
$PP\mathbf{P}$	0	0	0	0	b	0	0	b
$PN\mathbf{P}$	c	c	0	0	b	0	0	b
PNN	c	c	0	0	0	0	0	0

Table 3.4 Graphical explanation of the metarule M_3 from Example 3.8

$z_1z_2z_3 \setminus z_4z_5z_6 \longrightarrow$ ↓	$\mathbf{NN}N$	$\mathbf{NN}P$	NPP	NPN	PPN	PPP	PNP	PNN
NNN	0	0	0	0	0	0	0	0
NNP	a	a	a	a	b	0	0	b
NPP	a	a	a	a	b	0	0	b
NPN	0	0	0	0	0	0	0	0
$\mathrm{P}PN$	0	0	0	0	0	0	0	0
$\mathrm{P}PP$	0	0	0	0	b	0	0	b
$\mathbf{PN}P$	c	c	0	0	b	0	0	b
$\mathbf{PN}N$	c	c	0	0	0	0	0	0

Using Theorem 3.6 together with its nomenclature we have $S_1(z_1 \mid q_1, q_2)$ as in (3.11) and next, after recursive computations, we obtain

$$S_2(z_1, z_2 \mid q_1, q_2, q_3, q_4) = \frac{\beta_2 - z_2}{\alpha_2 + \beta_2} S_1(z_1 \mid q_1, q_2) + \frac{\alpha_2 + z_2}{\alpha_2 + \beta_2} S_1(z_1 \mid q_3, q_4),$$

$$\begin{aligned} S_3(z_1, z_2, z_3 \mid q_1, \ldots, q_8) &= \frac{\beta_3 - z_3}{\alpha_3 + \beta_3} S_2(z_1, z_2 \mid q_1, q_2, q_3, q_4) \\ &+ \frac{\alpha_3 + z_3}{\alpha_3 + \beta_3} S_2(z_1, z_2 \mid q_5, q_6, q_7, q_8), \end{aligned}$$

$$\begin{aligned} S_4(z_1, \ldots, z_4 \mid q_1, \ldots, q_{16}) &= \frac{\beta_4 - z_4}{\alpha_4 + \beta_4} S_3(z_1, z_2, z_3 \mid q_1, \ldots, q_8) \\ &+ \frac{\alpha_4 + z_4}{\alpha_4 + \beta_4} S_3(z_1, z_2, z_3 \mid q_9, \ldots, q_{16}), \end{aligned}$$

$$\begin{aligned} S_5(z_1, \ldots, z_5 \mid q_1, \ldots, q_{32}) &= \frac{\beta_5 - z_5}{\alpha_5 + \beta_5} S_4(z_1, \ldots, z_4 \mid q_1, \ldots, q_{16}) \\ &+ \frac{\alpha_5 + z_5}{\alpha_5 + \beta_5} S_4(z_1, \ldots, z_4 \mid q_{17}, \ldots, q_{32}), \end{aligned}$$

$$\begin{aligned} S_6(z_1, \ldots, z_6 \mid q_1, \ldots, q_{64}) &= \frac{\beta_6 - z_6}{\alpha_6 + \beta_6} S_5(z_1, \ldots, z_5 \mid q_1, \ldots, q_{32}) \\ &+ \frac{\alpha_6 + z_6}{\alpha_6 + \beta_6} S_5(z_1, \ldots, z_5 \mid q_{33}, \ldots, q_{64}). \end{aligned}$$

Now assume that all inputs z_k are from the unity interval $[0, 1]$, i.e. $\alpha_k = 0$ and $\beta_k = 1$ for $k = 1, \ldots, 6$. Thus, the crisp output of the system is given by

$$\begin{aligned} S(z_1, z_2, z_3, z_4, z_5, z_6) &= a(1 - z_1) z_3 (1 - z_4) + b z_3 z_4 (1 - z_6) \\ &+ c z_1 (1 - z_2)(1 - z_4)(1 - z_5). \end{aligned} \tag{3.18}$$

This result is intuitively clear, because the first part of the sum (3.18) corresponds to the metarule M_1, the second one – to the metarule M_2, and the third one – to the metarule M_3. Furthermore, for $a = b = c = 1$ we have to do with a system, which processes information expressed in multi-valued logic, since $\mathbf{z} \in [0, 1]^6$. The form of S in (3.18) resembles the sum of implicants of Boolean function [115]. The terms "$(1 - z_1) z_3 (1 - z_4)$", "$z_3 z_4 (1 - z_6)$" and "$z_1 (1 - z_2)(1 - z_4)(1 - z_5)$" may be regarded as the generalized implicants of the function $S : [0, 1]^6 \to [0, 1]$.

Example 3.9. Suppose the same system as in Example 3.8 is given, with the exception that the inputs are from various intervals. Let us take $\alpha_1 = 5$,

$\beta_1 = -3$, $\alpha_2 = 3$, $\beta_2 = 2$, $\alpha_3 = 2$, $\beta_3 = 0$, $\alpha_4 = 0$, $\beta_4 = 4$, $\alpha_5 = 1$, $\beta_5 = 3$, $\alpha_6 = -2$, and $\beta_6 = 6$, i.e. the input vector of the P1-TS system is from the hypercuboid

$$[z_1, z_2, z_3, z_4, z_5, z_6]^T \in [-5, -3] \times [-3, 2] \times [-2, 0] \times [0, 4] \times [-1, 3] \times [2, 6] .$$

One can check that the system of metarules from Example 3.8 is equivalent to the following function of six variables

$$S(z_1, z_2, z_3, z_4, z_5, z_6) = \frac{a}{16}(z_1 + 3)(z_3 + 2)(z_4 - 4) + \frac{b}{32}(6 - z_6)(z_3 + 2) z_4 + \frac{c}{160}(3 - z_5)(4 - z_4)(2 - z_2)(z_1 + 5) .$$

The power of Theorems 3.6 and 3.7 one can see for the systems with 10 or more inputs; many of the problems similar to those of Example 3.8 or 3.9 can be solved using our theorems on recursion and symbolic computations.

3.4 MIMO TS Systems with Inference Concerning the Structure Parameters

In this section we will consider a Takagi-Sugeno fuzzy model for nonlinear continuous or discrete-time dynamical systems, proposed in [180], comprehensively investigated in textbook [184] and elaborated in many contemporary papers, especially with regard to stability of control systems [91], [125], [149], [193]. By using such approach, a complex nonlinear system can be represented by a set of fuzzy rules of which the *consequent parts are linear state equations representing the local models.* The complex nonlinear model of the whole system can then be described (inferred) as a weighted sum of these linear state equations. It is widely accepted as a powerful modeling tool. According to the classification of fuzzy systems given in [178], the considered rule-base models belong to the Type III fuzzy systems. Their applications to various kinds of nonlinear systems can be found e.g. in [90], [181], [182], [196]. Based on the TS fuzzy model of a plant, a parallel distributed compensation (PDC) technique to design fuzzy logic controller has been proposed in [184].

- **The inputs.** The inputs of the MIMO TS system which should model a nonlinear dynamical system, are quantities $z_k(t)$ constituting the vector $\mathbf{z}(t) = [z_1(t), \ldots, z_r(t)]^T \in D^r$, (see Fig. 3.2). The vector $\mathbf{z}(t)$ contains known premise variables that are not functions of the control input $\mathbf{u}(t) = [u_1(t), \ldots, u_m(t)]^T$, but they may be functions of the state variables, external disturbances and/or time. For every variable z_k we assign two complementary membership functions $N_k(z_k)$ and $P_k(z_k)$, i.e. $N_k(z_k) + P_k(z_k) = 1$, that cannot be monotonic or linear for $k = 1, \ldots, r$. At any time t the vector $\mathbf{z}(t)$ belongs to some vth region of a fuzzy partition.

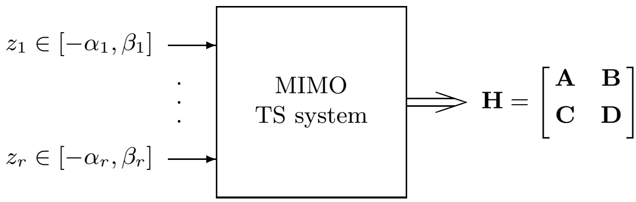


Fig. 3.2 TS system with the inputs $z_1, ..., z_r$ and the inference concerning the structure parameters

- **The outputs.** We assume that a local model of the nonlinear system can be described in a vth region by the state-space equation in the standard matrix form

$$\begin{cases} s\dot{\mathbf{x}}(t) = \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\mathbf{u}(t), \quad \mathbf{x}(0) \in \mathbf{X} \subset \mathbb{R}^n, \\ \mathbf{y}(t) = \mathbf{C}(t)\,\mathbf{x}(t) + \mathbf{D}(t)\,\mathbf{u}(t), \quad t \geq 0, \end{cases} \tag{3.19}$$

where

$$s\mathbf{x}(t) = \begin{cases} \dot{\mathbf{x}}(t) & \text{- for a continuos case, } t \geq 0, \\ \mathbf{x}(t+1) & \text{- for a discrete case, } t = 0, 1, 2, \ldots, \end{cases}$$

the matrices are $\mathbf{A}(t) : \mathbb{R}_+ \to \mathbb{R}^{n \times n}$, $\mathbf{B}(t) : \mathbb{R}_+ \to \mathbb{R}^{n \times m}$, $(m < n)$, $\mathbf{C}(t) : \mathbb{R}_+ \to \mathbb{R}^{l \times n}$, $\mathbf{D}(t) : \mathbb{R}_+ \to \mathbb{R}^{l \times m}$, $(l \leq m)$ and $\mathbb{R}_+ = [0, \infty)$. The vectors $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{y}(t)$ are contained in some hypercuboids (2.22) for $t \geq 0$. Without loss of generality we will consider a continuous-time systems and for the sake of simplicity, the time variable t will be sometimes omitted. The local linear model is unambiguously defined by the block matrix $\mathbf{H}_v$ which contains four submatrices

$$\mathbf{H}_v(t) = \begin{bmatrix} \mathbf{A}_v(t) & \mathbf{B}_v(t) \\ \mathbf{C}_v(t) & \mathbf{D}_v(t) \end{bmatrix}, \qquad v = 1, \ldots, 2^r, \tag{3.20}$$

where 2^r is the number of fuzzy rules. Thus, we define the matrix $\mathbf{H}$ as the output $\mathbf{q}$ of the MIMO TS rule-based system (see Fig. 3.2). Both the inputs and outputs of the TS system have more abstract meaning in comparison with the ones defined in the previous sections.

- **The fuzzy rules.** The following 2^r fuzzy rules are given

$$R_1 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_r \text{ is } N_r, \text{ then } \mathbf{q}_1 = \mathbf{H}_1 \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix},$$

$$R_2: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \dots \text{ and } z_r \text{ is } N_r, \text{ then } \mathbf{q}_2 = \mathbf{H}_2 \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix},$$

$$R_3: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \dots \text{ and } z_r \text{ is } N_r, \text{ then } \mathbf{q}_3 = \mathbf{H}_3 \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix},$$

$$R_4: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \dots \text{ and } z_r \text{ is } N_r, \text{ then } \mathbf{q}_4 = \mathbf{H}_4 \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix},$$

$$\vdots$$

$$R_{2^r}: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \dots \text{ and } z_r \text{ is } P_r, \text{ then } \mathbf{q}_{2^r} = \mathbf{H}_{2^r} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}. \tag{3.21}$$

By the above assumptions the inferred model of the system is as follows

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r)\,\mathbf{A}_v(t) & \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r)\,\mathbf{B}_v(t) \\ \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r)\,\mathbf{C}_v(t) & \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r)\,\mathbf{D}_v(t) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}, \tag{3.22}$$

where

$$h_v(z_1,\dots,z_r) = A_{i_1}(z_1) \times \dots \times A_{i_r}(z_r), \qquad v \in \{1,2,\dots,2^r\}, \tag{3.23}$$

the index v corresponds to only one antecedent of the fuzzy “If-then” rule, i.e. $v \leftrightarrow (i_1,\dots,i_r)$ according to the bijection (2.16), when replacing n by r. The symbol $A_{i_k} = N_k$ for $i_k = 0$ and $A_{i_k} = P_k$ for $i_k = 1$, $(k = 1,\dots,r)$. The inferred matrix $\mathbf{H}$ of the whole system is a convex combination of matrices given in (3.20)

$$\mathbf{H} = \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r)\,\mathbf{H}_v = \begin{bmatrix} \mathbf{A}(\mathbf{z}) & \mathbf{B}(\mathbf{z}) \\ \mathbf{C}(\mathbf{z}) & \mathbf{D}(\mathbf{z}) \end{bmatrix}, \quad \sum_{v=1}^{2^r} h_v(z_1,\dots,z_r) = 1. \tag{3.24}$$

Remark 3.10. In the simplest case, even if all premise variables are defined as the state variables ($\mathbf{z} = \mathbf{x}$) and all matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are constant, the model (3.22) defines a nonlinear dynamical system.

Remark 3.11. Knowing the assumptions and the way of fuzzy reasoning, we can write the fuzzy rules in a more readable format:

$$\text{If } \mathbf{z} \text{ is } \mathbf{P}_v, \text{ then } \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \mathbf{H}_v \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}, \tag{3.25}$$

where $\mathbf{P}_v$ is the premise of the rule. The vector $[\dot{\mathbf{x}}, \mathbf{y}]^T$ contains at least the velocity of state vector, i.e. we can be or not interested in modeling of the output $\mathbf{y}$. If the vector $\mathbf{y}$ is not defined, the matrices $\mathbf{C}$ and $\mathbf{D}$ are empty.

Now our goal is to apply Theorem 3.7. Let us define the output vector $\mathbf{S}_k(\mathbf{H}_1, \ldots, \mathbf{H}_{2^k})$ of the TS system with the inputs $z_1, \ldots, z_k$, in which the consequents of the rules are defined by matrices $\mathbf{H}_1, \ldots, \mathbf{H}_{2^k}$. We know that $\mathbf{S}_k(\mathbf{H}_1, \ldots, \mathbf{H}_{2^k})$ denotes the TS system output vector $\mathbf{q}$. On the other hand, for $k = 1$ we have

$$\begin{aligned} \mathbf{S}_1(\mathbf{H}_1, \mathbf{H}_2) &= N_1(z_1)\,\mathbf{q}_1 + P_1(z_1)\,\mathbf{q}_2 \\ &= \left(N_1(z_1)\,\mathbf{H}_1 + P_1(z_1)\,\mathbf{H}_2\right) \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}. \end{aligned} \tag{3.26}$$

From Theorem 3.7 for $r \geq 2$ we obtain

$$\begin{aligned} \mathbf{S}_r(\mathbf{H}_1, \ldots, \mathbf{H}_{2^r}) = N_r(z_r)\,\mathbf{S}_{r-1}(\mathbf{H}_1, \ldots, \mathbf{H}_{2^{r-1}}) \\ + P_r(z_r)\,\mathbf{S}_{r-1}(\mathbf{H}_{2^{r-1}+1}, \ldots, \mathbf{H}_{2^r}). \end{aligned} \tag{3.27}$$

This equation has the same graphic interpretation as formerly shown in Fig. 3.1 when substituting the consequents of the rules q_v by the vectors $\mathbf{H}_v\left[\mathbf{x}^T, \mathbf{u}^T\right]^T$.

Below we give some examples of modeling nonlinear systems.

Example 3.12. Consider the nonlinear differential equations of motion for the cart-pendulum mechanical system, well-known in the literature as the inverted pendulum of Fig. 3.3 (see e.g. [89], [92], [184]). The pivot of the

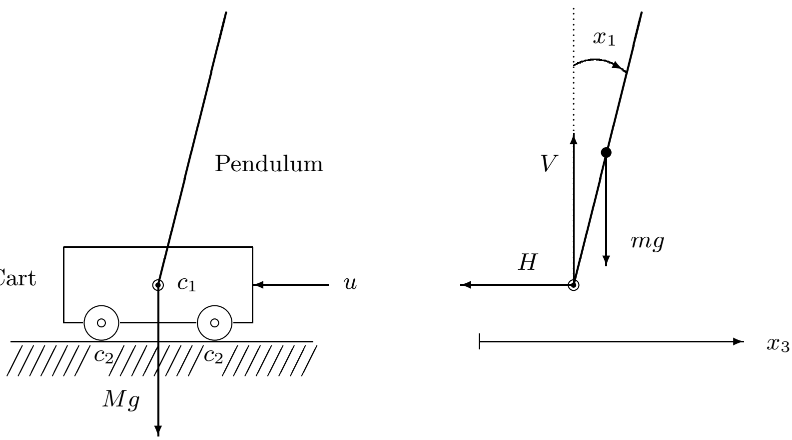


Fig. 3.3 Inverted pendulum on a cart

pendulum is mounted on a cart that can move in a horizontal direction. The cart is driven by a motor that exerts a horizontal force $u(t)$ on the cart [N]. The system variables are: $x_1(t)$ - the pendulum angle from vertical [rad], $x_2(t) = \dot{x}_1(t)$ - the pendulum angular velocity [rad / s], $x_3(t)$ - the cart position [m] and $x_4(t) = \dot{x}_3(t)$ is the cart velocity [m / s]. The system constants are: M - the mass of the cart [kg], m - the mass of the pendulum [kg], l - the distance from the center of gravity to the pivot [m], I - the moment of inertia of the pendulum [kg m^2], c_1 - the friction coefficient of the clasp [kg m^2 / s] and c_2 is the friction coefficient of the cart [N s / m]. Writing horizontal and vertical Newton's laws at the center of gravity of the pendulum, the torque equation and horizontal Newton's law for the cart, yields (see Fig. 3.3)

$$\left.\begin{aligned} &m\frac{d^2}{dt^2}(x_3 + l\sin x_1) = H, \\ &m\frac{d^2}{dt^2}(l\cos x_1) = V - mg, \\ &I\frac{dx_2}{dt} = Vl\sin x_1 - Hl\cos x_1 - c_1x_2, \\ &M\frac{dx_4}{dt} = u - H - c_2x_4. \end{aligned}\right\} \tag{3.28}$$

Let us introduce the following system constants

$$k_1 = \frac{g}{l}\left(1 + \frac{M}{m}\right), \qquad k_2 = \frac{c_1}{ml^2}\left(1 + \frac{M}{m}\right), \qquad k_3 = \frac{1}{ml},$$

$$k_4 = l\left(1 + \frac{I}{ml^2}\right), \qquad k_5 = k_3k_4 = \frac{1}{m}\left(1 + \frac{I}{ml^2}\right), \qquad k_6 = \frac{k_1k_4}{g}.$$

Taking into account the differential equations (3.28) and the above system constants and eliminating V and H, we obtain

$$\left.\begin{aligned} &\dot{x}_1 = x_2, \\ &\dot{x}_2 = \frac{k_1\sin x_1 - k_2x_2 - x_2^2\sin(x_1)\cos x_1 + c_2k_3x_4\cos x_1 - uk_3\cos x_1}{k_6 - \cos^2 x_1}, \\ &\dot{x}_3 = x_4, \\ &\dot{x}_4 = \frac{(-g\sin x_1 + c_1k_3x_2)\cos x_1 + k_4\left(x_2^2\sin(x_1) - c_2k_3x_4 + uk_3\right)}{k_6 - \cos^2 x_1}. \end{aligned}\right\} \tag{3.29}$$

To ensure controllability of the system we assume as in [184] that $x_1(t) \in \left[-x_1^H, x_1^H\right]$, where $0 < x_1^H < \pi/2$. The angular velocity is bounded, i.e. $x_2(t) \in \left[-x_2^H, x_2^H\right]$. Let us define new variables z_1, z_2, z_3 and z_4 as follows

$$z_1 = \frac{\sin x_1}{x_1} \in [-\alpha_1, \beta_1], \tag{3.30}$$

$$z_2 = \frac{1}{k_6 - \cos^2 x_1} \in [-\alpha_2, \beta_2], \tag{3.31}$$

$$z_3 = \frac{\cos x_1}{k_6 - \cos^2 x_1} \in [-\alpha_3, \beta_3], \tag{3.32}$$

$$z_4 = x_2 \sin x_1 \in [-\alpha_4, \beta_4], \tag{3.33}$$

where

$$-\alpha_1 = \frac{\sin x_1^H}{x_1^H} > 0, \qquad \beta_1 = 1, \tag{3.34}$$

$$-\alpha_2 = \frac{1}{k_6 - \cos^2 x_1^H} > 0, \qquad \beta_2 = \frac{1}{k_6 - 1} > 0, \tag{3.35}$$

$$-\alpha_3 = \frac{\cos x_1^H}{k_6 - \cos^2 x_1^H} > 0, \qquad \beta_3 = \frac{1}{k_6 - 1} = \beta_2, \tag{3.36}$$

$$-\alpha_4 = -x_2^H \sin x_1^H < 0, \qquad \beta_4 = x_2^H \sin x_1^H > 0. \tag{3.37}$$

One can check that the system (3.29) is equivalent to

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & -k_2 z_2 - z_3 z_4 & 0 & c_2 k_3 z_3 & -k_3 z_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g z_1 z_3 & c_1 k_3 z_3 + k_4 z_2 z_4 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix}, \tag{3.38}$$

Our goal is to obtain the TS rule-based system of the inverted pendulum in the following form

$$\text{If } [z_1, z_2, z_3, z_4] \text{ is } \left[A_{i_1} \ A_{i_2} \ A_{i_3} \ A_{i_4} \right],$$

$$\text{then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} \text{ is } \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ q_{1,v} & q_{2,v} & 0 & q_{4,v} & q_{5,v} \\ 0 & 0 & 0 & 1 & 0 \\ r_{1,v} & r_{2,v} & 0 & r_{4,v} & r_{5,v} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

where $(A_{i_1}, A_{i_2}, A_{i_3}, A_{i_4}) \in \{N_1, P_1\} \times \ldots \times \{N_4, P_4\}$. Thus, the inputs of the P1-TS system are z_1, z_2, z_3 and z_4 as the points of the hypercuboid D^4, whereas the outputs of this system are the velocities of the state vector, i.e. $\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$ and $\dot{x}_4$. The membership functions of fuzzy sets are $N_k(z_k) = (\alpha_k + \beta_k)^{-1}(\beta_k - z_k)$ and $P_k(z_k) = 1 - N_k(z_k)$, where α_k and β_k are given by (3.34)-(3.37) for $k = 1, 2, 3, 4$. We can evaluate the vertices of the hypercuboid D^4 for $I = ml^2/3$ and $M > m$. Independently of the values of the system constants, the inequalities $z_1(x_1) > z_2(x_1) \geq z_3(x_1)$ hold for $x_1(t) \in \left[-x_1^H, x_1^H\right]$, $(0 < x_1^H < \pi/2)$ as shown in Fig. 3.4 and the following relations

$$0 < -\alpha_3 \leq -\alpha_2 < \beta_3 = \beta_2 < -\alpha_1 < \beta_1 = 1, \qquad \alpha_4 = \beta_4 > 0,$$

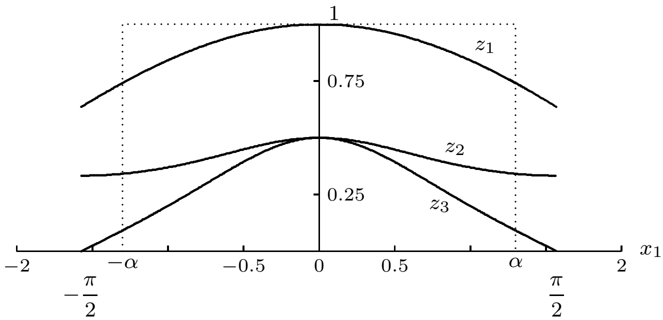


Fig. 3.4 Plot of the functions z_1, z_2 and z_3 defined in (3.30)-(3.32) by $I = ml^2/3$ and $M/m = 5/4$

are satisfied. Thus, we can interpret the fuzzy sets as follows: N_k means that z_k is *not large* and P_k that z_k is *large* (in its range) for $k = 1, 2, 3$, whereas N_4 means *negative* value of z_4, and P_4 - *positive* one.

Of course, it is possible to find all elements of the matrix containing the consequents of the rules, but we can simplify the problem, i.e. we will find $q_{i,v}$ and $r_{i,v}$. According to the above format of the rule, in order to find $q_{i,v}$ and $r_{i,v}$ we should take into account the second and the fourth row of the matrix (3.38), correspondingly. For the generator $\mathbf{g}$ for $n = 4$ (see (B.15) in Appendix B), we define the following matrices that define the functions of the inputs z_1, z_2, z_3 and z_4 of the TS system, respectively:

$$\boldsymbol{\Theta}_q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & 0 & 0 & 0 \\ k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_3c_2 & -k_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{\Theta}_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_2k_5 & k_5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & c_1k_3 & 0 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the columns of the matrix $\boldsymbol{\Theta}_q$ correspond to the conclusions $q_{1,v}, \ldots,$ $q_{5,v}$, and the columns of $\boldsymbol{\Theta}_r$ correspond to the conclusions $r_{1,v}, \ldots, r_{5,v}$. For the fundamental matrix (B.16)-(B.20) given in Appendix B we obtain

$$\mathbf{Q}_q = \mathbf{\Omega}^T \mathbf{\Theta}_q = \begin{bmatrix} \alpha_1\alpha_2k_1 & \alpha_2k_2 - \alpha_3\alpha_4 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ -\alpha_2\beta_1k_1 & \alpha_2k_2 - \alpha_3\alpha_4 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ -\alpha_1\beta_2k_1 & -\alpha_3\alpha_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ \beta_1\beta_2k_1 & -\alpha_3\alpha_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ \alpha_1\alpha_2k_1 & \alpha_4\beta_3 + \alpha_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ -\alpha_2\beta_1k_1 & \alpha_4\beta_3 + \alpha_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ -\alpha_1\beta_2k_1 & \alpha_4\beta_3 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ \beta_1\beta_2k_1 & \alpha_4\beta_3 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ \alpha_1\alpha_2k_1 & \alpha_3\beta_4 + \alpha_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ -\alpha_2\beta_1k_1 & \alpha_3\beta_4 + \alpha_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ -\alpha_1\beta_2k_1 & \alpha_3\beta_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ \beta_1\beta_2k_1 & \alpha_3\beta_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ \alpha_1\alpha_2k_1 & \alpha_2k_2 - \beta_3\beta_4 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ -\alpha_2\beta_1k_1 & \alpha_2k_2 - \beta_3\beta_4 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ -\alpha_1\beta_2k_1 & -\beta_3\beta_4 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ \beta_1\beta_2k_1 & -\beta_3\beta_4 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \end{bmatrix},$$

and

$$\mathbf{Q}_r = \mathbf{\Omega}^T \mathbf{\Theta}_r = \begin{bmatrix} -g\alpha_1\alpha_3 & \alpha_2\alpha_4k_4 - \alpha_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ g\beta_1\alpha_3 & \alpha_2\alpha_4k_4 - \alpha_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ -g\alpha_1\alpha_3 & -\beta_2\alpha_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ g\beta_1\alpha_3 & -\beta_2\alpha_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ g\alpha_1\beta_3 & \alpha_2\alpha_4k_4 + \beta_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ -g\beta_1\beta_3 & \alpha_2\alpha_4k_4 + \beta_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ g\alpha_1\beta_3 & \beta_3c_1k_3 - \beta_2\alpha_4k_4 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ -g\beta_1\beta_3 & \beta_3c_1k_3 - \beta_2\alpha_4k_4 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ -g\alpha_1\alpha_3 & -\alpha_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ g\beta_1\alpha_3 & -\alpha_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ -g\alpha_1\alpha_3 & \beta_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ g\beta_1\alpha_3 & \beta_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ g\alpha_1\beta_3 & \beta_3c_1k_3 - \alpha_2\beta_4k_4 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ -g\beta_1\beta_3 & \beta_3c_1k_3 - \alpha_2\beta_4k_4 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \\ g\alpha_1\beta_3 & \beta_2\beta_4k_4 + \beta_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \\ -g\beta_1\beta_3 & \beta_2\beta_4k_4 + \beta_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \end{bmatrix}.$$

Let

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \alpha_1\alpha_2 k_1 & \alpha_2 k_2 - \alpha_3\alpha_4 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\alpha_1\alpha_3 & \alpha_2\alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2\beta_1 k_1 & \alpha_2 k_2 - \alpha_3\alpha_4 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\beta_1\alpha_3 & \alpha_2\alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_1\beta_2 k_1 & -\alpha_3\alpha_4 - \beta_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\alpha_1\alpha_3 & -\beta_2\alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_1\beta_2 k_1 & -\alpha_3\alpha_4 - \beta_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\beta_1\alpha_3 & -\beta_2\alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \alpha_1\alpha_2 k_1 & \alpha_4\beta_3 + \alpha_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_1\beta_3 & \alpha_2\alpha_4 k_4 + \beta_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_6 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2\beta_1 k_1 & \alpha_4\beta_3 + \alpha_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_1\beta_3 & \alpha_2\alpha_4 k_4 + \beta_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_7 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_1\beta_2 k_1 & \alpha_4\beta_3 - \beta_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_1\beta_3 & \beta_3 c_1 k_3 - \beta_2\alpha_4 k_4 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_8 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_1\beta_2 k_1 & \alpha_4\beta_3 - \beta_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_1\beta_3 & \beta_3 c_1 k_3 - \beta_2\alpha_4 k_4 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_9 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \alpha_1\alpha_2 k_1 & \alpha_3\beta_4 + \alpha_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\alpha_1\alpha_3 & -\alpha_2\beta_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_{10} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2\beta_1 k_1 & \alpha_3\beta_4 + \alpha_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\beta_1\alpha_3 & -\alpha_2\beta_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix},$$

$$\mathbf{H}_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_1\beta_2k_1 & \alpha_3\beta_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\alpha_1\alpha_3 & \beta_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \end{bmatrix},$$

$$\mathbf{H}_{12} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_1\beta_2k_1 & \alpha_3\beta_4 - \beta_2k_2 & 0 & -\alpha_3c_2k_3 & \alpha_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\beta_1\alpha_3 & \beta_2\beta_4k_4 - \alpha_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \end{bmatrix},$$

$$\mathbf{H}_{13} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \alpha_1\alpha_2k_1 & \alpha_2k_2 - \beta_3\beta_4 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_1\beta_3 & \beta_3c_1k_3 - \alpha_2\beta_4k_4 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \end{bmatrix},$$

$$\mathbf{H}_{14} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2\beta_1k_1 & \alpha_2k_2 - \beta_3\beta_4 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_1\beta_3 & \beta_3c_1k_3 - \alpha_2\beta_4k_4 & 0 & \alpha_2c_2k_5 & -\alpha_2k_5 \end{bmatrix},$$

$$\mathbf{H}_{15} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_1\beta_2k_1 & -\beta_3\beta_4 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_1\beta_3 & \beta_2\beta_4k_4 + \beta_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \end{bmatrix},$$

$$\mathbf{H}_{16} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_1\beta_2k_1 & -\beta_3\beta_4 - \beta_2k_2 & 0 & \beta_3c_2k_3 & -\beta_3k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_1\beta_3 & \beta_2\beta_4k_4 + \beta_3c_1k_3 & 0 & -\beta_2c_2k_5 & \beta_2k_5 \end{bmatrix},$$

and let us denote $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$. Thus, the system of fuzzy rules which is equivalent to the model (3.29) is as follows

R_1: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_1 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_2: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & N_2 & N_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_2 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_3: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & P_2 & N_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_3 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_4: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & P_2 & N_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_4 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_5: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & N_2 & P_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_5 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_6: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & N_2 & P_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_6 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_7: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & P_2 & P_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_7 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_8: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & P_2 & P_3 & N_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_8 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_9: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & N_2 & N_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_9 \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{10}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & N_2 & N_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{10} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{11}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & P_2 & N_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{11} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{12}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & P_2 & N_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{12} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{13}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & N_2 & P_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{13} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{14}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & N_2 & P_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{14} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{15}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} N_1 & P_2 & P_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{15} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$,

R_{16}: If $[z_1, z_2, z_3, z_4]$ is $\begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix}$, then $\dot{\mathbf{x}} = \mathbf{H}_{16} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}$.

Now we will apply Theorem 3.7 on recurrence according to the equation (3.27). For the formerly defined membership functions $N_k(z_k)$ and $P_k(z_k)$, $(k = 1, 2, 3, 4)$, first we compute

$$\mathbf{S}_1(\mathbf{H}_1, \mathbf{H}_2) = N_1(z_1)\mathbf{H}_1 + P_1(z_1)\mathbf{H}_2$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2 k_1 z_1 & \alpha_2 k_2 - \alpha_3 \alpha_4 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & \alpha_2 \alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_3, \mathbf{H}_4) = N_1(z_1)\mathbf{H}_3 + P_1(z_1)\mathbf{H}_4$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_2 k_1 z_1 & -\alpha_3 \alpha_4 - \beta_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & -\beta_2 \alpha_4 k_4 - \alpha_3 c_1 k_3 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_5, \mathbf{H}_6) = N_1(z_1)\mathbf{H}_5 + P_1(z_1)\mathbf{H}_6$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2 k_1 z_1 & \alpha_4 \beta_3 + \alpha_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \alpha_2 \alpha_4 k_4 + \beta_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_7,\mathbf{H}_8) = N_1(z_1)\,\mathbf{H}_7 + P_1(z_1)\,\mathbf{H}_8$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_2 k_1 z_1 & \alpha_4\beta_3 - \beta_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \beta_3 c_1 k_3 - \beta_2\alpha_4 k_4 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_9,\mathbf{H}_{10}) = N_1(z_1)\,\mathbf{H}_9 + P_1(z_1)\,\mathbf{H}_{10}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2 k_1 z_1 & \alpha_3\beta_4 + \alpha_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & -\alpha_2\beta_4 k_4 - \alpha_3 c_1 k_3 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_{11},\mathbf{H}_{12}) = N_1(z_1)\,\mathbf{H}_{11} + P_1(z_1)\,\mathbf{H}_{12}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_2 k_1 z_1 & \alpha_3\beta_4 - \beta_2 k_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & \beta_2\beta_4 k_4 - \alpha_3 c_1 k_3 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_{13},\mathbf{H}_{14}) = N_1(z_1)\,\mathbf{H}_{13} + P_1(z_1)\,\mathbf{H}_{14}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\alpha_2 k_1 z_1 & \alpha_2 k_2 - \beta_3\beta_4 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \beta_3 c_1 k_3 - \alpha_2\beta_4 k_4 & 0 & \alpha_2 c_2 k_5 & -\alpha_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_1(\mathbf{H}_{15},\mathbf{H}_{16}) = N_1(z_1)\,\mathbf{H}_{15} + P_1(z_1)\,\mathbf{H}_{16}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \beta_2 k_1 z_1 & -\beta_3\beta_4 - \beta_2 k_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \beta_2\beta_4 k_4 + \beta_3 c_1 k_3 & 0 & -\beta_2 c_2 k_5 & \beta_2 k_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix}.$$

Next

$$\mathbf{S}_2(\mathbf{H}_1,\ldots,\mathbf{H}_4) = N_2(z_2)\,\mathbf{S}_1(\mathbf{H}_1,\mathbf{H}_2) + P_2(z_2)\,\mathbf{S}_1(\mathbf{H}_3,\mathbf{H}_4)$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & -\alpha_3\alpha_4 - k_2 z_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & -\alpha_3 c_1 k_3 - \alpha_4 k_4 z_2 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_2(\mathbf{H}_5,\ldots,\mathbf{H}_8) = N_2(z_2)\,\mathbf{S}_1(\mathbf{H}_5,\mathbf{H}_6) + P_2(z_2)\,\mathbf{S}_1(\mathbf{H}_7,\mathbf{H}_8)$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & \alpha_4\beta_3 - k_2 z_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \beta_3 c_1 k_3 - \alpha_4 k_4 z_2 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_2(\mathbf{H}_9,\ldots,\mathbf{H}_{12}) = N_2(z_2)\,\mathbf{S}_1(\mathbf{H}_9,\mathbf{H}_{10}) + P_2(z_2)\,\mathbf{S}_1(\mathbf{H}_{11},\mathbf{H}_{12})$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & \alpha_3\beta_4 - k_2 z_2 & 0 & -\alpha_3 c_2 k_3 & \alpha_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ g\alpha_3 z_1 & \beta_4 k_4 z_2 - \alpha_3 c_1 k_3 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_2(\mathbf{H}_{13},\ldots,\mathbf{H}_{16}) = N_2(z_2)\,\mathbf{S}_1(\mathbf{H}_{13},\mathbf{H}_{14}) + P_2(z_2)\,\mathbf{S}_1(\mathbf{H}_{15},\mathbf{H}_{16})$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & -\beta_3\beta_4 - k_2 z_2 & 0 & \beta_3 c_2 k_3 & -\beta_3 k_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g\beta_3 z_1 & \beta_3 c_1 k_3 + \beta_4 k_4 z_2 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

and

$$\mathbf{S}_3(\mathbf{H}_1,\ldots,\mathbf{H}_8) = N_3(z_3)\,\mathbf{S}_2(\mathbf{H}_1,\ldots,\mathbf{H}_4) + P_3(z_3)\,\mathbf{S}_2(\mathbf{H}_5,\ldots,\mathbf{H}_8)$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & \alpha_4 z_3 - k_2 z_2 & 0 & c_2 k_3 z_3 & -k_3 z_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g z_1 z_3 & c_1 k_3 z_3 - \alpha_4 k_4 z_2 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$\mathbf{S}_3(\mathbf{H}_9,\ldots,\mathbf{H}_{16}) = N_3(z_3)\,\mathbf{S}_2(\mathbf{H}_9,\ldots,\mathbf{H}_{12}) + P_3(z_3)\,\mathbf{S}_2(\mathbf{H}_{13},\ldots,\mathbf{H}_{16})$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & -\beta_4 z_3 - k_2 z_2 & 0 & c_2 k_3 z_3 & -k_3 z_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g z_1 z_3 & \beta_4 k_4 z_2 + c_1 k_3 z_3 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix}.$$

Finally we obtain

$$\mathbf{S}_4(\mathbf{H}_1,\ldots,\mathbf{H}_{16}) = N_4(z_4)\,\mathbf{S}_3(\mathbf{H}_1,\ldots,\mathbf{H}_8) + P_4(z_4)\,\mathbf{S}_3(\mathbf{H}_9,\ldots,\mathbf{H}_{16})$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ k_1 z_1 z_2 & -k_2 z_2 - z_3 z_4 & 0 & c_2 k_3 z_3 & -k_3 z_3 \\ 0 & 0 & 0 & 1 & 0 \\ -g z_1 z_3 & c_1 k_3 z_3 + k_4 z_2 z_4 & 0 & -c_2 k_5 z_2 & k_5 z_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix}.$$

The result is the same as formerly obtained, i.e. the fuzzy rules for the P1-TS system describe the inverted pendulum (3.29). The above model consists of 16 fuzzy rules.

Example 3.13. Consider the inverted pendulum from Example 3.12. On the basis of the above approach we can consider a two-dimensional system by neglecting the motion of the cart and the linear translation friction coefficient c_2. To ensure controllability of the system we assume $x_1(t) \in \left[-x_1^H, x_1^H\right]$, where $0 < x_1^H < \pi/2$. The angular velocity is bounded i.e. $x_2(t) \in \left[-x_2^H, x_2^H\right]$. Let us define new input variables for the P1-TS system: w_1, w_2 and w_3 as follows

$$w_1 = z_1 z_2 = \frac{1}{k_6 - \cos^2 x_1} \frac{\sin x_1}{x_1} \in [-a_1, b_1] = \left[\frac{1}{k_6 - \cos^2 x_1^H} \frac{\sin x_1^H}{x_1^H}, \frac{1}{k_6 - 1}\right], \tag{3.39}$$

$$w_2 = k_2 z_2 + z_3 z_4 = \frac{k_2 + \frac{x_2}{2} \sin 2x_1}{k_6 - \cos^2 x_1} \in [-a_2, b_2], \tag{3.40}$$

$$w_3 = k_3 z_3 = \frac{k_3 \cos x_1}{k_6 - \cos^2 x_1} \in [-a_3, b_3] = \left[\frac{k_3 \cos x_1^H}{k_6 - \cos^2 x_1^H}, \frac{k_3}{k_6 - 1}\right]. \tag{3.41}$$

The rough values of a_2 and b_2 can be easily evaluated, but in order to obtain the smallest interval for w_2, we should compute them numerically taking into account the maximal angular velocity x_2^H and the system constants (c_1, m, M and l). In general $a_2 \neq b_2$ as shown in Fig. 3.5. The membership functions

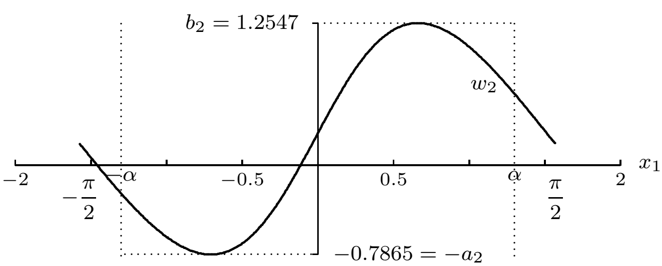


Fig. 3.5 Plot of the function w_2 for $x_2 = x_2^H = 5$, $c_1 = 0.05$, $m = 0.8$, $M = 1$ and $l = 0.5$, i.e. $w_2 = (0.5625 + 2.5 \sin 2x_1) / \left(3 - \cos^2 x_1\right)$

of fuzzy sets are $N_k(w_k)$ and $P_k(w_k) = 1 - N_k(w_k)$ by $w_k \in [-a_k, b_k]$ for $k = 1, 2, 3$. From the first two differential equations by $c_2 = 0$ we obtain

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ k_1 w_1 & -w_2 & -w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}. \tag{3.42}$$

Our goal is to obtain a TS model of the inverted pendulum in the form of fuzzy rules

$$\text{If } [w_1, w_2, w_3] \text{ is } \left[A_{i_1} \; A_{i_2} \; A_{i_3} \right], \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ q_{1,v} & q_{2,v} & q_{3,v} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix},$$

where $(A_{i_1}, A_{i_2}, A_{i_3}) \in \{N_1, P_1\} \times \{N_2, P_2\} \times \{N_3, P_3\}$. In accordance with the equations (3.42) and the generator $\mathbf{g} = \mathbf{g}(w_1, w_2, w_3)$ for $n = 3$ we define the following matrix of coefficients

$$\mathbf{\Theta} = \begin{bmatrix} 0 & 0 & 0 \\ k_1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Thus, from (2.52) and (2.41) the consequents matrix is computed as

$$\mathbf{Q} = \mathbf{\Omega}^T \mathbf{\Theta} = \begin{bmatrix} -a_1 k_1 & a_2 & a_3 \\ b_1 k_1 & a_2 & a_3 \\ -a_1 k_1 & -b_2 & a_3 \\ b_1 k_1 & -b_2 & a_3 \\ -a_1 k_1 & a_2 & -b_3 \\ b_1 k_1 & a_2 & -b_3 \\ -a_1 k_1 & -b_2 & -b_3 \\ b_1 k_1 & -b_2 & -b_3 \end{bmatrix}.$$

The exact model of the 2-dimensional inverted pendulum by $c_2 = 0$ we can write in the compact form

$$\text{If } [w_1, w_2, w_3] \text{ is } \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ P_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix}, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a_1 k_1 & a_2 & a_3 \\ b_1 k_1 & a_2 & a_3 \\ -a_1 k_1 & -b_2 & a_3 \\ b_1 k_1 & -b_2 & a_3 \\ -a_1 k_1 & a_2 & -b_3 \\ b_1 k_1 & a_2 & -b_3 \\ -a_1 k_1 & -b_2 & -b_3 \\ b_1 k_1 & -b_2 & -b_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}.$$

Let us observe that there are many ways in which we can choose the artificial variables as the inputs for the TS system. For example the first two equations in (3.29) by $c_2 = 0$ can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ z_1 & z_2 & z_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}$$

where

$$z_1 = \frac{k_1 \dfrac{\sin x_1}{x_1} - \dfrac{1}{4} x_2^2 \dfrac{\sin(2x_1)}{x_1}}{k_6 - \cos^2 x_1},$$

$$z_2 = -\frac{k_2 + \dfrac{x_2 \sin(2x_1)}{4}}{k_6 - \cos^2 x_1},$$

$$z_3 = -\frac{k_3 \cos x_1}{k_6 - \cos^2 x_1}.$$

The next steps for computing the conclusions of the fuzzy rules are the same as before.

As one can see, only 8 fuzzy rules for the P1-TS system are sufficient to obtain an exact model of the angular motion of the pendulum in the two-dimensional case.

Example 3.14. From the practical point of view an approximation model may be quite sufficient. The pendulum nonlinearities are simple because the nonlinear functions depend on one variable x_1. Below we propose to consider a truncated Taylor series expansion of the second and fourth equation in (3.29) with respect to the state variable x_1, about the point $x_1 = 0$, up to some number of terms, say 4. The 4th-degree Taylor polynomials should guarantee a sufficiently high accuracy; otherwise we should take the Taylor polynomials of the higher degree. Thus,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix}, \tag{3.43}$$

where the matrix $\mathbf{H}$ is given by

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ h_{21}(w_1) & h_{22}(w_1, w_2) & 0 & h_{24}(w_1) & h_{25}(w_1) \\ 0 & 0 & 0 & 1 & 0 \\ h_{41}(w_1) & h_{42}(w_1, w_2) & 0 & h_{44}(w_1) & h_{45}(w_1) \end{bmatrix}. \tag{3.44}$$

We use two new artificial variables $w_1 = x_1^2$ and $w_2 = x_1 x_2$. The elements of the matrix $\mathbf{H}$ are as follows

$$h_{21}(w_1) = a_1 w_1 + b_1 + O\left(x_1^4\right),$$

$$h_{22}(w_1, w_2) = h_0 + h_1 w_1 + h_2 w_2 + h_3 w_1 w_2 + O\left(x_1^4\right),$$
$$h_{24}(w_1) = a_2 w_1 + b_2 + O\left(x_1^4\right),$$
$$h_{25}(w_1) = a_3 w_1 + b_3 + O\left(x_1^4\right),$$
$$h_{41}(w_1) = a_4 w_1 + b_4 + O\left(x_1^4\right),$$
$$h_{42}(w_1, w_2) = r_0 + r_1 w_1 + r_2 w_2 + r_3 w_1 w_2 + O\left(x_1^4\right),$$
$$h_{44}(w_1) = a_5 w_1 + b_5 + O\left(x_1^4\right),$$
$$h_{45}(w_1) = a_6 w_1 + b_6 + O\left(x_1^4\right),$$

and the constants are

$$a_1 = -ak_1(a + 1/6), \quad b_1 = ak_1, \quad a_2 = -ac_2k_3(2a+1)/2, \quad b_2 = ac_2k_3,$$
$$a_3 = ak_3(2a+1)/2, \quad b_3 = -ak_3, \quad a_4 = ag(a + 2/3), \quad b_4 = -ag,$$
$$a_5 = a^2c_2k_3k_4, \quad b_5 = -ac_2k_3k_4, \quad a_6 = -a^2k_3k_4, \quad b_6 = ak_3k_4,$$

and $a = 1/(k_6 - 1)$. Observe that all elements of the matrix $\mathbf{H}$ are functions of variables w_1 and w_2 of the form (2.26) and therefore the system (3.43) can be exactly modeled by a P1-TS rule-based system. Our goal is to obtain the TS rule-based system of the inverted pendulum in the following form

$$\text{If } [w_1, w_2] \text{ is } \left[A_{i_1} \; A_{i_2}\right], \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ q_{1,v} & q_{2,v} & 0 & q_{4,v} & q_{5,v} \\ 0 & 0 & 0 & 1 & 0 \\ r_{1,v} & r_{2,v} & 0 & r_{4,v} & r_{5,v} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

where $(A_{i_1}, A_{i_2}) \in \{N_1, P_1\} \times \{N_2, P_2\}$. The inputs of the P1-TS system are w_1 and w_2 as the points of the rectangle D^2, whereas the outputs of this system are the velocities of the state vector, i.e. $\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$ and $\dot{x}_4$. The membership functions of fuzzy sets are $N_k(w_k) = (\alpha_k + \beta_k)^{-1}(\beta_k - w_k)$ and $P_k(w_k) = 1 - N_k(w_k)$, where α_k and β_k, $(k = 1, 2)$, can be easily obtained

$$w_1 \in [-\alpha_1, \beta_1] = \left[0, \left(x_1^H\right)^2\right], \quad w_2 \in [-\alpha_2, \beta_2] = \left[-x_1^H x_2^H, x_1^H x_2^H\right].$$

The values x_1^H and x_2^H are the maximal angle and its maximal speed, respectively. Thus, we obtain highly interpretable membership functions of the fuzzy sets

$$N_1(w_1) = 1 - \frac{w_1}{\beta_1} = 1 - \left(\frac{x_1}{x_1^H}\right)^2,$$

$$N_2(w_2) = \frac{1}{2}\left(1 - \frac{w_2}{\alpha_2}\right) = \frac{1}{2}\left(1 - \frac{x_1 x_2}{x_1^H x_2^H}\right).$$

According to the format of the rules, in order to find $q_{i,v}$ and $r_{i,v}$ we should take into account the second and the fourth row of the matrix (3.44), correspondingly. For the generator $\mathbf{g}$ given by (2.37) we define the following matrices that define the functions of the inputs w_1 and w_2 of the TS system

$$\boldsymbol{\Theta}_q = \begin{bmatrix} b_1 & h_0 & 0 & b_2 & b_3 \\ a_1 & h_1 & 0 & a_2 & a_3 \\ 0 & h_2 & 0 & 0 & 0 \\ 0 & h_3 & 0 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{\Theta}_r = \begin{bmatrix} b_4 & r_0 & 0 & b_5 & b_6 \\ a_4 & r_1 & 0 & a_4 & a_6 \\ 0 & r_2 & 0 & 0 & 0 \\ 0 & r_3 & 0 & 0 & 0 \end{bmatrix}.$$

From (2.38) we have $\mathbf{Q}_q = \boldsymbol{\Omega}^T \boldsymbol{\Theta}_q$ and $\mathbf{Q}_r = \boldsymbol{\Omega}^T \boldsymbol{\Theta}_r$. Thus,

$$\mathbf{Q}_q = \begin{bmatrix} b_1 - \alpha_1 a_1 & h_0 - \alpha_1 h_1 - \alpha_2 h_2 + \alpha_1 \alpha_2 h_3 & 0 & b_2 - \alpha_1 a_2 & b_3 - \alpha_1 a_3 \\ b_1 + \beta_1 a_1 & h_0 + \beta_1 h_1 - \alpha_2 h_2 - \alpha_2 \beta_1 h_3 & 0 & b_2 + \beta_1 a_2 & b_3 + \beta_1 a_3 \\ b_1 - \alpha_1 a_1 & h_0 - \alpha_1 h_1 + \beta_2 h_2 - \alpha_1 \beta_2 h_3 & 0 & b_2 - \alpha_1 a_2 & b_3 - \alpha_1 a_3 \\ b_1 + \beta_1 a_1 & h_0 + \beta_1 h_1 + \beta_2 h_2 + \beta_1 \beta_2 h_3 & 0 & b_2 + \beta_1 a_2 & b_3 + \beta_1 a_3 \end{bmatrix},$$

$$\mathbf{Q}_r = \begin{bmatrix} b_4 - \alpha_1 a_4 & r_0 - \alpha_1 r_1 - \alpha_2 r_2 + \alpha_1 \alpha_2 r_3 & 0 & b_5 - \alpha_1 a_4 & b_6 - \alpha_1 a_6 \\ b_4 + \beta_1 a_4 & r_0 + \beta_1 r_1 - \alpha_2 r_2 - \alpha_2 \beta_1 r_3 & 0 & b_5 + \beta_1 a_4 & b_6 + \beta_1 a_6 \\ b_4 - \alpha_1 a_4 & r_0 - \alpha_1 r_1 + \beta_2 r_2 - \alpha_1 \beta_2 r_3 & 0 & b_5 - \alpha_1 a_4 & b_6 - \alpha_1 a_6 \\ b_4 + \beta_1 a_4 & r_0 + \beta_1 r_1 + \beta_2 r_2 + \beta_1 \beta_2 r_3 & 0 & b_5 + \beta_1 a_4 & b_6 + \beta_1 a_6 \end{bmatrix}.$$

Finally, the system of rules that models exactly the system (3.43) by the given matrix $\mathbf{H}$ is as follows

$$R_1: \text{ If } [w_1, w_2] \text{ is } \begin{bmatrix} N_1 \; N_2 \end{bmatrix}, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \mathbf{H}_1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$R_2: \text{ If } [w_1, w_2] \text{ is } \begin{bmatrix} P_1 \; N_2 \end{bmatrix}, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$R_3: \text{ If } [w_1, w_2] \text{ is } \begin{bmatrix} N_1 \; P_2 \end{bmatrix}, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \mathbf{H}_3 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

$$R_4: \text{ If } [w_1, w_2] \text{ is } \begin{bmatrix} P_1 & P_2 \end{bmatrix}, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \mathbf{H}_4 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix},$$

where

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ b_1 - \alpha_1 a_1 & h_0 - \alpha_1 h_1 - \alpha_2 h_2 + \alpha_1 \alpha_2 h_3 & 0 & b_2 - \alpha_1 a_2 & b_3 - \alpha_1 a_3 \\ 0 & 0 & 0 & 1 & 0 \\ b_4 - \alpha_1 a_4 & r_0 - \alpha_1 r_1 - \alpha_2 r_2 + \alpha_1 \alpha_2 r_3 & 0 & b_5 - \alpha_1 a_4 & b_6 - \alpha_1 a_6 \end{bmatrix},$$

$$\mathbf{H}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ b_1 + \beta_1 a_1 & h_0 + \beta_1 h_1 - \alpha_2 h_2 - \alpha_2 \beta_1 h_3 & 0 & b_2 + \beta_1 a_2 & b_3 + \beta_1 a_3 \\ 0 & 0 & 0 & 1 & 0 \\ b_4 + \beta_1 a_4 & r_0 + \beta_1 r_1 - \alpha_2 r_2 - \alpha_2 \beta_1 r_3 & 0 & b_5 + \beta_1 a_4 & b_6 + \beta_1 a_6 \end{bmatrix},$$

$$\mathbf{H}_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ b_1 - \alpha_1 a_1 & h_0 - \alpha_1 h_1 + \beta_2 h_2 - \alpha_1 \beta_2 h_3 & 0 & b_2 - \alpha_1 a_2 & b_3 - \alpha_1 a_3 \\ 0 & 0 & 0 & 1 & 0 \\ b_4 - \alpha_1 a_4 & r_0 - \alpha_1 r_1 + \beta_2 r_2 - \alpha_1 \beta_2 r_3 & 0 & b_5 - \alpha_1 a_4 & b_6 - \alpha_1 a_6 \end{bmatrix},$$

$$\mathbf{H}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ b_1 + \beta_1 a_1 & h_0 + \beta_1 h_1 + \beta_2 h_2 + \beta_1 \beta_2 h_3 & 0 & b_2 + \beta_1 a_2 & b_3 + \beta_1 a_3 \\ 0 & 0 & 0 & 1 & 0 \\ b_4 + \beta_1 a_4 & r_0 + \beta_1 r_1 + \beta_2 r_2 + \beta_1 \beta_2 r_3 & 0 & b_5 + \beta_1 a_4 & b_6 + \beta_1 a_6 \end{bmatrix}.$$

Assume as in [184] a large value of x_1^H, say $x_1^H = 88\pi/180$ [deg]. In order to have a good interpretation of real situations which are modeled by the rule-based system we should observe that the subsets of the arguments $(x_1, x_2) \in D^2$, for which the functions used in the premises of the rules are maximal, are as follows

- the premise function $N_1 N_2$ takes its maximal value (16/27) at $(x_1, x_2) \in \left\{ \left(x_1^H/3, -x_2^H \right), \left(-x_1^H/3, x_2^H \right) \right\}$,
- the premise function $P_1 N_2$ takes its maximal value (1) at $(x_1, x_2) \in \left\{ \left(-x_1^H, x_2^H \right), \left(x_1^H, -x_2^H \right) \right\}$,
- the premise function $N_1 P_2$ takes its maximal value (16/27) at $(x_1, x_2) \in \left\{ \left(x_1^H/3, x_2^H \right), \left(-x_1^H/3, -x_2^H \right) \right\}$,
- the premise function $P_1 P_2$ takes its maximal value (1) at $(x_1, x_2) \in \left\{ \left(x_1^H, x_2^H \right), \left(-x_1^H, -x_2^H \right) \right\}$.

Thus, the premises of the rules $R_1 - R_4$ can be substituted by the above description of situations.

In the above example we used 4th-degree Taylor polynomials for approximation of nonlinear functions at the equilibrium point of the dynamical system. One can check that the above result is much better than the one obtained by the linearization of differential equations around the equilibrium. Thanks to the Taylor series expansion we obtained a small number of highly interpretable fuzzy rules.

3.5 Boundedness of P1-TS Systems

In this section we evaluate the lower and upper bound of the function (2.26) to which the rule-based P1-TS system is equivalent.

Theorem 3.15. *The crisp output S of every MISO P1-TS system with inputs $z_k \in [-\alpha_k, \beta_k]$ for $k = 1, \ldots, n$, defined by "If-then" rules (2.32) with consequents of the rules q_v, $v = 1, 2, \ldots, 2^n$, is bounded. The lower and upper bounds are independent of boundaries of the intervals $-\alpha_k$ or β_k, and they are given by*

$$\min\{q_1, q_2, \ldots, q_{2^n}\} \leq S \leq \max\{q_1, q_2, \ldots, q_{2^n}\}. \tag{3.45}$$

The above assessment cannot be improved.

Proof. According to Theorem 3.6, for $n = 1$, by any arbitrarily chosen z_1 from the interval $[-\alpha_1, \beta_1]$, the system output is the convex combination of q_1 and q_2, as expressed by (3.11). Thus,

$$\min\{q_1, q_2\} \leq S(z_1 \mid q_1, q_2) = \lambda_1 q_1 + (1 - \lambda_1) q_2 \leq \max\{q_1, q_2\}$$

holds, since $\lambda_1 = (\alpha_1 + \beta_1)^{-1}(\beta_1 - z_1) \in [0, 1]$ for every $z_1 \in [-\alpha_1, \beta_1]$. Suppose that the theorem is true for the system with input variables $z_1, \ldots, z_{n-1}$ and the output S_{n-1}. This means that S_{n-1} in (3.10) both by parameters $q_1, \ldots, q_{2^{n-1}}$, and by parameters $q_{2^{n-1}+1}, \ldots, q_{2^n}$, as a function of independent variables $z_1, \ldots, z_n$ is bounded as follows

$$\min\{q_1, \ldots, q_{2^{n-1}}\} \leq S_{n-1}(z_1, \ldots, z_{n-1} \mid q_1, \ldots, q_{2^{n-1}}) \leq \max\{q_1, \ldots, q_{2^{n-1}}\}, \tag{3.46}$$

and

$$\begin{aligned}\min\{q_{2^{n-1}+1}, \ldots, q_{2^n}\} &\leq S_{n-1}(z_1, \ldots, z_{n-1} \mid q_{2^{n-1}+1}, \ldots, q_{2^n}) \\ &\leq \max\{q_{2^{n-1}+1}, \ldots, q_{2^n}\}.\end{aligned} \tag{3.47}$$

But

$$\min\{\min\{q_1, \ldots, q_{2^{n-1}}\}, \min\{q_{2^{n-1}+1}, \ldots, q_{2^n}\}\} = \min\{q_1, \ldots, q_{2^n}\},$$

and

$$\max\{\max\{q_1, \ldots, q_{2^{n-1}}\}, \max\{q_{2^{n-1}+1}, \ldots, q_{2^n}\}\} = \max\{q_1, \ldots, q_{2^n}\}.$$

Thus,

$$\min\{q_1,\ldots,q_{2^n}\} \le S_{n-1}(z_1,\ldots,z_{n-1}) \le \max\{q_1,\ldots,q_{2^n}\}.$$

According to (3.10) by any arbitrarily chosen input vector $[z_1,\ldots,z_n]^T \in D^n$, the system output $S = S(z_1,\ldots,z_n \mid q_1,\ldots,q_{2^n})$ is the convex combination of $S_{n-1}(z_1,\ldots,z_{n-1} \mid q_1,\ldots,q_{2^{n-1}})$ and $S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n})$, which both are bounded by $\min\{q_1,\ldots,q_{2^n}\}$ and $\max\{q_1,\ldots,q_{2^n}\}$:

$$\begin{aligned}\min\{q_1,\ldots,q_{2^n}\} \le S &= \lambda_n S_{n-1}(z_1,\ldots,z_{n-1} \mid q_1,\ldots,q_{2^{n-1}}) \\ &\quad + (1-\lambda_n) S_{n-1}(z_1,\ldots,z_{n-1} \mid q_{2^{n-1}+1},\ldots,q_{2^n}) \\ &\le \max\{q_1,\ldots,q_{2^n}\},\end{aligned}$$

since $\lambda_n = (\alpha_n+\beta_n)^{-1}(\beta_n - z_n) \in [0,1]$ for $z_n \in [-\alpha_n,\beta_n]$. Finally, observe that for every consequent q_v of the rule, there exists a vertex in the hyper-cuboid D^n such, that the output $S = q_v$ (see the Proof of Theorem 2.4). Thus, both the lower, and upper bounds in (3.45) cannot be improved. This ends the proof of Theorem 3.15. □

3.6 Summary

Thanks to Theorem 3.1 the computation of the inverse of the fundamental matrix does not need classical matrix inversion procedures; the matrix inverse can be found recursively using multiplication operations only. Some features of the fundamental matrix and its inverse were given as well.

The decomposition of the rule-base of the P1-TS system that uses two complementary fuzzy sets for every input, was helpful in developing the recursive procedure. Thanks to Theorem 3.6 and its generalization the curse of dimensionality problem in the rule-based systems can be substantially reduced. Recursive procedure can be used for rather large rule-bases, even if the membership functions of fuzzy sets are not linear or monotonic. The membership functions were assumed to be complementary, but this requirement is not very restrictive, since in practice the membership functions can always be normalized. In the case of the P1-TS system with n-iputs, the computational architecture of the recursion seems to be simple and can be viewed as a feedforward n-layer neural network. As an example of application of the recursion, the rule-based system with 6 inputs was considered. It is not a big problem to consider a P1-TS system with about 10 inputs; such examples require only more space to write equations.

It was shown that P1-TS systems can be used for the exact modeling of the nonlinear continuous or discrete-time dynamical systems, where the inputs of the fuzzy rule-based system are more abstract quantities and the outputs refer to the system structure. The system inputs can contain known premise variables that are not functions of the control input, but they may be functions of the state variables, external disturbances and/or time. For every input variable we assumed two complementary membership functions that cannot be monotonic or linear. For inverted pendulum system we obtained better results than in other works; e.g. only 8 fuzzy rules (instead of 16) for the P1-TS system turned out to be sufficient to model exactly the angular motion of the pendulum in the two-dimensional case. However, with the above approach there are many ways in which the artificial variables as the TS system inputs can be defined. Another disadvantage of the above approach which coincides in many respects with the one described in [184] is that the fuzzy rules are not simple for interpretation. By using recurrence we can easily check validity of other models of nonlinear systems in the P1-TS form, e.g. a translational oscillator with an eccentric rotational proof mass actuator, a vehicle with triple trailers described in [184] and many other dynamical systems.

We showed that application of the Taylor series expansion can be very attractive in practice. In one example we used 4th-degree Taylor polynomials for a good approximation of nonlinear functions at the equilibrium point of the dynamical system. The result is much better than the one obtained by the linearization of differential equations around the equilibrium. By using the Taylor series expansion we obtained a small number of highly interpretable fuzzy rules.

Finally we found the best evaluation for the lower and upper bound of the function, to which the rule-based P1-TS system is equivalent.

Chapter 4
Fuzzy Rule-Based Systems with Polynomial Membership Functions

In order to obtain a richer class of functions to which the fuzzy rule-based system is equivalent, one can use nonlinear membership functions of fuzzy sets, to which *polynomials* of the second or higher degree belong. Such polynomials are defined by three or more parameters. It would appear that by using nonlinear membership functions, one can get a sufficiently large class of functions, to which the rule-based system is equivalent. However, if we increase the complexity of membership functions of fuzzy sets only, while preserving the number of fuzzy sets assigned for the input variables, our intuition about richness of the class of functions performed by the rule-based system can fail us. The number of fuzzy sets is important, since it determines the number of consequents of the rules; thus, it constrains the class of functions performed by the zero-order TS rule-based systems. This fact will be shown further on.

The consequents of "If-then" rules can be defined as functions depending on input variables, e.g. they can be polynomials. However, if it is not stated differently, we will consider the zero-order rule-based systems. A special attention will be paid to the TS systems which use the second degree polynomials as the membership functions of fuzzy sets. First we will show that it is not possible to obtain any second degree polynomial function, to which a TS rule-based system is equivalent, on the assumption that only two complementary membership functions as the second degree polynomials are defined for the input variables for this system. We prove however, that three quadratic membership functions suffice to model every second degree polynomial function. For such membership functions the natural requirements that guarantee a clear interpretability of fuzzy sets will be defined as well. The TS systems that use as a basis three normalized second degree polynomial membership functions, called *P2-TS systems*, will be thoroughly investigated. Similarly to the fuzzy rule-based systems with linear membership functions, we will define both a generator and a fundamental matrix for the P2-TS systems. The features of the fundamental matrix for such systems and its inverse will be given.

The curse of dimensionality problem is more serious for the P2-TS systems than the one for the P1-TS systems. Therefore we will develop the recursive procedures for the computation of the inverse of the fundamental matrix and for the crisp output of the P2-TS systems.

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 61–99.
springerlink.com

4.1 TS Systems with Two Polynomial Membership Functions for Every Input

Below we prove the following

Remark 4.1. Suppose the inputs of a zero-order TS system are $z_k \in [-\alpha_k, \beta_k]$, $(k = 1, 2, \ldots, n)$, and every input has assigned two complementary membership functions, say $N_k(z_k)$ and $P_k(z_k) = 1 - N_k(z_k)$. If all membership functions are polynomials of the degree d, then

(1) the crisp output $f(z_1, \ldots, z_n)$ of this system is the following multivariate polynomial

$$f(z_1, \ldots, z_n) = \sum_{p_1, \ldots, p_n \in \{0,1,2,\ldots,d\}^n} \theta_{p_1,\ldots,p_n} z_1^{p_1} z_2^{p_2} \cdots z_n^{p_n}, \tag{4.1}$$

where $\theta_{p_1,\ldots,p_n} \in \mathbb{R}$,
(2) every multilinear function of type (4.1), can be exactly expressed by the fuzzy "If-then" rules if, and only if the degree of polynomials is $d = 1$,
(3) not every nonlinear function of type (4.1) can be unambiguously expressed by the fuzzy "If-then" rules, when the degree $d > 1$.

Proof.

(1) First observe that the system output S is a linear combination of 2^n polynomials in the form "$\prod_{k=1}^{n} \left(a_{d,k} z_k^d + \ldots + a_{1,k} z_k + a_{0,k}\right)$". Thus, the output S is in the form (4.1), indeed.
(2) For two fuzzy sets for every input (N_k and P_k), there are 2^n consequents of the rules, which are free design parameters. The polynomial of degree d is described by $(d+1)$ parameters. Thus, the number of functions (4.1), which are structurally different one from another, is $(d+1)^n$, and it is equal to the number of different consequents of the rules if, and only if $(d+1)^n = 2^n$. In this case we apply Theorem 2.4.
(3) For $d \geq 2$ we have $(d+1)^n > 2^n$. Thus, not every nonlinear function (4.1) can be exactly expressed by TS system; this finishes the proof of Remark 4.1. □

Let us consider an example which is of twofold goal. Firstly, we will give an additional proof of Remark 4.1 for the second degree polynomial ($d = 2$). Secondly, we will show that by using some nonlinear bijection for the crisp input x of the TS system with two linear membership functions, we can obtain its nonlinear output $S(x)$, (see Fig. 4.1). Of course, the use of such bijection is not necessary to prove Remark 4.1.

Example 4.2. Let us consider the zero-order TS system with the input x and the output S, as shown in Fig. 4.1. We define a nonlinear mapping between the original input $x \in [-\alpha, \beta]$ and an ancillary variable $z \in [-\alpha, \beta]$, in the form of the second order polynomial

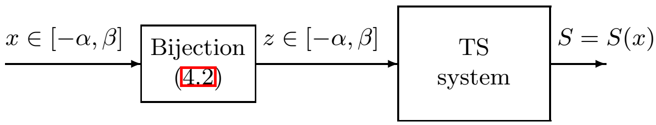


Fig. 4.1 SISO TS system from Example 4.2

$$z(x) = x + m\frac{(x+\alpha)(x-\beta)}{\alpha+\beta}, \tag{4.2}$$

where m is a parameter - see Fig. 4.2.We assume that $0 \neq |m| < 1$, since (4.2) is a bijection $z : [-\alpha, \beta] \to [-\alpha, \beta]$ if, and only if $|m| < 1$, and we omit the trivial case $z = x$. If the membership functions are linear:

$$N(z) = (\alpha+\beta)^{-1}(\beta - z), \qquad P(z) = 1 - N(z),$$

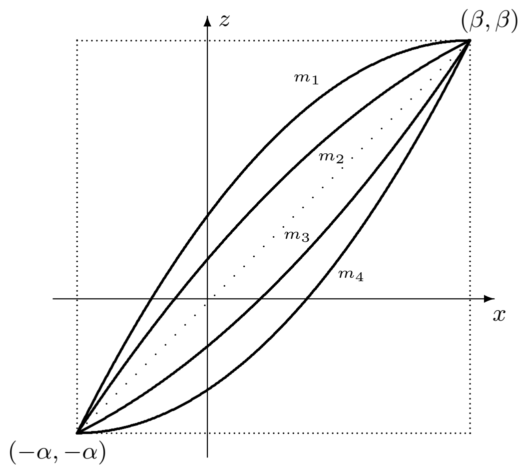
then from two fuzzy rules:

$$\left.\begin{array}{l} R_1 : \text{If } z \text{ is } N, \text{ then } S = q_1, \\ R_2 : \text{If } z \text{ is } P, \text{ then } S = q_2, \end{array}\right\}$$

we obtain the system output

$$\begin{aligned} S &= \frac{q_1 N(z) + q_2 P(z)}{N(z) + P(z)} \\ &= \frac{(q_2 - q_1)x + \alpha q_2 + \beta q_1}{\alpha + \beta} + m\frac{(q_2 - q_1)(x+\alpha)(x-\beta)}{(\alpha+\beta)^2}. \end{aligned} \tag{4.3}$$

Fig. 4.2 The bijection (4.2) with parameter $m = m_i$: $m_1 = -1$, $m_2 = -0.5$, $m_3 = 0.5$ and $m_4 = 1$



It can be equivalently written as

$$S(x) = Ax^2 + Bx + C,$$

where

$$A = m\frac{q_2 - q_1}{(\alpha+\beta)^2},$$

$$B = \frac{(\alpha+\beta+m(\alpha-\beta))(q_2-q_1)}{(\alpha+\beta)^2},$$

$$C = \frac{q_1\beta(\alpha+\beta+m\alpha)+q_2\left(\alpha\beta-m\alpha\beta+\alpha^2\right)}{(\alpha+\beta)^2}.$$

Thus, independently of the consequents of the rules (q_1 and q_2), the system output is restricted to the following class of functions as second degree polynomials

$$S(x) = Ax^2 + \left(\frac{\alpha+\beta}{m}+\alpha-\beta\right)Ax + C, \qquad x \in [-\alpha,\beta], \tag{4.4}$$

where $A, C \in \mathbb{R}$, by $1 > |m| \neq 0$. This means that there are "many", but not all second degree polynomials, which can be exactly represented by the rule-based system. For example, by the fixed interval $[-\alpha, \beta]$, we are not able to formulate such two fuzzy rules, that the rule-based system would be equivalent to the following polynomial

$$f(x) = Ax^2 + A(\alpha-\beta)x + C, \qquad x \in [-\alpha,\beta], \tag{4.5}$$

where $A, C \in \mathbb{R}$. This is because there is no m such that $0 \neq |m| < 1$ and $\left(\frac{\alpha+\beta}{m}+\alpha-\beta\right)A = (\alpha-\beta)A$ for any real α, β and A. In other words, the function (4.5) is not from the class of functions defined by (4.4). This example shows by contradiction that the second part of Remark 4.1 is true.

The zero-order rule-based TS systems in which the membership functions of input variables are polynomials of the degree d will be called *Pd-TS systems*. A special attention will be paid to *P2-TS systems* further on.

4.2 The Normalized Membership Functions for P2-TS Systems

From the preceding section we know that it is not possible to obtain any second degree polynomial by using the TS systems, in which only two complementary membership functions as second degree polynomials are defined. However, we will prove further on that three membership functions as the second degree polynomials suffice to model any second degree polynomial function. Such membership functions defining the fuzzy sets for input variables will be defined below.

In the interval $[-\alpha, \beta]$ we define three membership functions of fuzzy sets, say $N(z)$, $Z(z)$ and $P(z)$, which are the second degree polynomials and satisfy the following additional conditions:

1. $N : [-\alpha, \beta] \to [0, 1]$ is a monotonic function with *negative slope*, i.e. $dN(z)/dz < 0$ for $z \in [-\alpha, \beta]$, which satisfies two boundary conditions:
   a) $N(-\alpha) = 1$,
   b) $N(\beta) = 0$.
2. $P : [-\alpha, \beta] \to [0, 1]$ is the monotonic function with *positive slope*, i.e. $dP(z)/dz > 0$ for $z \in [-\alpha, \beta]$, symmetric to the function N with respect to the interval centre $\sigma \in [-\alpha, \beta]$:
$$\sigma = \frac{-\alpha + \beta}{2}. \tag{4.6}$$
3. $Z : [-\alpha, \beta] \to [0, 1]$ is the function which reaches *zero slope* in σ, i.e. $dZ(\sigma)/dz = 0$.
4. The functions N, Z and P satisfy the *normalization condition*
$$N(z) + Z(z) + P(z) = 1, \qquad \forall\, z \in [-\alpha, \beta]. \tag{4.7}$$

One can prove that the functions N, Z and P meeting the above needs can be expressed as follows

$$N(z) = \frac{(\alpha + \beta - \lambda(z + \alpha))(\beta - z)}{(\alpha + \beta)^2}, \tag{4.8}$$

$$Z(z) = 2\lambda \frac{(\beta - z)(z + \alpha)}{(\alpha + \beta)^2}, \tag{4.9}$$

$$P(z) = \frac{(\alpha + \beta + \lambda(z - \beta))(z + \alpha)}{(\alpha + \beta)^2}, \tag{4.10}$$

where the parameter λ satisfies the condition

$$0 < \lambda \leq 1. \tag{4.11}$$

We do not allow $\lambda = 0$, since in such case $Z(z) = 0$ for all z, and there would be two nonzero membership functions only: $N(z)$ and $P(z)$. In other words, by $\lambda = 0$, the class of rule-based systems reduces to the formerly considered P1-TS systems. Figures 4.3 and 4.4 show plots of functions (4.8)-(4.10) for different values of parameter λ. Observe that N and P are normal fuzzy sets but Z is not normal. The cores of the fuzzy sets N, Z and P are three *characteristic points* of the universe of discourse: "$-\alpha$", "σ" and "β", respectively.

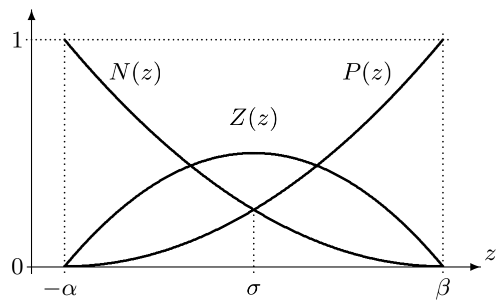


Fig. 4.3 The basis of normalized second degree polynomial membership functions by the maximal value of parameter λ, ($\lambda = 1$)

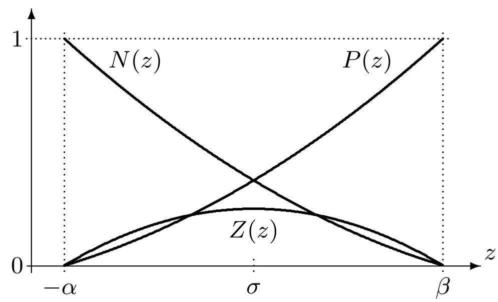


Fig. 4.4 The basis of normalized second degree polynomial membership functions by parameter $\lambda = 0.5$

The membership functions N, Z and P have a clear linguistic interpretation in any case of boundaries "$-\alpha$" and "β" as real numbers:

1. If $-\alpha < \beta < 0$, then N can be interpreted as *negative big*, Z - *negative medium* and P - *negative small*,
2. If $-\alpha < \beta = 0$, then N can be interpreted as *negative*, Z - *negative small* and P - *negative zero*,
3. If $-\alpha < 0 < \beta$, then N can be interpreted as *negative*, Z - *zero* and P - *positive*,
4. If $0 = -\alpha < \beta$, then N can be interpreted as *positive zero*, Z - *positive small* and P - *positive*,
5. If $0 < -\alpha < \beta$, then N can be interpreted as *positive small*, Z - *positive medium* and P - *positive big*.

As discussed in Section 2.2, the linguistic terms can be substituted by others depending on the context or specific application.

The rule-based TS systems with the above membership functions we will call *P2-TS systems* for short.

4.3 SISO P2-TS System

Now we will consider P2-TS system with single input $z \in [-\alpha, \beta]$ and single output S. The rule-base structure is as follows

$$\left.\begin{array}{l} R_1 : \text{If } z \text{ is } N, \text{ then } S = q_0, \\ R_2 : \text{If } z \text{ is } Z, \text{ then } S = q_1, \\ R_3 : \text{If } z \text{ is } P, \text{ then } S = q_2. \end{array}\right\} \tag{4.12}$$

The system output as a function of the input variable z is given by

$$S(z) = N(z)\, q_0 + Z(z)\, q_1 + P(z)\, q_2 = [N(z), Z(z), P(z)]\, \mathbf{q}, \tag{4.13}$$

where $\mathbf{q} = [q_0, q_1, q_2]^T$, and N, Z and P are defined in (4.8)-(4.10). By $\mathbf{s}$ we denote the vector containing values of system output in the cores of the fuzzy sets N, Z and P, respectively

$$\mathbf{s} = [S(-\alpha), S(\sigma), S(\beta)]^T .$$

It can be expressed equivalently by

$$\mathbf{s} = \mathbf{R}\mathbf{q}, \tag{4.14}$$

where the matrix $\mathbf{R}$ contains the membership degrees in the cores of the fuzzy sets

$$\mathbf{R} = \begin{bmatrix} N(-\alpha) & Z(-\alpha) & P(-\alpha) \\ N(\sigma) & Z(\sigma) & P(\sigma) \\ N(\beta) & Z(\beta) & P(\beta) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ (2-\lambda)/4 & \lambda/2 & (2-\lambda)/4 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.15}$$

Observe that $S(-\alpha) = q_0$ and $S(\beta) = q_2$. However, the consequent of the fuzzy rule R_2 in (4.12) is q_1, but

$$S(\sigma) = q_1' = \frac{2-\lambda}{4} q_0 + \frac{\lambda}{2} q_1 + \frac{2-\lambda}{4} q_2 \neq q_1,$$

and there is no such $\lambda \in (0, 1]$ for which q_1' would be equal to q_1. The maximal influence of the rule consequent q_1 for the crisp output q_1' one obtains for maximal value of the parameter λ. Therefore we prefer to use $\lambda = 1$.

Corollary 4.3. *The crisp output of the SISO P2-TS system is exactly the same as the consequent of the rule, if the input is either "$-\alpha$" or "β". The interpretation of the fuzzy rules R_1 and R_3 given by (4.12) for the P2-TS system is straightforward and analogous to the P1-TS systems.*

Similar considerations concerning P2-TS systems with many inputs will be given further on (see Theorem 4.11 and Example 4.13).

Now we introduce a *generator* for the SISO P2-TS system

$$\mathbf{g}(z) = \begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix}. \tag{4.16}$$

According to Remark 4.1, the function $f(z)$ to which the rule-based system (4.12) is equivalent, has the form

$$f(z) = \mathbf{g}^T(z)\,\boldsymbol{\theta}, \tag{4.17}$$

where $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2]^T$. The equality $S(z) = f(z)$ must be satisfied for $z \in [-\alpha, \beta]$, particularly for all three characteristic points from the set $\{-\alpha, \sigma, \beta\} \subset [-\alpha, \beta]$. Thus,

$$\mathbf{s} = \begin{bmatrix} f(-\alpha) \\ f(\sigma) \\ f(\beta) \end{bmatrix} = \begin{bmatrix} \mathbf{g}^T(-\alpha) \\ \mathbf{g}^T(\sigma) \\ \mathbf{g}^T(\beta) \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \boldsymbol{\Gamma}^T \boldsymbol{\theta}$$

must be satisfied, where the matrix $\boldsymbol{\Gamma}$ is the concatenation of the values of the generator (4.16) in the points "$-\alpha$", "σ", and "β", respectively, i.e.

$$\boldsymbol{\Gamma} = [\mathbf{g}(-\alpha), \mathbf{g}(\sigma), \mathbf{g}(\beta)].$$

Thus, we obtain the exact relationship between consequents $\mathbf{q}$ of the rules (4.12) and parameters $\boldsymbol{\theta}$ of the function (4.17) as follows

$$\mathbf{R}\mathbf{q} = \boldsymbol{\Gamma}^T \boldsymbol{\theta}.$$

Thus,

$$\mathbf{q} = \mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\theta} = \boldsymbol{\Omega}^T\boldsymbol{\theta}, \tag{4.18}$$

where the *fundamental matrix* for the SISO P2-TS system is defined by

$$\boldsymbol{\Omega} = \boldsymbol{\Gamma}\left(\mathbf{R}^T\right)^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha & \sigma & \beta \\ \alpha^2 & \left(\alpha^2+\beta^2\right)/2 - (\alpha+\beta)^2/(2\lambda) & \beta^2 \end{bmatrix}, \tag{4.19}$$

where $0 < \lambda \leq 1$. The inverse of $\boldsymbol{\Omega}$ always exists and is given by

$$\begin{aligned} \boldsymbol{\Omega}^{-1} &= \mathbf{R}^T\boldsymbol{\Gamma}^{-1} \\ &= \frac{1}{(\alpha+\beta)^2} \begin{bmatrix} \beta^2 + \alpha\beta(1-\lambda) & -\alpha(1-\lambda) - \beta(1+\lambda) & \lambda \\ 2\lambda\alpha\beta & 4\lambda\sigma & -2\lambda \\ \alpha^2 + \alpha\beta(1-\lambda) & \alpha(1+\lambda) + \beta(1-\lambda) & \lambda \end{bmatrix}. \end{aligned} \tag{4.20}$$

All equations are valid for any parameter value λ from the interval $(0, 1]$. Assuming $\lambda = 1$ and adding the index "1" for matrices in the case of SISO P2-TS system $(n = 1)$, we obtain

- the matrix (4.15) of membership degrees in the points from the set $\{-\alpha, \sigma, \beta\}$

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.21}$$

- the fundamental matrix of the SISO P2-TS system

$$\mathbf{\Omega}_1 = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_1 & \sigma_1 & \beta_1 \\ \alpha_1^2 & -\alpha_1\beta_1 & \beta_1^2 \end{bmatrix}, \tag{4.22}$$

- and the inverse of the fundamental matrix

$$\mathbf{\Omega}_1^{-1} = \frac{1}{(\alpha_1+\beta_1)^2} \begin{bmatrix} \beta_1^2 & -2\beta_1 & 1 \\ 2\beta_1\alpha_1 & 4\sigma_1 & -2 \\ \alpha_1^2 & 2\alpha_1 & 1 \end{bmatrix}. \tag{4.23}$$

The above formulas will be useful further on.

4.4 P2-TS System with Two and More Inputs

In this section we will investigate P2-TS systems with the inputs $z_1, \ldots, z_n$. For such systems, in order to define three membership functions N_k, Z_k and P_k as the functions of variables z_k, $(k = 1, 2, \ldots, n)$, we can choose individual parameter values $\lambda_1, \lambda_2, \ldots, \lambda_n$ for the particular inputs. The membership functions take the following general form

$$N_k(z_k) = \frac{(\alpha_k + \beta_k - \lambda_k(z_k + \alpha_k))(\beta_k - z_k)}{(\alpha_k+\beta_k)^2}, \tag{4.24}$$

$$Z_k(z_k) = 2\lambda_k \frac{(\beta_k - z_k)(z_k + \alpha_k)}{(\alpha_k+\beta_k)^2}, \tag{4.25}$$

$$P_k(z_k) = \frac{(\alpha_k + \beta_k + \lambda_k(z_k - \beta_k))(z_k + \alpha_k)}{(\alpha_k+\beta_k)^2}, \tag{4.26}$$

where $\lambda_k \in (0, 1]$, $(k = 1, \ldots, n)$. If there are no contraindications, we prefer to assume in practice the same value $\lambda_k = 1$ for all variables (see Section 4.3) – this corresponds to membership functions shown in Fig. 4.3.

Let M_n be a crisp set of 3^n *characteristic points* for the P2-TS system as n-dimensional vectors

$$M_n = \{-\alpha_1, \sigma_1, \beta_1\} \times \{-\alpha_2, \sigma_2, \beta_2\} \times \ldots \times \{-\alpha_n, \sigma_n, \beta_n\} \subset D^n. \tag{4.27}$$

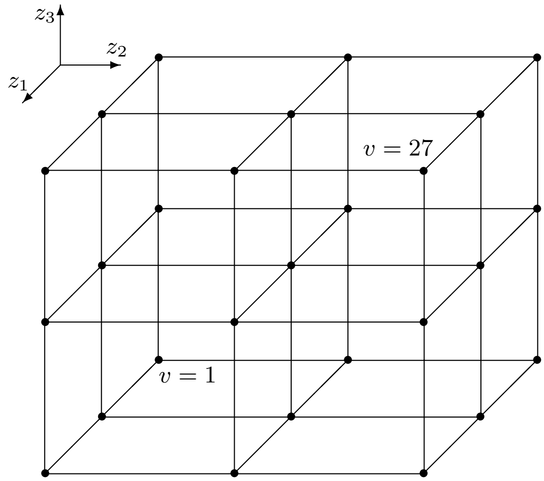


Fig. 4.5 The ordered set M_n for $n = 3$ with two depicted elements. The first one ($v = 1$) corresponds to the vector $(-\alpha_1, -\alpha_2, -\alpha_3)$ and the last one ($v = 27$) - to the vector $(\beta_1, \beta_2, \beta_3)$.

The set of characteristic points for P2-TS system includes all vertices of the hypercuboid D^n. We order M_n as follows. For every n-dimensional vector $(\gamma_1, \ldots, \gamma_n)$ as an element of the set M_n (see Fig. 4.5) we define the corresponding index v according to the following bijection

$$v = 1 + \sum_{i=1}^{n} 3^{i-1} p_i, \tag{4.28}$$

where

$$p_i = \begin{cases} 0 \Leftrightarrow \gamma_i = -\alpha_i \\ 1 \Leftrightarrow \gamma_i = \sigma_i \\ 2 \Leftrightarrow \gamma_i = \beta_i \end{cases}, \qquad i = 1, \ldots, n. \tag{4.29}$$

Thus, every element of the set M_n unambiguously corresponds to some index. For $(\gamma_1', \ldots, \gamma_n') \in M_n$ and $(\gamma_1'', \ldots, \gamma_n'') \in M_n$ we define an ordering relation "$\prec$" as follows

$$(\gamma_1', \ldots, \gamma_n') \prec (\gamma_1'', \ldots, \gamma_n'') \qquad \Leftrightarrow \qquad v_{\gamma_1', \ldots, \gamma_n'} < v_{\gamma_1'', \ldots, \gamma_n''}. \tag{4.30}$$

- For $n = 1$ we have $v_{-\alpha} = 1 < v_\sigma = 2 < v_\beta = 3$ and therefore $-\alpha \prec \sigma \prec \beta$.
- For $n = 2$ the inequalities between indices are $v_{-\alpha_1,-\alpha_2} = 1 < v_{\sigma_1,-\alpha_2} = 2 < v_{\beta_1,-\alpha_2} = 3 < v_{-\alpha_1,\sigma_2} = 4 < v_{\sigma_1,\sigma_2} = 5 < v_{\beta_1,\sigma_2} = 6 < v_{-\alpha_1,\beta_2} = 7 < v_{\sigma_1,\beta_2} = 8 < v_{\beta_1,\beta_2} = 9$.

Thus, the members of M_2 are ordered as follows
$(-\alpha_1, -\alpha_2) \prec (\sigma_1, -\alpha_2) \prec (\beta_1, -\alpha_2) \prec (-\alpha_1, \sigma_2) \prec (\sigma_1, \sigma_2) \prec (\beta_1, \sigma_2) \prec (-\alpha_1, \beta_2) \prec (\sigma_1, \beta_2) \prec (\beta_1, \beta_2)$.

- For the ordered set M_3 we have
$(-\alpha_1, -\alpha_2, -\alpha_3) \prec (\sigma_1, -\alpha_2, -\alpha_3) \prec (\beta_1, -\alpha_2, -\alpha_3) \prec (-\alpha_1, \sigma_2, -\alpha_3) \prec (\sigma_1, \sigma_2, -\alpha_3) \prec (\beta_1, \sigma_2, -\alpha_3) \prec (-\alpha_1, \beta_2, -\alpha_3) \prec (\sigma_1, \beta_2, -\alpha_3) \prec (\beta_1, \beta_2, -\alpha_3) \prec (-\alpha_1, -\alpha_2, \sigma_3) \prec (\sigma_1, -\alpha_2, \sigma_3) \prec (\beta_1, -\alpha_2, \sigma_3) \prec (-\alpha_1, \sigma_2, \sigma_3) \prec (\sigma_1, \sigma_2, \sigma_3) \prec (\beta_1, \sigma_2, \sigma_3) \prec (-\alpha_1, \beta_2, \sigma_3) \prec (\sigma_1, \beta_2, \sigma_3) \prec (\beta_1, \beta_2, \sigma_3) \prec (-\alpha_1, -\alpha_2, \beta_3) \prec (\sigma_1, -\alpha_2, \beta_3) \prec (\beta_1, -\alpha_2, \beta_3) \prec (-\alpha_1, \sigma_2, \beta_3) \prec (\sigma_1, \sigma_2, \beta_3) \prec (\beta_1, \sigma_2, \beta_3) \prec (-\alpha_1, \beta_2, \beta_3) \prec (\sigma_1, \beta_2, \beta_3) \prec (\beta_1, \beta_2, \beta_3)$.

The process of ordering the set M_n is simple and unambiguous for any number of system inputs.

Finally, for the MISO P2-TS system with the inputs z_1, …, z_k let us introduce a *generator*

$$\begin{gathered} \mathbf{g}_0 = 1, \\ \mathbf{g}_{k+1}(z_1, \ldots, z_{k+1}) = \begin{bmatrix} \mathbf{g}_k(z_1, \ldots, z_k) \\ z_{k+1}\mathbf{g}_k(z_1, \ldots, z_k) \\ z_{k+1}^2\mathbf{g}_k(z_1, \ldots, z_k) \end{bmatrix} \in \mathbb{R}^{3^{k+1}}, \quad k = 0, 1, 2, \ldots, n-1, \end{gathered} \tag{4.31}$$

which is of great importance for such systems.

4.4.1 Rule-Base Structure for Two-Inputs-One-Output P2-TS System

For $n = 2$ the rule-base structure is as follows

$$\left.\begin{aligned} R_1 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_{00}, \\ R_2 &: \text{If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_{10}, \\ R_3 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = q_{20}, \\ R_4 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = q_{01}, \\ R_5 &: \text{If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = q_{11}, \\ R_6 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = q_{21}, \\ R_7 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_{02}, \\ R_8 &: \text{If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_{12}, \\ R_9 &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = q_{22}, \end{aligned}\right\} \tag{4.32}$$

and, in accordance with (4.31), the generator is given by

$$\mathbf{g}_2(z_1,z_2)=\begin{bmatrix}\mathbf{g}_1(z_1)\\ z_2\mathbf{g}_1(z_1)\\ z_2^2\mathbf{g}_1(z_1)\end{bmatrix}=\left[1,z_1,z_1^2,z_2,z_1z_2,z_1^2z_2,z_2^2,z_1z_2^2,z_1^2z_2^2\right]^T. \quad (4.33)$$

The crisp output of the system can be expressed as a scalar product of two vectors

$$S(z_1,z_2)=[N_1N_2,Z_1N_2,P_1N_2,N_1Z_2,Z_1Z_2,P_1Z_2,N_1P_2,Z_1P_2,P_1P_2]\,\mathbf{q}, \quad (4.34)$$

where $N_k=N_k(z_k)$, $Z_k=Z_k(z_k)$ and $P_k=P_k(z_k)$ for $k=1,2$ are the membership functions defined by (4.24)-(4.26), and the vector $\mathbf{q}$ consists of the conclusions of the rules (4.32)

$$\mathbf{q}=[q_{00},q_{10},q_{20},q_{01},q_{11},q_{21},q_{02},q_{12},q_{22}]^T. \quad (4.35)$$

On the other hand, according to Remark 4.1 we have

$$S(\mathbf{z})=\mathbf{g}_2^T(\mathbf{z})\,\boldsymbol{\theta}, \qquad \mathbf{z}\in D^2,$$

where $\boldsymbol{\theta}=[\theta_{00},\theta_{10},\theta_{20},\theta_{01},\theta_{11},\theta_{21},\theta_{02},\theta_{12},\theta_{22}]^T$ and $\mathbf{g}_2(\mathbf{z})$ is given by (4.33).

4.4.2 Rule-Base Structure for Three-Inputs-One-Output P2-TS System

For $n=3$ the rule base consists of 27 rules. Its abbreviated structure is as follows

R_1 : If z_1 is N_1 and z_2 is N_2 and z_3 is N_3, then $S=q_{000}$,
R_2 : If z_1 is Z_1 and z_2 is N_2 and z_3 is N_3, then $S=q_{100}$,
R_3 : If z_1 is P_1 and z_2 is N_2 and z_3 is N_3, then $S=q_{200}$,
R_4 : If z_1 is N_1 and z_2 is Z_2 and z_3 is N_3, then $S=q_{010}$,
R_5 : If z_1 is Z_1 and z_2 is Z_2 and z_3 is N_3, then $S=q_{110}$,
R_6 : If z_1 is P_1 and z_2 is Z_2 and z_3 is N_3, then $S=q_{210}$,
R_7 : If z_1 is N_1 and z_2 is P_2 and z_3 is N_3, then $S=q_{020}$,
R_8 : If z_1 is Z_1 and z_2 is P_2 and z_3 is N_3, then $S=q_{120}$,
R_9 : If z_1 is P_1 and z_2 is P_2 and z_3 is N_3, then $S=q_{220}$,
R_{10} : If z_1 is N_1 and z_2 is N_2 and z_3 is Z_3, then $S=q_{001}$,

$\vdots$

R_{27} : If z_1 is P_1 and z_2 is P_2 and z_3 is P_3, then $S=q_{222}$,

and the generator

$$\mathbf{g}_3(z_1,z_2,z_3) = \begin{bmatrix} \mathbf{g}_2(z_1,z_2) \\ z_3\mathbf{g}_2(z_1,z_2) \\ z_3^2\mathbf{g}_2(z_1,z_2) \end{bmatrix}$$
$$= [1, z_1, z_1^2, z_2, z_1z_2, z_1^2z_2, z_2^2, z_1z_2^2, z_1^2z_2^2,$$
$$z_3, z_1z_3, z_1^2z_3, z_2z_3, z_1z_2z_3, z_1^2z_2z_3,$$
$$z_2^2z_3, z_1z_2^2z_3, z_1^2z_2^2z_3, z_3^2, z_1z_3^2, z_1^2z_3^2, z_2z_3^2,$$
$$z_1z_2z_3^2, z_1^2z_2z_3^2, z_2^2z_3^2, z_1z_2^2z_3^2, z_1^2z_2^2z_3^2]^T. \tag{4.36}$$

The output of a three-input P2-TS system can be expressed and computed in the same way as for a two-input system - this is rather a simple task, but the equations are large for the number of inputs $n \geq 3$. For MISO P2-TS systems with $n \geq 3$ inputs we prefer to use the methods based on recurrence, which will be presented in the next sections.

4.5 The Fundamental Matrix for MISO P2-TS System

Similarly to SISO P2-TS systems, for the MISO P2-TS systems, the same equations as in (4.18) hold, namely

$$\mathbf{q} = \mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\theta} = \boldsymbol{\Omega}^T\boldsymbol{\theta}, \tag{4.37}$$

where

- the vector $\mathbf{q}$ contains the consequents of the "If-then" rules,
- $\boldsymbol{\theta}$ is the vector of parameters of the crisp function (4.1) to which the MISO P2-TS system is equivalent,
- the meaning of matrices $\mathbf{R}$ and $\boldsymbol{\Gamma}$ is the same as in Section 4.3, after some generalization for MISO systems,
- the matrix
$$\boldsymbol{\Omega} = \boldsymbol{\Gamma}\left(\mathbf{R}^{-1}\right)^T \tag{4.38}$$
we will call the *fundamental matrix* for P2-TS system.

Both $\boldsymbol{\Omega}$ and its inverse are important, since they enable one to establish an exact relationship between the consequents $\mathbf{q}$ of the "If-then" rules and the parameters $\boldsymbol{\theta}$ of the crisp function (4.1), to which the rule-based system is equivalent. Therefore our goal in this section is to give a procedure of how to compute the fundamental matrix and its inverse in the general case.

First we prove the following

Lemma 4.4. *For the MISO P2-TS system with the inputs* $[z_1,\ldots,z_k]^T \in D^k$*, we define the matrix*

$$\boldsymbol{\Gamma}_k = [\mathbf{g}_k(-\alpha_1,\ldots,-\alpha_k), \cdots, \mathbf{g}_k(\beta_1,\beta_2,\ldots,\beta_k)], \tag{4.39}$$

for $k = 1, 2, \ldots, n$, *where the values of the generator* $\mathbf{g}_k$ *defined in* (4.31) *are computed for the subsequent elements of the totally ordered set* M_k *defined by* (4.27)*. The matrix* $\boldsymbol{\Gamma}_k$ *can be computed recursively as follows*

$$\boldsymbol{\Gamma}_0 = 1,$$
$$\boldsymbol{\Gamma}_{k+1} = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_{k+1} & \sigma_{k+1} & \beta_{k+1} \\ \alpha_{k+1}^2 & \sigma_{k+1}^2 & \beta_{k+1}^2 \end{bmatrix} \otimes \boldsymbol{\Gamma}_k, \tag{4.40}$$

for $k = 0, 1, 2, \ldots, n-1$.

Proof. From (4.39) by $\mathbf{g}_1(z_1) = \mathbf{g}(z)$ defined in (4.16) we obtain

$$\boldsymbol{\Gamma}_1 = [\mathbf{g}_1(-\alpha_1), \mathbf{g}_1(\sigma_1), \mathbf{g}_1(\beta_1)] = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_1 & \sigma_1 & \beta_1 \\ \alpha_1^2 & \sigma_1^2 & \beta_1^2 \end{bmatrix}.$$

On the other hand from (4.40) for $k = 0$ we have

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_1 \cdot 1 & \sigma_1 \cdot 1 & \beta_1 \cdot 1 \\ (-\alpha_1)^2 \cdot 1 & \sigma_1^2 \cdot 1 & \beta_1^2 \cdot 1 \end{bmatrix}.$$

Thus, for $k = 0$ the recurrence (4.40) is true.

For $k \geq 1$ let us rewrite the equation (4.39), taking into account (4.31)

$$\boldsymbol{\Gamma}_{k+1} = \left[\mathbf{g}_{k+1}(-\alpha_1, \ldots, -\alpha_k, -\alpha_{k+1}), \ \cdots, \ \mathbf{g}_{k+1}(\beta_1, \ldots, \beta_k, \beta_{k+1}) \right]$$
$$= \left[\begin{bmatrix} \mathbf{g}_k(-\alpha_1, \ldots, -\alpha_k) \\ -\alpha_{k+1}\mathbf{g}_k(-\alpha_1, \ldots, -\alpha_k) \\ (-\alpha_{k+1})^2 \mathbf{g}_k(-\alpha_1, \ldots, -\alpha_k) \end{bmatrix}, \cdots, \begin{bmatrix} \mathbf{g}_k(\beta_1, \ldots, \beta_k) \\ \beta_{k+1}\mathbf{g}_k(\beta_1, \ldots, \beta_k) \\ \beta_{k+1}^2 \mathbf{g}_k(\beta_1, \ldots, \beta_k) \end{bmatrix} \right]. \tag{4.41}$$

For example

$$\boldsymbol{\Gamma}_2 = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3],$$

where the column vectors $\mathbf{a}_i$, $\mathbf{b}_i$ and $\mathbf{c}_i$ are

$$\mathbf{a}_1 = \begin{bmatrix} \mathbf{g}_1(-\alpha_1) \\ (-\alpha_2)\mathbf{g}_1(-\alpha_1) \\ \alpha_2^2 \mathbf{g}_1(-\alpha_1) \end{bmatrix}, \ \mathbf{a}_2 = \begin{bmatrix} \mathbf{g}_1(\sigma_1) \\ (-\alpha_2)\mathbf{g}_1(\sigma_1) \\ \alpha_2^2 \mathbf{g}_1(\sigma_1) \end{bmatrix}, \ \mathbf{a}_3 = \begin{bmatrix} \mathbf{g}_1(\beta_1) \\ (-\alpha_2)\mathbf{g}_1(\beta_1) \\ \alpha_2^2 \mathbf{g}_1(\beta_1) \end{bmatrix},$$

$$\mathbf{b}_1 = \begin{bmatrix} \mathbf{g}_1(-\alpha_1) \\ \sigma_2 \mathbf{g}_1(-\alpha_1) \\ \sigma_2^2 \mathbf{g}_1(-\alpha_1) \end{bmatrix}, \qquad \mathbf{b}_2 = \begin{bmatrix} \mathbf{g}_1(\sigma_1) \\ \sigma_2 \mathbf{g}_1(\sigma_1) \\ \sigma_2^2 \mathbf{g}_1(\sigma_1) \end{bmatrix}, \qquad \mathbf{b}_3 = \begin{bmatrix} \mathbf{g}_1(\beta_1) \\ \sigma_2 \mathbf{g}_1(\beta_1) \\ \sigma_2^2 \mathbf{g}_1(\beta_1) \end{bmatrix},$$

$$\mathbf{c}_1 = \begin{bmatrix} \mathbf{g}_1(-\alpha_1) \\ \beta_2 \mathbf{g}_1(-\alpha_1) \\ \beta_2^2 \mathbf{g}_1(-\alpha_1) \end{bmatrix}, \qquad \mathbf{c}_2 = \begin{bmatrix} \mathbf{g}_1(\sigma_1) \\ \beta_2 \mathbf{g}_1(\sigma_1) \\ \beta_2^2 \mathbf{g}_1(\sigma_1) \end{bmatrix}, \qquad \mathbf{c}_3 = \begin{bmatrix} \mathbf{g}_1(\beta_1) \\ \beta_2 \mathbf{g}_1(\beta_1) \\ \beta_2^2 \mathbf{g}_1(\beta_1) \end{bmatrix},$$

what results in

$$\boldsymbol{\Gamma}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -\alpha_1 & \sigma_1 & \beta_1 & -\alpha_1 & \sigma_1 & \beta_1 & -\alpha_1 & \sigma_1 & \beta_1 \\ \alpha_1^2 & \sigma_1^2 & \beta_1^2 & \alpha_1^2 & \sigma_1^2 & \beta_1^2 & \alpha_1^2 & \sigma_1^2 & \beta_1^2 \\ -\alpha_2 & -\alpha_2 & -\alpha_2 & \sigma_2 & \sigma_2 & \sigma_2 & \beta_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\sigma_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\sigma_2 & \sigma_1\sigma_2 & \beta_1\sigma_2 & -\alpha_1\beta_2 & \sigma_1\beta_2 & \beta_1\beta_2 \\ -\alpha_1^2\alpha_2 & -\sigma_1^2\alpha_2 & -\alpha_2\beta_1^2 & \alpha_1^2\sigma_2 & \sigma_1^2\sigma_2 & \beta_1^2\sigma_2 & \alpha_1^2\beta_2 & \sigma_1^2\beta_2 & \beta_1^2\beta_2 \\ \alpha_2^2 & \alpha_2^2 & \alpha_2^2 & \sigma_2^2 & \sigma_2^2 & \sigma_2^2 & \beta_2^2 & \beta_2^2 & \beta_2^2 \\ -\alpha_1\alpha_2^2 & \sigma_1\alpha_2^2 & \alpha_2^2\beta_1 & -\alpha_1\sigma_2^2 & \sigma_1\sigma_2^2 & \beta_1\sigma_2^2 & -\alpha_1\beta_2^2 & \sigma_1\beta_2^2 & \beta_1\beta_2^2 \\ \alpha_1^2\alpha_2^2 & \sigma_1^2\alpha_2^2 & \alpha_2^2\beta_1^2 & \alpha_1^2\sigma_2^2 & \sigma_1^2\sigma_2^2 & \beta_1^2\sigma_2^2 & \alpha_1^2\beta_2^2 & \sigma_1^2\beta_2^2 & \beta_1^2\beta_2^2 \end{bmatrix},$$

or equivalently

$$\boldsymbol{\Gamma}_2 = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_2 & \sigma_2 & \beta_2 \\ (-\alpha_2)^2 & \sigma_2^2 & \beta_2^2 \end{bmatrix} \otimes \boldsymbol{\Gamma}_1.$$

One can observe that in the general case, because of the generator structure (4.31) and the sequence of the characteristic points from the set M_k, the structure of the matrix $\boldsymbol{\Gamma}_{k+1}$ is as follows

(a) The first 3^k columns of $\boldsymbol{\Gamma}_{k+1}$ constitute the submatrix $\begin{bmatrix} \boldsymbol{\Gamma}_k \\ -\alpha_{k+1}\boldsymbol{\Gamma}_k \\ (-\alpha_{k+1})^2 \boldsymbol{\Gamma}_k \end{bmatrix}$.

(b) The next 3^k columns of $\boldsymbol{\Gamma}_{k+1}$ constitute the submatrix $\begin{bmatrix} \boldsymbol{\Gamma}_k \\ \sigma_{k+1}\boldsymbol{\Gamma}_k \\ \sigma_{k+1}^2 \boldsymbol{\Gamma}_k \end{bmatrix}$.

(c) The last 3^k columns of $\boldsymbol{\Gamma}_{k+1}$ constitute the submatrix $\begin{bmatrix} \boldsymbol{\Gamma}_k \\ \beta_{k+1}\boldsymbol{\Gamma}_k \\ \beta_{k+1}^2 \boldsymbol{\Gamma}_k \end{bmatrix}$.

This finishes the proof of Lemma 4.4. □

Lemma 4.5. *For the MISO P2-TS system with the inputs $z_1, \ldots, z_k$, let us denote by $\mathbf{s}_k$ the vector of its outputs in the consecutive points of the ordered set M_k defined by (4.27), and the vector $\mathbf{q}_k$ containing the consequents of the rules*

$$\mathbf{s}_k = \begin{bmatrix} S(-\alpha_1, -\alpha_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\sigma_1, -\alpha_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\beta_1, -\alpha_2, -\alpha_3, \ldots, -\alpha_k) \\ S(-\alpha_1, \sigma_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\sigma_1, \sigma_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\beta_1, \sigma_2, -\alpha_3, \ldots, -\alpha_k) \\ S(-\alpha_1, \beta_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\sigma_1, \beta_2, -\alpha_3, \ldots, -\alpha_k) \\ S(\beta_1, \beta_2, -\alpha_3, \ldots, -\alpha_k) \\ \vdots \\ S(\beta_1, \beta_2, \beta_3, \ldots, \beta_k) \end{bmatrix}, \qquad \mathbf{q}_k = \begin{bmatrix} q_{000\ldots0} \\ q_{100\ldots0} \\ q_{200\ldots0} \\ q_{010\ldots0} \\ q_{110\ldots0} \\ q_{210\ldots0} \\ q_{020\ldots0} \\ q_{120\ldots0} \\ q_{220\ldots0} \\ \vdots \\ q_{222\ldots2} \end{bmatrix}. \tag{4.42}$$

There exists a matrix $\mathbf{R}_k \in \mathbb{R}^{3^k \times 3^k}$ such that

$$\mathbf{s}_k = \mathbf{R}_k \mathbf{q}_k, \tag{4.43}$$

and $\mathbf{R}_k$ can be recursively computed as follows

$$\begin{aligned} \mathbf{R}_0 &= 1, \\ \mathbf{R}_{k+1} &= \begin{bmatrix} 1 & 0 & 0 \\ (2-\lambda_{k+1})/4 & \lambda_{k+1}/2 & (2-\lambda_{k+1})/4 \\ 0 & 0 & 1 \end{bmatrix} \otimes \mathbf{R}_k, \end{aligned} \tag{4.44}$$

for $k = 0, 1, 2, \ldots, n-1$, where $\lambda_k \in (0, 1]$ is the parameter of membership functions (4.24)-(4.26).

Proof. Let us consider the system with one input $z_1 \in [-\alpha_1, \beta_1]$. From the results in Section 4.3 we have

$$\begin{bmatrix} S(-\alpha_1) \\ S(\sigma_1) \\ S(\beta_1) \end{bmatrix} = \mathbf{R}_1 \mathbf{q}_1 = \begin{bmatrix} N_1(-\alpha_1) & Z_1(-\alpha_1) & P_1(-\alpha_1) \\ N_1(\sigma_1) & Z_1(\sigma_1) & P_1(\sigma_1) \\ N_1(\beta_1) & Z_1(\beta_1) & P_1(\beta_1) \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix}.$$

Thus,

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ (2-\lambda_1)/4 & \lambda_1/2 & (2-\lambda_1)/4 \\ 0 & 0 & 1 \end{bmatrix} \otimes 1,$$

i.e. the result is the same as in (4.44).

For the system with two inputs the equality $\mathbf{s}_2 = \mathbf{R}_2 \mathbf{q}_2$ holds, where

$$\mathbf{s}_2 = \begin{bmatrix} S(-\alpha_1, -\alpha_2) \\ S(\sigma_1, -\alpha_2) \\ S(\beta_1, -\alpha_2) \\ S(-\alpha_1, \sigma_2) \\ S(\sigma_1, \sigma_2) \\ S(\beta_1, \sigma_2) \\ S(-\alpha_1, \beta_2) \\ S(\sigma_1, \beta_2) \\ S(\beta_1, \beta_2) \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} \mathbf{R}_{2,11} & \mathbf{R}_{2,12} & \mathbf{R}_{2,13} \\ \mathbf{R}_{2,21} & \mathbf{R}_{2,22} & \mathbf{R}_{2,23} \\ \mathbf{R}_{2,31} & \mathbf{R}_{2,32} & \mathbf{R}_{2,33} \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} q_{00} \\ q_{10} \\ q_{20} \\ q_{01} \\ q_{11} \\ q_{21} \\ q_{02} \\ q_{12} \\ q_{22} \end{bmatrix},$$

and

$$\begin{aligned}\mathbf{R}_{2,11} &= \begin{bmatrix} N_1(-\alpha_1) N_2(-\alpha_2) & Z_1(-\alpha_1) N_2(-\alpha_2) & P_1(-\alpha_1) N_2(-\alpha_2) \\ N_1(\sigma_1) N_2(-\alpha_2) & Z_1(\sigma_1) N_2(-\alpha_2) & P_1(\sigma_1) N_2(-\alpha_2) \\ N_1(\beta_1) N_2(-\alpha_2) & Z_1(\beta_1) N_2(-\alpha_2) & P_1(\beta_1) N_2(-\alpha_2) \end{bmatrix} \\ &= \mathbf{R}_1 \cdot N_2(-\alpha_2),\end{aligned}$$

$$\begin{aligned}\mathbf{R}_{2,12} &= \begin{bmatrix} N_1(-\alpha_1) Z_2(-\alpha_2) & Z_1(-\alpha_1) Z_2(-\alpha_2) & P_1(-\alpha_1) Z_2(-\alpha_2) \\ N_1(\sigma_1) Z_2(-\alpha_2) & Z_1(\sigma_1) Z_2(-\alpha_2) & P_1(\sigma_1) Z_2(-\alpha_2) \\ N_1(\beta_1) Z_2(-\alpha_2) & Z_1(\beta_1) Z_2(-\alpha_2) & P_1(\beta_1) Z_2(-\alpha_2) \end{bmatrix} \\ &= \mathbf{R}_1 \cdot Z_2(-\alpha_2),\end{aligned}$$

$$\begin{aligned}\mathbf{R}_{2,13} &= \begin{bmatrix} N_1(-\alpha_1) P_2(-\alpha_2) & Z_1(-\alpha_1) P_2(-\alpha_2) & P_1(-\alpha_1) P_2(-\alpha_2) \\ N_1(\sigma_1) P_2(-\alpha_2) & Z_1(\sigma_1) P_2(-\alpha_2) & P_1(\sigma_1) P_2(-\alpha_2) \\ N_1(\beta_1) P_2(-\alpha_2) & Z_1(\beta_1) P_2(-\alpha_2) & P_1(\beta_1) P_2(-\alpha_2) \end{bmatrix} \\ &= \mathbf{R}_1 \cdot P_2(-\alpha_2),\end{aligned}$$

$$\begin{aligned}\mathbf{R}_{2,21} &= \begin{bmatrix} N_1(-\alpha_1) N_2(\sigma_2) & Z_1(-\alpha_1) N_2(\sigma_2) & P_1(-\alpha_1) N_2(\sigma_2) \\ N_1(\sigma_1) N_2(\sigma_2) & Z_1(\sigma_1) N_2(\sigma_2) & P_1(\sigma_1) N_2(\sigma_2) \\ N_1(\beta_1) N_2(\sigma_2) & Z_1(\beta_1) N_2(\sigma_2) & P_1(\beta_1) N_2(\sigma_2) \end{bmatrix} \\ &= \mathbf{R}_1 \cdot N_2(\sigma_2),\end{aligned}$$

$$\begin{aligned}\mathbf{R}_{2,22} &= \begin{bmatrix} N_1(-\alpha_1) Z_2(\sigma_2) & Z_1(-\alpha_1) Z_2(\sigma_2) & P_1(-\alpha_1) Z_2(\sigma_2) \\ N_1(\sigma_1) Z_2(\sigma_2) & Z_1(\sigma_1) Z_2(\sigma_2) & P_1(\sigma_1) Z_2(\sigma_2) \\ N_1(\beta_1) Z_2(\sigma_2) & Z_1(\beta_1) Z_2(\sigma_2) & P_1(\beta_1) Z_2(\sigma_2) \end{bmatrix} \\ &= \mathbf{R}_1 \cdot Z_2(\sigma_2),\end{aligned}$$

$$\mathbf{R}_{2,23} = \begin{bmatrix} N_1(-\alpha_1)\,P_2(\sigma_2) & Z_1(-\alpha_1)\,P_2(\sigma_2) & P_1(-\alpha_1)\,P_2(\sigma_2) \\ N_1(\sigma_1)\,P_2(\sigma_2) & Z_1(\sigma_1)\,P_2(\sigma_2) & P_1(\sigma_1)\,P_2(\sigma_2) \\ N_1(\beta_1)\,P_2(\sigma_2) & Z_1(\beta_1)\,P_2(\sigma_2) & P_1(\beta_1)\,P_2(\sigma_2) \end{bmatrix}$$

$$= \mathbf{R}_1 \cdot P_2(\sigma_2)\,,$$

$$\mathbf{R}_{2,31} = \begin{bmatrix} N_1(-\alpha_1)\,N_2(\beta_2) & Z_1(-\alpha_1)\,N_2(\beta_2) & P_1(-\alpha_1)\,N_2(\beta_2) \\ N_1(\sigma_1)\,N_2(\beta_2) & Z_1(\sigma_1)\,N_2(\beta_2) & P_1(\sigma_1)\,N_2(\beta_2) \\ N_1(\beta_1)\,N_2(\beta_2) & Z_1(\beta_1)\,N_2(\beta_2) & P_1(\beta_1)\,N_2(\beta_2) \end{bmatrix}$$

$$= \mathbf{R}_1 \cdot N_2(\beta_2)\,,$$

$$\mathbf{R}_{2,32} = \begin{bmatrix} N_1(-\alpha_1)\,Z_2(\beta_2) & Z_1(-\alpha_1)\,Z_2(\beta_2) & P_1(-\alpha_1)\,Z_2(\beta_2) \\ N_1(\sigma_1)\,Z_2(\beta_2) & Z_1(\sigma_1)\,Z_2(\beta_2) & P_1(\sigma_1)\,Z_2(\beta_2) \\ N_1(\beta_1)\,Z_2(\beta_2) & Z_1(\beta_1)\,Z_2(\beta_2) & P_1(\beta_1)\,Z_2(\beta_2) \end{bmatrix}$$

$$= \mathbf{R}_1 \cdot Z_2(\beta_2)\,,$$

and

$$\mathbf{R}_{2,33} = \begin{bmatrix} N_1(-\alpha_1)\,P_2(\beta_2) & Z_1(-\alpha_1)\,P_2(\beta_2) & P_1(-\alpha_1)\,P_2(\beta_2) \\ N_1(\sigma_1)\,P_2(\beta_2) & Z_1(\sigma_1)\,P_2(\beta_2) & P_1(\sigma_1)\,P_2(\beta_2) \\ N_1(\beta_1)\,P_2(\beta_2) & Z_1(\beta_1)\,P_2(\beta_2) & P_1(\beta_1)\,P_2(\beta_2) \end{bmatrix}$$

$$= \mathbf{R}_1 \cdot P_2(\beta_2)\,.$$

In a more compact form we can write

$$\mathbf{R}_2 = \begin{bmatrix} N_2(-\alpha_2) & Z_2(-\alpha_2) & P_2(-\alpha_2) \\ N_2(\sigma_2) & Z_2(\sigma_2) & P_2(\sigma_2) \\ N_2(\beta_2) & Z_2(\beta_2) & P_2(\beta_2) \end{bmatrix} \otimes \mathbf{R}_1.$$

The same procedure must be applied for the construction of the matrix $\mathbf{R}_k$ in (4.43), remembering the order of the set M_k. Finally, we conclude that the following recurrence

$$\mathbf{R}_{k+1} = \begin{bmatrix} N_{k+1}(-\alpha_{k+1}) & Z_{k+1}(-\alpha_{k+1}) & P_{k+1}(-\alpha_{k+1}) \\ N_{k+1}(\sigma_{k+1}) & Z_{k+1}(\sigma_{k+1}) & P_{k+1}(\sigma_{k+1}) \\ N_{k+1}(\beta_{k+1}) & Z_{k+1}(\beta_{k+1}) & P_{k+1}(\beta_{k+1}) \end{bmatrix} \otimes \mathbf{R}_k$$

holds for every natural k. After computing the membership degrees according to (4.24)-(4.26) we obtain the recursive formula (4.44). This ends the proof of Lemma 4.5. □

Observe that for $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$ we have

$$\mathbf{R}_{k+1} = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1 \end{bmatrix} \otimes \mathbf{R}_k, \qquad k = 0, 1, 2, \ldots, n-1.$$

For example

$$\mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1/2 & 1/4 \\ 0 & 0 & 1 \end{bmatrix} \otimes \mathbf{R}_1$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 1/2 & 0 & 0 & 1/4 & 0 & 0 \\ 1/16 & 1/8 & 1/16 & 1/8 & 1/4 & 1/8 & 1/16 & 1/8 & 1/16 \\ 0 & 0 & 1/4 & 0 & 0 & 1/2 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrix $\mathbf{R}_3$ contains $3^3 \times 3^3 = 729$ elements and because of its large size it will not be presented here.

Now we prove the following

Theorem 4.6. *The fundamental matrix of the MISO P2-TS system with the inputs $[z_1, \ldots, z_k]^T \in D^k$ and the membership functions of fuzzy sets for the inputs defined by (4.24)-(4.26), can be computed recursively as follows*

$$\mathbf{\Omega}_0 = 1,$$
$$\mathbf{\Omega}_k = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_k & \sigma_k & \beta_k \\ \alpha_k^2 & \frac{1}{2}\left(\alpha_k^2 + \beta_k^2 - \frac{(\alpha_k + \beta_k)^2}{\lambda_k}\right) & \beta_k^2 \end{bmatrix} \otimes \mathbf{\Omega}_{k-1}, \tag{4.45}$$

for $k = 1, \ldots, n$, where $\lambda_k \in (0, 1]$ is the parameter of membership functions.

Proof. From (4.38) for MISO P2-TS system with the inputs $[z_1, \ldots, z_k]^T \in D^k$ we have

$$\mathbf{\Omega}_k = \boldsymbol{\Gamma}_k \left(\mathbf{R}_k^{-1}\right)^T. \tag{4.46}$$

Next we apply Lemma 4.4 and Lemma 4.5

$$\boldsymbol{\Omega}_k = (\mathbf{A}_k \otimes \boldsymbol{\Gamma}_{k-1}) \left((\mathbf{B}_k \otimes \mathbf{R}_{k-1})^{-1} \right)^T ,$$

where according to (4.40) and (4.44) the matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ are

$$\mathbf{A}_k = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_k & \sigma_k & \beta_k \\ \alpha_k^2 & \sigma_k^2 & \beta_k^2 \end{bmatrix}, \qquad \mathbf{B}_k = \begin{bmatrix} 1 & 0 & 0 \\ (2-\lambda_k)/4 & \lambda_k/2 & (2-\lambda_k)/4 \\ 0 & 0 & 1 \end{bmatrix}.$$

From (A.4) and (A.5) given in Appendix A we obtain

$$\left((\mathbf{B}_k \otimes \mathbf{R}_{k-1})^{-1} \right)^T = \left(\mathbf{B}_k^{-1} \otimes \mathbf{R}_{k-1}^{-1} \right)^T = \left(\mathbf{B}_k^{-1} \right)^T \otimes \left(\mathbf{R}_{k-1}^{-1} \right)^T .$$

Thus,

$$\boldsymbol{\Omega}_k = (\mathbf{A}_k \otimes \boldsymbol{\Gamma}_{k-1}) \left(\mathbf{B}_k^{-1} \right)^T \otimes \left(\mathbf{R}_{k-1}^{-1} \right)^T = \left(\mathbf{A}_k \left(\mathbf{B}_k^{-1} \right)^T \right) \otimes \left(\boldsymbol{\Gamma}_{k-1} \left(\mathbf{R}_{k-1}^{-1} \right)^T \right).$$

One can check that

$$\mathbf{A}_k \left(\mathbf{B}_k^{-1} \right)^T = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_k & \sigma_k & \beta_k \\ \alpha_k^2 & \frac{1}{2}\left(\alpha_k^2 + \beta_k^2 - \frac{(\alpha_k + \beta_k)^2}{\lambda_k} \right) & \beta_k^2 \end{bmatrix}.$$

Now we apply the Kronecker product property (A.3) from Appendix A:

$$\boldsymbol{\Omega}_k = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_k & \sigma_k & \beta_k \\ \alpha_k^2 & \frac{1}{2}\left(\alpha_k^2 + \beta_k^2 - \frac{(\alpha_k + \beta_k)^2}{\lambda_k} \right) & \beta_k^2 \end{bmatrix} \otimes \left(\boldsymbol{\Gamma}_{k-1} \left(\mathbf{R}_{k-1}^{-1} \right)^T \right). \tag{4.47}$$

According to (4.46) the equality $\boldsymbol{\Gamma}_{k-1} \left(\mathbf{R}_{k-1}^{-1} \right)^T = \boldsymbol{\Omega}_{k-1}$ holds. Thus, the equation (4.47) is the same as (4.45) and this finishes the proof of Theorem 4.6. □

For $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$ we obtain a much simpler recurrence

$$\boldsymbol{\Omega}_0 = 1,$$
$$\boldsymbol{\Omega}_k = \begin{bmatrix} 1 & 1 & 1 \\ -\alpha_k & \sigma_k & \beta_k \\ \alpha_k^2 & -\beta_k \alpha_k & \beta_k^2 \end{bmatrix} \otimes \boldsymbol{\Omega}_{k-1}, \qquad k = 1, \ldots, n, \tag{4.48}$$

which we prefer to use in practice.

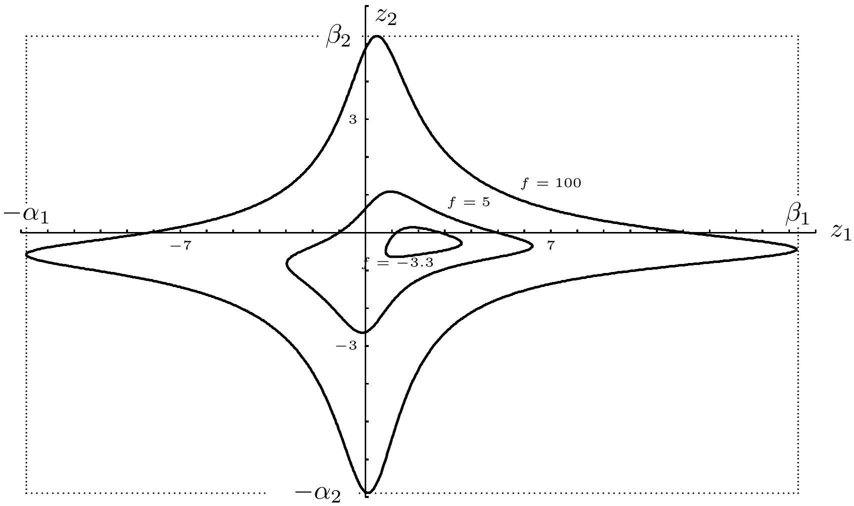


Fig. 4.6 Contour lines of the function (4.49)

Example 4.7. Our goal is to obtain the fuzzy rules for P2-TS system which exactly model the following nonlinear function

$$f(z_1, z_2) = 2z_1^2 z_2^2 + 2z_1^2 z_2 - z_1 z_2^2 + z_1^2 - 5z_1 z_2 + 3z_2^2 - 4z_1 + 6z_2 \tag{4.49}$$

for $(z_1, z_2) \in D^2 = [-12.8040, 16.2860] \times [-6.8844, 5.2029]$. Three contour lines of the above function as the set of points

$$\bigcup_{c \in \{-3.3,\ 5,\ 100\}} \left\{(z_1, z_2) \in D^2 : f(z_1, z_2) = c\right\}$$

are shown in Fig. 4.6. We assume that the first input z_1 of the TS system has assigned the fuzzy sets N_1, Z_1 and P_1, whereas the second one - the fuzzy sets N_2, Z_2 and P_2. The membership functions are defined by (4.24)-(4.26), with the parameters $\lambda_1 = \lambda_2 = 1$, and boundaries of the intervals $\alpha_1 = 12.8040$, $\beta_1 = 16.2860$, $\alpha_2 = 6.8844$, and $\beta_2 = 5.2029$. The cores of fuzzy sets Z_1 and Z_2 are $\sigma_1 = 1.7410$ and $\sigma_2 = -0.8407$, respectively. Observe that the function (4.49) can be written equivalently as

$$f(z_1, z_2) = \boldsymbol{\theta}^T \mathbf{g}_2(z_1, z_2) = [0, -4, 1, 6, -5, 2, 3, -1, 2]\, \mathbf{g}_2(z_1, z_2),$$

where the generator $\mathbf{g}_2(z_1, z_2)$ is given by (4.33). Taking from (4.22) the fundamental matrix $\mathbf{\Omega}_1$ for one-input P2-TS system, we compute the fundamental matrix $\mathbf{\Omega}_2$ for two-inputs P2-TS system, according to Theorem 4.6 (for $\lambda_1 = \lambda_2 = 1$). After computations we obtain

$$\mathbf{\Omega}_2^T = \begin{bmatrix} 1 & -\alpha_1 & \alpha_1^2 & -\alpha_2 & \alpha_1\alpha_2 & -\alpha_1^2\alpha_2 & \alpha_2^2 & -\alpha_1\alpha_2^2 & \alpha_1^2\alpha_2^2 \\ 1 & \sigma_1 & -\alpha_1\beta_1 & -\alpha_2 & -\sigma_1\alpha_2 & \alpha_1\alpha_2\beta_1 & \alpha_2^2 & \sigma_1\alpha_2^2 & -\alpha_1\alpha_2^2\beta_1 \\ 1 & \beta_1 & \beta_1^2 & -\alpha_2 & -\alpha_2\beta_1 & -\alpha_2\beta_1^2 & \alpha_2^2 & \alpha_2^2\beta_1 & \alpha_2^2\beta_1^2 \\ 1 & -\alpha_1 & \alpha_1^2 & \sigma_2 & -\alpha_1\sigma_2 & \alpha_1^2\sigma_2 & -\alpha_2\beta_2 & \alpha_1\alpha_2\beta_2 & -\alpha_1^2\alpha_2\beta_2 \\ 1 & \sigma_1 & -\alpha_1\beta_1 & \sigma_2 & \sigma_1\sigma_2 & -\alpha_1\beta_1\sigma_2 & -\alpha_2\beta_2 & -\sigma_1\alpha_2\beta_2 & \alpha_1\alpha_2\beta_1\beta_2 \\ 1 & \beta_1 & \beta_1^2 & \sigma_2 & \beta_1\sigma_2 & \beta_1^2\sigma_2 & -\alpha_2\beta_2 & -\alpha_2\beta_1\beta_2 & -\alpha_2\beta_1^2\beta_2 \\ 1 & -\alpha_1 & \alpha_1^2 & \beta_2 & -\alpha_1\beta_2 & \alpha_1^2\beta_2 & \beta_2^2 & -\alpha_1\beta_2^2 & \alpha_1^2\beta_2^2 \\ 1 & \sigma_1 & -\alpha_1\beta_1 & \beta_2 & \sigma_1\beta_2 & -\alpha_1\beta_1\beta_2 & \beta_2^2 & \sigma_1\beta_2^2 & -\alpha_1\beta_1\beta_2^2 \\ 1 & \beta_1 & \beta_1^2 & \beta_2 & \beta_1\beta_2 & \beta_1^2\beta_2 & \beta_2^2 & \beta_1\beta_2^2 & \beta_1^2\beta_2^2 \end{bmatrix}, \tag{4.50}$$

and numerically

$$\mathbf{\Omega}_2^T = \begin{bmatrix} 1 & -12.80 & 163.94 & -6.884 & 88.15 & -1128.6 & 47.39 & -606.85 & 7770.0 \\ 1 & 1.74 & -208.53 & -6.884 & -11.99 & 1435.6 & 47.39 & 82.515 & -9883.1 \\ 1 & 16.29 & 265.23 & -6.884 & -112.1 & -1826.0 & 47.39 & 771.87 & 12571. \\ 1 & -12.80 & 163.94 & -0.841 & 10.76 & -137.83 & -35.82 & 458.62 & -5872.2 \\ 1 & 1.74 & -208.53 & -0.841 & -1.464 & 175.31 & -35.82 & -62.361 & 7469.2 \\ 1 & 16.29 & 265.23 & -0.841 & -13.69 & -222.98 & -35.82 & -583.35 & -9500.4 \\ 1 & -12.80 & 163.94 & 5.203 & -66.62 & 852.98 & 27.07 & -346.61 & 4437.9 \\ 1 & 1.74 & -208.53 & 5.203 & 9.058 & -1084.9 & 27.07 & 47.129 & -5644.8 \\ 1 & 16.29 & 265.23 & 5.203 & 84.73 & 1380.0 & 27.07 & 440.86 & 7179.9 \end{bmatrix}.$$

For the P2-TS systems we have

$$S_2 = \mathbf{g}_2^T(z_1, z_2)\left(\mathbf{\Omega}_2^T\right)^{-1}\mathbf{q}_2 = f(z_1, z_2) = \boldsymbol{\theta}^T\mathbf{g}_2(z_1, z_2).$$

Thus, the vector of conclusions of the fuzzy rules is given by

$$\begin{aligned} \mathbf{q}_2 &= \mathbf{\Omega}_2^T\boldsymbol{\theta} \\ &= [13764.9420,\ -17032.2043,\ 21579.2316,\ -12429.8971,\ 15030.6206, \\ &\quad -18707.3075,\ 11589.1320,\ -13655.0266,\ 16567.7977]^T. \end{aligned}$$

Finally, the system of fuzzy rules for the 2-iputs-1-output P2-TS system is as follows

$$\left.\begin{array}{l} R_1: \text{ If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = 13764.9420, \\ R_2: \text{ If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = -17032.2043, \\ R_3: \text{ If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = 21579.2316, \\ R_4: \text{ If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = -12429.8971, \\ R_5: \text{ If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = 15030.6206, \\ R_6: \text{ If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } Z_2, \text{ then } S = -18707.3075, \\ R_7: \text{ If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = 11589.1320, \\ R_8: \text{ If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = -13655.0266, \\ R_9: \text{ If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = 16567.7977. \end{array}\right\} \tag{4.51}$$

One can check that the above rule-based system exactly models the function (4.49), since the expression $\mathbf{g}_2^T(\mathbf{z})\left(\boldsymbol{\Omega}_2^T\right)^{-1}\mathbf{q}_2$ results in the same polynomial as in (4.49) for all points $\mathbf{z}$ from the rectangle D^2.

Example 4.8. Let us consider the system of fuzzy rules (4.51) for 2-inputs-one-output P2-TS system from Example 4.7. Assume the same data $\alpha_1 = 12.8040$, $\beta_1 = 16.2860$, $\alpha_2 = 6.8844$, $\beta_2 = 5.2029$, $\lambda_1 = \lambda_2 = 1$ and the consequents of the rules (4.51): $q_{00} = 13764.9420$, $q_{10} = -17032.2043$, $q_{20} = 21579.2316$, $q_{01} = -12429.8971$, $q_{11} = 15030.6206$, $q_{21} = -18707.3075$, $q_{02} = 11589.1320$, $q_{12} = -13655.0266$ and $q_{22} = 16567.7977$. From (4.24)-(4.26) and (4.34)-(4.35) we obtain the system output $S = S_2(z_1, z_2 \mid q_{00}, \ldots, q_{22})$ which can be expressed by

$$\begin{aligned} S = {} & N_2(z_2)\left(N_1(z_1)\,q_{00} + Z_1(z_1)\,q_{10} + P_1(z_1)\,q_{20}\right) \\ & + Z_2(z_2)\left(N_1(z_1)\,q_{01} + Z_1(z_1)\,q_{11} + P_1(z_1)\,q_{21}\right) \\ & + P_2(z_2)\left(N_1(z_1)\,q_{02} + Z_1(z_1)\,q_{12} + P_1(z_1)\,q_{22}\right). \end{aligned}$$

The above expression gives the same function as in (4.49) exact to numerical errors.

4.6 Recursion in MISO P2-TS Systems

In order to obtain the crisp output of a MISO P2-TS system, we need to obtain an inverse of the fundamental matrix. Our first goal is to give a procedure for computing this inverse. We prove the following

Theorem 4.9. *Let $\boldsymbol{\Omega}_0 = 1$ and $\boldsymbol{\Omega}_n$ be the fundamental matrix of the P2-TS system with n inputs, ($n \geq 1$). The inverse of the fundamental matrix can be computed as follows*

$$\mathbf{\Omega}_n^{-1} = \frac{1}{L_n^2} \begin{bmatrix} \beta_n \left(L_n - \alpha_n \lambda_n\right) & -L_n + \left(\alpha_n - \beta_n\right) \lambda_n & \lambda_n \\ 2\alpha_n \beta_n \lambda_n & 4\sigma_n \lambda_n & -2\lambda_n \\ \alpha_n \left(L_n - \beta_n \lambda_n\right) & L_n + \left(\alpha_n - \beta_n\right) \lambda_n & \lambda_n \end{bmatrix} \otimes \mathbf{\Omega}_{n-1}^{-1}, \tag{4.52}$$

where $L_n = \alpha_n + \beta_n$.

Proof. Taking into account Theorem 4.6, the Kronecker product property (A.4) from Appendix A, the equalities $\sigma_n = \left(-\alpha_n + \beta_n\right)/2$ and $L_n = \alpha_n + \beta_n$, we have

$$\mathbf{\Omega}_n^{-1} = \left(\begin{bmatrix} 1 & 1 & 1 \\ -\alpha_n & \sigma_n & \beta_n \\ \left(-\alpha_n\right)^2 & \frac{1}{2}\left(\alpha_n^2 + \beta_n^2 - \frac{\left(\alpha_n + \beta_n\right)^2}{\lambda_n}\right) & \beta_n^2 \end{bmatrix} \otimes \mathbf{\Omega}_{n-1} \right)^{-1}.$$

Thus,

$$\mathbf{\Omega}_n^{-1} = \frac{1}{L_n^2} \begin{bmatrix} \beta_n \left(L_n - \alpha_n \lambda_n\right) & -L_n + \left(\alpha_n - \beta_n\right) \lambda_n & \lambda_n \\ 2\alpha_n \beta_n \lambda_n & 2\lambda_n \left(\beta_n - \alpha_n\right) & -2\lambda_n \\ \alpha_n \left(L_n - \beta_n \lambda_n\right) & L_n + \left(\alpha_n - \beta_n\right) \lambda_n & \lambda_n \end{bmatrix} \otimes \mathbf{\Omega}_{n-1}^{-1}.$$

The last matrix is the same as in (4.52), because $2\lambda_n \left(\beta_n - \alpha_n\right) = 4\sigma_n \lambda_n$. This finishes the proof of Theorem 4.9. □

For $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$ we obtain a much simpler recurrence

$$\begin{aligned} \mathbf{\Omega}_0 &= 1, \\ \mathbf{\Omega}_n^{-1} &= \frac{1}{L_n^2} \begin{bmatrix} \beta_n^2 & -2\beta_n & 1 \\ 2\alpha_n \beta_n & 4\sigma_n & -2 \\ \alpha_n^2 & 2\alpha_n & 1 \end{bmatrix} \otimes \mathbf{\Omega}_{n-1}^{-1}, \qquad n = 1, 2, \ldots, \end{aligned} \tag{4.53}$$

which can be used in practice.

4.6.1 Rule-Base Decomposition

Without loss of generality we will consider a zero-order TS system with one output. The inputs are components of the vector $\mathbf{z} = \left[z_1, \ldots, z_n\right]^T \in D^n$, $(n = 2, 3, \ldots)$. We assume that three polynomial membership functions $N_k\left(z_k\right)$, $Z_k\left(z_k\right)$ and $P_k\left(z_k\right)$ defined by (4.24)-(4.26), are assigned for every input z_k, $(k = 1, \ldots, n)$.

The complete and noncontradictory rule-base is defined by the following 3^n "If-then" fuzzy rules:

$$\left.\begin{array}{l} R_1 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{0,0,\ldots,0,0}, \\ R_2 : \text{If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{1,0,\ldots,0,0}, \\ R_3 : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{2,0,\ldots,0,0}, \\ R_4 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } Z_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{0,1,\ldots,0,0}, \\ \vdots \\ R_{3^n} : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{2,2,\ldots,2,2}. \end{array}\right\} \tag{4.54}$$

One can decompose this system into the following three subsystems:

$$\left.\begin{array}{l} \left.\begin{array}{l} R_1 : \quad \text{If } \mathcal{P}_1 \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{0,\ldots,0,0}, \\ \vdots \\ R_{3^{n-1}} : \quad \text{If } \mathcal{P}_{3^{n-1}} \text{ and } z_n \text{ is } N_n, \text{ then } S = q_{2,\ldots,2,0}, \end{array}\right\} \\ \left.\begin{array}{l} R_{3^{n-1}+1} : \text{If } \mathcal{P}_1 \text{ and } z_n \text{ is } Z_n, \text{ then } S = q_{0,\ldots,0,1}, \\ \vdots \\ R_{2\cdot 3^{n-1}} : \text{If } \mathcal{P}_{3^{n-1}} \text{ and } z_n \text{ is } Z_n, \text{ then } S = q_{2,\ldots,2,1}, \end{array}\right\} \\ \left.\begin{array}{l} R_{2\cdot 3^{n-1}+1} : \text{If } \mathcal{P}_1 \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{0,\ldots,0,2}, \\ \vdots \\ R_{3^n} : \text{If } \mathcal{P}_{3^{n-1}} \text{ and } z_n \text{ is } P_n, \text{ then } S = q_{2,\ldots,2,2}, \end{array}\right\} \end{array}\right\} \tag{4.55}$$

where $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{3^{n-1}}$ are "If" parts in the system with $(n-1)$ inputs $[z_1, \ldots, z_{n-1}]^T \in D^{n-1}$, $(n = 2, 3, \ldots)$:

$$\left.\begin{array}{l} R'_1 : \underbrace{\text{If } z_1 \text{ is } N_1 \text{ and } \ldots \text{ and } z_{n-1} \text{ is } N_{n-1}}_{\mathcal{P}_1}, \text{ then } S = q_{0,0,\ldots,0}, \\ \vdots \\ R'_{3^{n-1}} : \underbrace{\text{If } z_1 \text{ is } P_1 \text{ and } \ldots \text{ and } z_{n-1} \text{ is } P_{n-1}}_{\mathcal{P}_{3^{n-1}}}, \text{ then } S = q_{2,2,\ldots,2}. \end{array}\right\} \tag{4.56}$$

The decomposition (4.55) of the original P2-TS system (4.54) will be used for proving the most important recurrence for such systems.

4.6.2 Crisp Output Calculation for P2-TS System Using Recursion

Now we prove the following

Theorem 4.10. ***(on recursion in systems with membership functions as second degree polynomials)*** *The recursive formula that enables one to compute the crisp output for any P2-TS system with inputs $z_1 \in [-\alpha_1, \beta_1]$, $\ldots$, $z_n \in [-\alpha_n, \beta_n]$, for $n = 2, 3, \ldots$, is as follows*

$$\begin{aligned} S_n(\mathbf{z} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}) = {} & N_n(z_n)\, S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,0}) \\ & + Z_n(z_n)\, S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,1}, \ldots, q_{2,\ldots,2,1}) \\ & + P_n(z_n)\, S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,2}, \ldots, q_{2,\ldots,2,2}), \end{aligned} \tag{4.57}$$

where

- $\mathbf{z}_{n-1} = [z_1, \ldots, z_{n-1}]^T \in D^{n-1}$ *and* $\mathbf{z} = \begin{bmatrix} \mathbf{z}_{n-1} \\ z_n \end{bmatrix} \in D^n$ *are the input vectors,*
- $S_n(\mathbf{z} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2})$ *is the crisp output of the system (4.54) with input vector* $\mathbf{z} \in D^n$ *and the consequents of the rules constituting the vector* $[q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}]^T$,
- $N_n(z_n)$, $Z_n(z_n)$ *and* $P_n(z_n)$ *are membership functions for the input* $z_n \in [-\alpha_n, \beta_n]$ *defined by (4.24)-(4.26),*
- $S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,0})$ *is the crisp output of the first subsystem in (4.55) with input vector* $\mathbf{z}_{n-1} \in D^{n-1}$ *and the consequents of the rules constituting the vector* $[q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,0}]^T$,
- $S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,1}, \ldots, q_{2,\ldots,2,1})$ *is the crisp output of the second subsystem in (4.55) with input vector* $\mathbf{z}_{n-1} \in D^{n-1}$ *and the consequents of the rules constituting the vector* $[q_{0,\ldots,0,1}, \ldots, q_{2,\ldots,2,1}]^T$,
- $S_{n-1}(\mathbf{z}_{n-1} \mid q_{0,\ldots,0,2}, \ldots, q_{2,\ldots,2,2})$ *is the crisp output of the third subsystem in (4.55) with input vector* $\mathbf{z}_{n-1} \in D^{n-1}$ *and the consequents of the rules constituting the vector* $[q_{0,\ldots,0,2}, \ldots, q_{2,\ldots,2,2}]^T$.

Proof. We will use notation of Theorem 4.10. The rules for SISO P2-TS system are given by (4.12). According to (4.13) the system output is as follows

$$\begin{aligned} S_1(z_1 \mid a, b, c) &= N_1(z_1)\, a + Z_1(z_1)\, b + P_1(z_1)\, c \\ &= [N_1(z_1), Z_1(z_1), P_1(z_1)]\, [a, b, c]^T \end{aligned}$$

First we prove theorem for $n = 2$. The rules for P2-TS system are given by (4.32), where the consequents of the fuzzy rules constitute the vector $\mathbf{q} = [q_{00}, q_{10}, q_{20}, q_{01}, q_{11}, q_{21}, q_{02}, q_{12}, q_{22}]^T$. According to (4.57) and (4.35) the system output is as follows

$$\begin{aligned}
S_2(z_1, z_2 \mid q_{00}, \ldots, q_{22}) &= N_2(z_2) S_1(z_1 \mid q_{00}, q_{10}, q_{20}) \\
&+ Z_2(z_2) S_1(z_1 \mid q_{01}, q_{11}, q_{21}) \\
&+ P_2(z_2) S_1(z_1 \mid q_{02}, q_{12}, q_{22}) \\
&= N_2(z_2) [N_1(z_1), Z_1(z_1), P_1(z_1)] [q_{00}, q_{10}, q_{20}]^T \\
&+ Z_2(z_2) [N_1(z_1), Z_1(z_1), P_1(z_1)] [q_{01}, q_{11}, q_{21}]^T \\
&+ P_2(z_2) [N_1(z_1), Z_1(z_1), P_1(z_1)] [q_{02}, q_{12}, q_{22}]^T .
\end{aligned}$$

The last formula gives the same result as the scalar product (4.34). This implies that Theorem 4.10 is true for P2-TS systems with $n = 2$ inputs.

The output of the MISO P2-TS system defined by the rules (4.54), can be expressed as follows

$$S_n = S_n(\mathbf{z} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}) = \mathbf{q}_n^T \mathbf{\Omega}_n^{-1} \mathbf{g}_n(\mathbf{z}), \tag{4.58}$$

where $\mathbf{q}_n^T = [q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}]$ is the vector of consequents of the rules (4.54), $\mathbf{\Omega}_n$ is the fundamental matrix, and $\mathbf{g}_n(\mathbf{z})$ is the generator of the system with n-inputs $z_1, \ldots, z_n$. Taking into account (4.58), the Kronecker product properties (A.4) and (A.2c) from Appendix A, the equalities $\sigma_n = (-\alpha_n + \beta_n)/2$ and $L_n = \alpha_n + \beta_n$, we obtain

$$\begin{aligned}
S_n &= \mathbf{q}_n^T \left(\begin{bmatrix} 1 & 1 & 1 \\ -\alpha_n & \sigma_n & \beta_n \\ (-\alpha_n)^2 & \frac{1}{2}\left(\alpha_n^2 + \beta_n^2 - \frac{(\alpha_n + \beta_n)^2}{\lambda_n}\right) & \beta_n^2 \end{bmatrix} \otimes \mathbf{\Omega}_{n-1} \right)^{-1} \mathbf{g}_n(\mathbf{z}) \\
&= \mathbf{q}_n^T \left(\begin{bmatrix} 1 & 1 & 1 \\ -\alpha_n & \sigma_n & \beta_n \\ (-\alpha_n)^2 & \frac{1}{2}\left(\alpha_n^2 + \beta_n^2 - \frac{(\alpha_n + \beta_n)^2}{\lambda_n}\right) & \beta_n^2 \end{bmatrix}^{-1} \otimes \mathbf{\Omega}_{n-1}^{-1} \right) \mathbf{g}_n(\mathbf{z}) \\
&= \frac{\mathbf{q}_n^T}{L_n^2} \left(\begin{bmatrix} \beta_n(L_n - \alpha_n\lambda_n) & (\alpha_n - \beta_n)\lambda_n - L_n & \lambda_n \\ 2\alpha_n\beta_n\lambda_n & 2\lambda_n(\beta_n - \alpha_n) & -2\lambda_n \\ \alpha_n(L_n - \beta_n\lambda_n) & (\alpha_n - \beta_n)\lambda_n + L_n & \lambda_n \end{bmatrix} \otimes \mathbf{\Omega}_{n-1}^{-1} \right) \mathbf{g}_n(\mathbf{z}).
\end{aligned}$$

According to the definition (4.31) of the generator for P2-TS system, we have

$$\begin{aligned}
S_n &= \frac{\mathbf{q}_n^T}{L_n^2} \begin{bmatrix} \beta_n(L_n - \alpha_n\lambda_n)\mathbf{\Omega}_{n-1}^{-1} & ((\alpha_n - \beta_n)\lambda_n - L_n)\mathbf{\Omega}_{n-1}^{-1} & \lambda_n\mathbf{\Omega}_{n-1}^{-1} \\ 2\alpha_n\beta_n\lambda_n\mathbf{\Omega}_{n-1}^{-1} & 2\lambda_n(\beta_n - \alpha_n)\mathbf{\Omega}_{n-1}^{-1} & -2\lambda_n\mathbf{\Omega}_{n-1}^{-1} \\ \alpha_n(L_n - \beta_n\lambda_n)\mathbf{\Omega}_{n-1}^{-1} & ((\alpha_n - \beta_n)\lambda_n + L_n)\mathbf{\Omega}_{n-1}^{-1} & \lambda_n\mathbf{\Omega}_{n-1}^{-1} \end{bmatrix} \\
&\times \begin{bmatrix} \mathbf{g}_{n-1}(z_1, \ldots, z_{n-1}) \\ z_n\mathbf{g}_{n-1}(z_1, \ldots, z_{n-1}) \\ z_n^2\mathbf{g}_{n-1}(z_1, \ldots, z_{n-1}) \end{bmatrix}.
\end{aligned}$$

Let us denote

$$\mathbf{q}_n^T = \left[\mathbf{a}^T, \mathbf{b}^T, \mathbf{c}^T\right],$$

where **a**, **b**, and **c** correspond to the three consecutive parts of the conclusions in the decomposed system (4.55), i.e.

$$\mathbf{a} = \begin{bmatrix} q_{0,\dots,0,0} \\ \vdots \\ q_{2,\dots,2,0} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} q_{0,\dots,0,1} \\ \vdots \\ q_{2,\dots,2,1} \end{bmatrix}, \qquad \mathbf{c} = \begin{bmatrix} q_{0,\dots,0,2} \\ \vdots \\ q_{2,\dots,2,2} \end{bmatrix}.$$

According to (4.58) for the MISO P2-TS system with the inputs $\mathbf{z}_{n-1} \in D^{n-1}$, the crisp outputs S_{n-1} can be expressed as follows

$$\begin{aligned} S_{n-1}\left(\mathbf{z}_{n-1} \mid q_{0,\dots,0,0}, \dots, q_{2,\dots,2,0}\right) &= \left[q_{0,\dots,0,0}, \dots, q_{2,\dots,2,0}\right] \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ &= \mathbf{a}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right), \end{aligned} \tag{4.59}$$

$$\begin{aligned} S_{n-1}\left(\mathbf{z}_{n-1} \mid q_{0,\dots,0,1}, \dots, q_{2,\dots,2,1}\right) &= \left[q_{0,\dots,0,1}, \dots, q_{2,\dots,2,1}\right] \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ &= \mathbf{b}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right), \end{aligned} \tag{4.60}$$

$$\begin{aligned} S_{n-1}\left(\mathbf{z}_{n-1} \mid q_{0,\dots,0,2}, \dots, q_{2,\dots,2,2}\right) &= \left[q_{0,\dots,0,2}, \dots, q_{2,\dots,2,2}\right] \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ &= \mathbf{c}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right). \end{aligned} \tag{4.61}$$

Thus,

$$\begin{aligned} S_n\left(\mathbf{z} \mid \mathbf{q}_n\right) = & \frac{\beta_n\left(L_n - \alpha_n\lambda_n\right) + \left(-L_n + \left(\alpha_n - \beta_n\right)\lambda_n\right) z_n + \lambda_n z_n^2}{L_n^2} \\ & \times \mathbf{a}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ & + \frac{2\alpha_n\beta_n\lambda_n + 2\lambda_n\left(\beta_n - \alpha_n\right) z_n - 2\lambda_n z_n^2}{L_n^2} \mathbf{b}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ & + \frac{\alpha_n\left(L_n - \beta_n\lambda_n\right) + \left(L_n + \left(\alpha_n - \beta_n\right)\lambda_n\right) z_n + \lambda_n z_n^2}{L_n^2} \\ & \times \mathbf{c}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right). \end{aligned}$$

Taking into consideration (4.24)-(4.26) we obtain

$$\begin{aligned} S_n\left(\mathbf{z} \mid \mathbf{q}_n\right) = & N_n\left(z_n\right) \mathbf{a}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ & + Z_n\left(z_n\right) \mathbf{b}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right) \\ & + P_n\left(z_n\right) \mathbf{c}^T \mathbf{\Omega}_{n-1}^{-1} \mathbf{g}_{n-1}\left(\mathbf{z}_{n-1}\right). \end{aligned}$$

Finally, according to equations (4.59)-(4.61) we obtain (4.57) and this ends the proof of Theorem 4.10. □

The above theorem is important, because it says that we do not need to inverse large matrices to obtain the crisp output of the P2-TS systems. As a result of this theorem the curse of dimensionality in P2-TS systems is going to disappear. A generalization of Theorem 4.10 for MIMO systems is straightforward and will be omitted.

Now we generalize Corollary 4.3 for MISO P2-TS systems.

Theorem 4.11. *The crisp output of the MISO P2-TS system in the vertex of the hypercuboid D^n is exactly the same as the appropriate conclusion of the fuzzy rule contained in the rule-base.*

Proof. The crisp output of the MISO P2-TS system with the input vector $\mathbf{z} = [z_1, \ldots, z_n]^T$, for which consequents of the rules constitute the vector $\mathbf{q} = [q_{0,\ldots,0}, \ldots, q_{p_1,\ldots,p_n}, \ldots, q_{2,\ldots,2}]^T$, can be expressed as follows

$$S(\mathbf{z} \mid \mathbf{q}) = \sum_{(p_1,\ldots,p_n)\in\{0,1,2\}^n} q_{p_1,\ldots,p_n} \prod_{k=1}^{n} A_{p_k}(z_k), \tag{4.62}$$

where $q_{p_1,\ldots,p_n}$ is a consequent of the fuzzy rule and $A_{p_k}(z_k)$ is the membership degree to which input z_k belongs to A_{p_k}. The name of the membership function A_{p_k} in (4.62) depends on the index $p_k \in \{0,1,2\}$ as follows

$$A_{p_k} = \begin{cases} N_k & \text{for } p_k = 0 \\ Z_k & \text{for } p_k = 1 \\ P_k & \text{for } p_k = 2 \end{cases}, \qquad k = 1, \ldots, n. \tag{4.63}$$

If the input vector is a fixed vertex $\boldsymbol{\gamma}_v$ of the hypercuboid D^n, i.e.

$$\mathbf{z} = \boldsymbol{\gamma}_v = [\gamma_1, \ldots, \gamma_n]^T \in \{-\alpha_1, \beta_1\} \times \ldots \times \{-\alpha_n, \beta_n\},$$

then the equation (4.62) reduces to

$$S(\gamma_1, \ldots, \gamma_n \mid \mathbf{q}) = \sum_{(p_1,\ldots,p_n)\in\{0,2\}^n} q_{p_1,\ldots,p_n} \prod_{k=1}^{n} A_{p_k}(\gamma_k), \tag{4.64}$$

since $\prod_{k=1}^{n} A_{p_k}(\gamma_k) = 0$ by $\gamma_k \in \{-\alpha_k, \beta_k\}$ if among indices at least one index $p_k = 1$, $(k = 1, \ldots, n)$. This follows from (4.63) and (4.25). In the summation (4.64) if $\gamma_k = -\alpha_k$, then $p_k = 0$, and if $\gamma_k = \beta_k$, then $p_k = 2$, $(k = 1, \ldots, n)$, but in both cases $\prod_{k=1}^{n} A_{p_k}(\gamma_k) = 1$ according to (4.24) and (4.26). Finally, taking into account the bijection (4.28) we obtain the complete proof of Theorem 4.11. □

It should be noticed that we are able to choose the consequents of the rules so that, the crisp output of a given P2-TS system will be exactly the same as the appropriate conclusions of its fuzzy rules, not only in 2^n vertices of the hypercuboid D^n, but also in all 3^n characteristic points of the set M_n defined in (4.27). However, the class of crisp functions to which such P2-TS system is equivalent becomes much simpler than expected for systems with membership functions as the second degree polynomials.

Example 4.12. Let us consider the P2-TS system with 2 inputs $z_1 \in [-\alpha_1, \beta_1]$ and $z_2 \in [-\alpha_2, \beta_2]$ with quadratic membership functions of fuzzy sets as in (4.24)-(4.26) by $\lambda_k \in (0, 1]$, $(k = 1, 2)$. If this system is defined by the following fuzzy rules:

R_1 : If z_1 is N_1 and z_2 is N_2, then $S = q_{00}$,
R_2 : If z_1 is Z_1 and z_2 is N_2, then $S = q_{10} = (q_{00} + q_{20})/2$,
R_3 : If z_1 is P_1 and z_2 is N_2, then $S = q_{20}$,
R_4 : If z_1 is N_1 and z_2 is Z_2, then $S = q_{01} = (q_{00} + q_{02})/2$,
R_5 : If z_1 is Z_1 and z_2 is Z_2, then $S = q_{11} = (q_{00} + q_{02} + q_{20} + q_{22})/4$,
R_6 : If z_1 is P_1 and z_2 is Z_2, then $S = q_{21} = (q_{20} + q_{22})/2$,
R_7 : If z_1 is N_1 and z_2 is P_2, then $S = q_{02}$,
R_8 : If z_1 is Z_1 and z_2 is P_2, then $S = q_{12} = (q_{02} + q_{22})/2$,
R_9 : If z_1 is P_1 and z_2 is P_2, then $S = q_{22}$,

then

(i) The crisp output of this system as a function of the inputs $S(z_1, z_2)$ takes the same values in all points of the set $M_2 = \{-\alpha_1, \sigma_1, \beta_1\} \times \{-\alpha_2, \sigma_2, \beta_2\}$, as appear in the appropriate conclusions of the fuzzy rules, i.e.

$$S(-\alpha_1, -\alpha_2) = q_{00}, \qquad S(\sigma_1, -\alpha_2) = q_{10}, \qquad S(\beta_1, -\alpha_2) = q_{20},$$

$$S(-\alpha_1, \sigma_2) = q_{01}, \qquad S(\sigma_1, \sigma_2) = q_{11}, \qquad S(\beta_1, \sigma_2) = q_{21},$$

$$S(-\alpha_1, \beta_2) = q_{02}, \qquad S(\sigma_1, \beta_2) = q_{12}, \qquad S(\beta_1, \beta_2) = q_{22},$$

where $\sigma_k = (-\alpha_k + \beta_k)/2$, $k = 1, 2$.

(ii) The crisp output of this system is equivalent to a simple bilinear function

$$S(z_1, z_2) = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + \theta_{12} z_1 z_2,$$

where

$$\theta_0 = (\alpha_1 + \beta_1)^{-1} (\alpha_2 + \beta_2)^{-1} (q_{00}\beta_1\beta_2 + q_{02}\alpha_2\beta_1 + q_{20}\alpha_1\beta_2 + q_{22}\alpha_1\alpha_2),$$

$$\theta_1 = (\alpha_1 + \beta_1)^{-1} (\alpha_2 + \beta_2)^{-1} (q_{20}\beta_2 - q_{02}\alpha_2 - q_{00}\beta_2 + q_{22}\alpha_2),$$

$$\theta_2 = (\alpha_1 + \beta_1)^{-1} (\alpha_2 + \beta_2)^{-1} (q_{02}\beta_1 - q_{20}\alpha_1 - q_{00}\beta_1 + q_{22}\alpha_1),$$

$$\theta_{12} = (\alpha_1 + \beta_1)^{-1} (\alpha_2 + \beta_2)^{-1} (q_{00} - q_{02} - q_{20} + q_{22}) .$$

Taking into account e.g. the equation (4.34) the proof of the above facts is simple and will be omitted.

Example 4.13. Let us consider a P2-TS system with four inputs which constitute the vector $\mathbf{z} = [z_1, z_2, z_3, z_4]^T \in D^4$, where D^4 is the hypercube $[-1, 1]^4$. The output of the system is S. For every input z_k we assume three membership functions of fuzzy sets: N_k, Z_k and P_k, defined by (4.24)-(4.26) with the parameter $\lambda_k = 1$ for $k = 1, 2, 3, 4$. The system is defined by the following metarules and ordinary rules:

M_1 : If z_2 is N_2 and z_3 is N_3 and z_4 is N_4, then $S = 1$,
M_2 : If z_2 is Z_2 and z_3 is N_3 and z_4 is N_4, then $S = 2$,
M_3 : If z_2 is P_2 and z_3 is N_3 and z_4 is N_4, then $S = 3$,
M_4 : If z_1 is N_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is N_4, then $S = 4$,
M_5 : If z_1 is Z_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is N_4, then $S = 5$,
M_6 : If z_1 is P_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is N_4, then $S = 6$,
M_7 : If z_2 is Z_2 and z_3 is Z_3 and z_4 is N_4, then $S = 7$,
M_8 : If z_2 is P_2 and z_3 is Z_3 and z_4 is N_4, then $S = 8$,
M_9 : If z_3 is P_3 and z_4 is N_4, then $S = 9$,
M_{10} : If z_2 is N_2 and z_3 is N_3 and z_4 is Z_4, then $S = -1$,
M_{11} : If z_2 is Z_2 and z_3 is N_3 and z_4 is Z_4, then $S = -2$,
M_{12} : If z_2 is P_2 and z_3 is N_3 and z_4 is Z_4, then $S = -3$,
M_{13} : If z_1 is N_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is Z_4, then $S = -4$,
M_{14} : If z_1 is Z_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is Z_4, then $S = -5$,
M_{15} : If z_1 is P_1 and z_2 is N_2 and z_3 is Z_3 and z_4 is Z_4, then $S = -6$,
M_{16} : If z_2 is Z_2 and z_3 is Z_3 and z_4 is Z_4, then $S = -7$,
M_{17} : If z_2 is P_2 and z_3 is Z_3 and z_4 is Z_4, then $S = -8$,
M_{18} : If z_3 is P_3 and z_4 is Z_4, then $S = -9$,
M_{19} : If z_3 is N_3 and z_4 is P_4, then $S = 1$,
M_{20} : If z_3 is $(Z_3$ or $P_3)$ and z_4 is P_4, then $S = 0$,

We assume that the fragment " z_3 is $(Z_3$ or $P_3)$" in the "If" part of the metarule M_{20} is equivalent to "z_3 is not N_3" and generates two metarules.

The above 20 metarules are equivalent to 81 complete and noncontradictory fuzzy rules with consequents given symbolically in Table 4.1 and numerically in Table 4.2.

Formally the system output $S = S_4 (z_1, z_2, z_3, z_4 \mid q_{0000}, \ldots, q_{2222})$. According to Theorem 4.10 a general form of the crisp system output is given by (4.57) for $n = 4$, i.e.

Table 4.1 Look-up-table for the P2-TS system with $n = 4$ input variables in the general case

$z_1z_2 \setminus z_3z_4$ $\rightarrow$ $\downarrow$	N_3N_4	Z_3N_4	P_3N_4	N_3Z_4	Z_3Z_4	P_3Z_4	N_3P_4	Z_3P_4	P_3P_4
N_1N_2	q_{0000}	q_{0010}	q_{0020}	q_{0001}	q_{0011}	q_{0021}	q_{0002}	q_{0012}	q_{0022}
Z_1N_2	q_{1000}	q_{1010}	q_{1020}	q_{1001}	q_{1011}	q_{1021}	q_{1002}	q_{1012}	q_{1022}
P_1N_2	q_{2000}	q_{2010}	q_{2020}	q_{2001}	q_{2011}	q_{2021}	q_{2002}	q_{2012}	q_{2022}
N_1Z_2	q_{0100}	q_{0110}	q_{0120}	q_{0101}	q_{0111}	q_{0121}	q_{0102}	q_{0112}	q_{0122}
Z_1Z_2	q_{1100}	q_{1110}	q_{1120}	q_{1101}	q_{1111}	q_{1121}	q_{1102}	q_{1112}	q_{1122}
P_1Z_2	q_{2100}	q_{2110}	q_{2120}	q_{2101}	q_{2111}	q_{2121}	q_{2102}	q_{2112}	q_{2122}
N_1P_2	q_{0200}	q_{0210}	q_{0220}	q_{0201}	q_{0211}	q_{0221}	q_{0202}	q_{0212}	q_{0222}
Z_1P_2	q_{1200}	q_{1210}	q_{1220}	q_{1201}	q_{1211}	q_{1221}	q_{1202}	q_{1212}	q_{1222}
P_1P_2	q_{2200}	q_{2210}	q_{2220}	q_{2201}	q_{2211}	q_{2221}	q_{2202}	q_{2212}	q_{2222}

Table 4.2 Look-up-table for the P2-TS system from Example 4.13

$z_1z_2 \setminus z_3z_4$ $\rightarrow$ $\downarrow$	N_3N_4	Z_3N_4	P_3N_4	N_3Z_4	Z_3Z_4	P_3Z_4	N_3P_4	Z_3P_4	P_3P_4
N_1N_2	1	4	9	−1	−4	−9	1	0	0
Z_1N_2	1	5	9	−1	−5	−9	1	0	0
P_1N_2	1	6	9	−1	−6	−9	1	0	0
N_1Z_2	2	7	9	−2	−7	−9	1	0	0
Z_1Z_2	2	7	9	−2	−7	−9	1	0	0
P_1Z_2	2	7	9	−2	−7	−9	1	0	0
N_1P_2	3	8	9	−3	−8	−9	1	0	0
Z_1P_2	3	8	9	−3	−8	−9	1	0	0
P_1P_2	3	8	9	−3	−8	−9	1	0	0

$$\begin{aligned} S = {} & N_4(z_4)\, S_3(z_1, z_2, z_3 \mid q_{0000}, q_{1000}, q_{2000}, \ldots, q_{0220}, q_{1220}, q_{2220}) \\ & + Z_4(z_4)\, S_3(z_1, z_2, z_3 \mid q_{0001}, q_{1001}, q_{2001}, \ldots, q_{0221}, q_{1221}, q_{2221}) \\ & + P_4(z_4)\, S_3(z_1, z_2, z_3 \mid q_{0002}, q_{1002}, q_{2002}, \ldots, q_{0222}, q_{1222}, q_{2222}), \end{aligned} \tag{4.65}$$

where for $S_3 = S_3(z_1, z_2, z_3 \mid q_{000}, \ldots, q_{222})$ we have

$$\begin{aligned} S_3 &= N_3(z_3)\, S_2(z_1, z_2 \mid q_{000}, q_{100}, q_{200}, q_{010}, q_{110}, q_{210}, q_{020}, q_{120}, q_{220}) \\ &+ Z_3(z_3)\, S_2(z_1, z_2 \mid q_{001}, q_{101}, q_{201}, q_{011}, q_{111}, q_{211}, q_{021}, q_{121}, q_{221}) \\ &+ P_3(z_3)\, S_2(z_1, z_2 \mid q_{002}, q_{102}, q_{202}, q_{012}, q_{112}, q_{212}, q_{022}, q_{122}, q_{222}), \end{aligned} \tag{4.66}$$

$$\begin{aligned} S_2(z_1, z_2 \mid q_{00}, q_{10}, q_{20}, q_{01}, q_{11}, q_{21}, q_{02}, q_{12}, q_{22}) &= N_2(z_2)\, S_1(z_1 \mid q_{00}, q_{10}, q_{20}) \\ &+ Z_2(z_2)\, S_1(z_1 \mid q_{01}, q_{11}, q_{21}) \\ &+ P_2(z_2)\, S_1(z_1 \mid q_{02}, q_{12}, q_{22}), \end{aligned} \tag{4.67}$$

$$S_1(z_1 \mid q_0, q_1, q_2) = N_1(z_1)\, q_0 + Z_1(z_1)\, q_1 + P_1(z_1)\, q_2. \tag{4.68}$$

Assume that the membership functions of the fuzzy sets are

$$N_k(z_k) = \frac{(1-z_k)^2}{4}, \quad Z_k(z_k) = \frac{1-z_k^2}{2}, \quad P_k(z_k) = \frac{(1+z_k)^2}{4},$$

for $\alpha_k = \beta_k = 1$ and $\lambda_k = 1$, $(k = 1, 2, 3, 4)$. After computations we obtain

$$\begin{aligned} S &= \frac{1}{32} z_1 z_3^2 - \frac{1}{4} z_2 - z_3 - \frac{47}{16} z_4 - \frac{1}{32} z_1 z_2^2 - \frac{1}{32} z_1 + \frac{3}{32} z_1 z_4^2 + \frac{1}{8} z_2 z_3^2 \\ &+ \frac{3}{4} z_2 z_4^2 + \frac{1}{16} z_2^2 z_4 + \frac{5}{2} z_3 z_4^2 + \frac{7}{16} z_3^2 z_4 + \frac{1}{32} z_2^2 + \frac{7}{32} z_3^2 \\ &+ \frac{149}{32} z_4^2 - \frac{1}{32} z_2^2 z_3^2 - \frac{3}{32} z_2^2 z_4^2 - \frac{13}{32} z_3^2 z_4^2 + \frac{1}{16} z_1 z_2 - \frac{1}{16} z_1 z_4 \\ &+ \frac{1}{8} z_2 z_3 - \frac{1}{2} z_2 z_4 - 2 z_3 z_4 + \frac{3}{32} z_2^2 z_3^2 z_4^2 - \frac{1}{16} z_1 z_2 z_3^2 - \frac{3}{16} z_1 z_2 z_4^2 \\ &- \frac{1}{16} z_1 z_2^2 z_4 + \frac{1}{16} z_1 z_3^2 z_4 - \frac{3}{8} z_2 z_3 z_4^2 + \frac{1}{4} z_2 z_3^2 z_4 + \frac{1}{32} z_1 z_2^2 z_3^2 \\ &+ \frac{3}{32} z_1 z_2^2 z_4^2 - \frac{3}{32} z_1 z_3^2 z_4^2 - \frac{3}{8} z_2 z_3^2 z_4^2 - \frac{1}{16} z_2^2 z_3^2 z_4 + \frac{1}{8} z_1 z_2 z_4 + \frac{1}{4} z_2 z_3 z_4 \\ &+ \frac{3}{16} z_1 z_2 z_3^2 z_4^2 + \frac{1}{16} z_1 z_2^2 z_3^2 z_4 - \frac{3}{32} z_1 z_2^2 z_3^2 z_4^2 - \frac{1}{8} z_1 z_2 z_3^2 z_4 - \frac{47}{32}. \end{aligned}$$

If we consider the output S as a function of four independent variables, i.e. $S = S(z_1, z_2, z_3, z_4)$, we have

$$\begin{aligned} &S(-1,-1,-1,-1) = 1,\ S(1,-1,-1,-1) = 1,\ S(-1,1,-1,-1) = 3, \\ &S(1,1,-1,-1) = 3,\ S(-1,-1,1,-1) = 9,\ S(1,-1,1,-1) = 9, \\ &S(-1,1,1,-1) = 9,\ S(1,1,1,-1) = 9,\ S(-1,-1,-1,1) = 1, \\ &S(1,-1,-1,1) = 1,\ S(-1,1,-1,1) = 1,\ S(1,1,-1,1) = 1, \\ &S(-1,-1,1,1) = 0,\ S(1,-1,1,1) = 0,\ S(-1,1,1,1) = 0,\ S(1,1,1,1) = 0. \end{aligned}$$

This means that in all 2^n points from the set $\times_{k=1}^{n} \{-\alpha_k, \beta_k\}$, $(n = 4)$, the values of the output of the P2-TS system are exactly the same as the

Table 4.3 The metarules M_1, M_2, M_3 and all fuzzy rules ($M_1\&M_2\&M_3\&R_1$) for the first system in Example 4.14 in the form of look-up-tables

$z_1z_2 \setminus z_3$	M_1: N_3	M_1: Z_3	M_1: P_3	M_2: N_3	M_2: Z_3	M_2: P_3	M_3: N_3	M_3: Z_3	M_3: P_3	all rules: N_3	all rules: Z_3	all rules: P_3
N_1N_2	0	0	0	0	0	0	0	$*$	0	0	0	0
Z_1N_2	$*$	$*$	$*$	0	0	0	0	$*$	0	0	0	0
P_1N_2	0	0	0	0	0	0	0	$*$	0	0	0	0
N_1Z_2	0	0	0	$*$	$*$	$*$	0	$*$	0	0	0	0
Z_1Z_2	$*$	$*$	$*$	$*$	$*$	$*$	0	$*$	0	0	a	0
P_1Z_2	0	0	0	$*$	$*$	$*$	0	$*$	0	0	0	0
N_1P_2	0	0	0	0	0	0	0	$*$	0	0	0	0
Z_1P_2	$*$	$*$	$*$	0	0	0	0	$*$	0	0	0	0
P_1P_2	0	0	0	0	0	0	0	$*$	0	0	0	0

appropriate conclusions of the fuzzy rules (see Table 4.2). However, the value $S(z_1, z_2, z_3, z_4)$ in the other points (z_1, z_2, z_3, z_4) from the set M_n defined by (4.27) for $n = 4$, does not satisfy this condition, e.g. $S(-1, -1, 0, -1) = 4.5 \neq 4$. The result confirms the correctness of Theorem 4.11.

Example 4.14. Let us consider two simple P2-TS systems with 3 inputs $z_k \in [-\alpha_k, \beta_k]$ and quadratic membership functions (4.24)-(4.26), for $k = 1, 2, 3$. The first system is given by three metarules M_1-M_3 and one rule R_1:

$$\begin{aligned}
M_1 &: \text{ If } z_1 \text{ is not } Z_1, \text{ then } S = 0,\\
M_2 &: \text{ If } z_2 \text{ is not } Z_2, \text{ then } S = 0,\\
M_3 &: \text{ If } z_3 \text{ is not } Z_3, \text{ then } S = 0,\\
R_1 &: \text{ If } z_1 \text{ is } Z_1 \text{ and } z_2 \text{ is } Z_2 \text{ and } z_3 \text{ is } Z_3, \text{ then } S = a,
\end{aligned}$$

and the second one by three metarules M_1'-M_3' and one rule R_1':

$$\begin{aligned}
M_1' &: \text{ If } z_1 \text{ is not } N_1, \text{ then } S' = 0,\\
M_2' &: \text{ If } z_2 \text{ is not } N_2, \text{ then } S' = 0,\\
M_3' &: \text{ If } z_3 \text{ is not } N_3, \text{ then } S' = 0,\\
R_1' &: \text{ If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_3 \text{ is } N_3, \text{ then } S' = b.
\end{aligned}$$

The meaning of all logical operators "and", "or", "not" used in the "If" parts of the metarules is natural and explained by the look-up-tables (see Tables 4.3 and 4.4). They describe the metarules and all the fuzzy rules. Zero in a table

Table 4.4 The metarules M_1', M_2', M_3' and all fuzzy rules $(M_1'\&M_2'\&M_3'\&R_1')$ for the first system in Example 4.14 in the form of look-up-tables

M_1':

$z_1z_2 \setminus z_3$	N_3	Z_3	P_3
N_1N_2	$*$	$*$	$*$
Z_1N_2	0	0	0
P_1N_2	0	0	0
N_1Z_2	$*$	$*$	$*$
Z_1Z_2	0	0	0
P_1Z_2	0	0	0
N_1P_2	$*$	$*$	$*$
Z_1P_2	0	0	0
P_1P_2	0	0	0

M_2':

$z_1z_2 \setminus z_3$	N_3	Z_3	P_3
N_1N_2	$*$	$*$	$*$
Z_1N_2	$*$	$*$	$*$
P_1N_2	$*$	$*$	$*$
N_1Z_2	0	0	0
Z_1Z_2	0	0	0
P_1Z_2	0	0	0
N_1P_2	0	0	0
Z_1P_2	0	0	0
P_1P_2	0	0	0

M_3':

$z_1z_2 \setminus z_3$	N_3	Z_3	P_3
N_1N_2	$*$	0	0
Z_1N_2	$*$	0	0
P_1N_2	$*$	0	0
N_1Z_2	$*$	0	0
Z_1Z_2	$*$	0	0
P_1Z_2	$*$	0	0
N_1P_2	$*$	0	0
Z_1P_2	$*$	0	0
P_1P_2	$*$	0	0

all rules:

$z_1z_2 \setminus z_3$	N_3	Z_3	P_3
N_1N_2	b	0	0
Z_1N_2	0	0	0
P_1N_2	0	0	0
N_1Z_2	0	0	0
Z_1Z_2	0	0	0
P_1Z_2	0	0	0
N_1P_2	0	0	0
Z_1P_2	0	0	0
P_1P_2	0	0	0

denotes the consequent "0" expressed by some metarule and a star denotes any number (including 0). Observe that the metarules define a complete and noncontradictory system of rules.

One can check that the crisp output of the first system is given by

$$S\left(z_1, z_2, z_3\right) = 8a \prod_{k=1}^{3} \frac{\lambda_k}{\left(\alpha_k + \beta_k\right)^2} \prod_{k=1}^{3} \left(\beta_k - z_k\right)\left(z_k + \alpha_k\right).$$

The sign of S is the same as the sign of the consequent of the rule R_1. Furthermore, $S = 0$ if there is some $k \in \{1, 2, 3\}$ for which $z_k = -\alpha_k$ or $z_k = \beta_k$.

The crisp output of the second system is given by

$$S'\left(z_1, z_2, z_3\right) = b \prod_{k=1}^{3} \frac{\lambda_k}{\left(\alpha_k + \beta_k\right)^2} \prod_{k=1}^{3} \left(\beta_k - z_k\right)\left(\frac{\beta_k + \alpha_k\left(1 - \lambda_k\right)}{\lambda_k} - z_k\right).$$

The sign of the crisp output S' in the second system is the same as the sign of b, since $\frac{1}{\lambda_k}\left(\beta_k + \alpha_k\left(1 - \lambda_k\right)\right) - z_k \geq 0$ and $\left(\beta_k - z_k\right) \geq 0$ for $z_k \in \left[-\alpha_k, \beta_k\right]$, $k = 1, 2, 3$. Furthermore, $S' = 0$ for all points where $z_1 = \beta_1$ or $z_2 = \beta_2$ or $z_3 = \beta_3$.

As one can see, the interpretation of the fuzzy rules in both P2-TS systems is natural and simple. The crisp functions $S\left(z_1, z_2, z_3\right)$ and $S'\left(z_1, z_2, z_3\right)$ intuitively correspond to the systems of rules in any case.

4.7 Recursion in More General TS Systems with Three Fuzzy Sets for Every Input

Theorem 4.10 has been proved using the idea of the fundamental matrix for P2-TS systems, since this matrix is important for many applications. However, we will show below that the same theorem is valid for a more general class of the fuzzy rule-based TS systems, i.e. the systems with three fuzzy sets for every input, where the assumptions 1, 2 and 3 for the membership functions from Section 4.2 are not necessary. We will prove the following generalization of Theorem 4.10.

Theorem 4.15. *Theorem 4.10 is valid for any TS system described by the fuzzy rules (4.54), with the inputs* $z_1 \in [-\alpha_1, \beta_1]$, $\ldots$, $z_n \in [-\alpha_n, \beta_n]$, *where for any input* z_k *there are assigned three fuzzy sets with the normalized membership functions, i.e.* $N_k : [-\alpha_k, \beta_k] \to [0,1]$, $Z_k : [-\alpha_k, \beta_k] \to [0,1]$, *and* $P_k : [-\alpha_k, \beta_k] \to [0,1]$ *and* $N_k(z_k) + Z_k(z_k) + P_k(z_k) = 1$ *for* $k = 1, \ldots, n$. *This means that if* $S_n(\mathbf{z} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2})$ *denotes the crisp output of the system (4.54) with input vector* $\mathbf{z} \in D^n$ *and the consequents of the rules constituting the vector* $[q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}]^T$, *then for any natural* $n \geq 2$ *the recursive formula that enables one to compute the crisp system output is the same as (4.57).*

Proof. For $n = 1$ the system is defined by the rules (4.12). Thus, the system output is as follows

$$S_1(z_1 \mid q_0, q_1, q_2) = \frac{N_1(z_1)\,q_0}{N_1(z_1) + Z_1(z_1) + P_1(z_1)} + \frac{Z_1(z_1)\,q_1}{N_1(z_1) + Z_1(z_1) + P_1(z_1)} + \frac{P_1(z_1)\,q_2}{N_1(z_1) + Z_1(z_1) + P_1(z_1)}. \tag{4.69}$$

It is the same as in (4.68) since the normalization condition (4.7) is satisfied. Let us use a simplified notation: $N_k(z_k) = N_k$, $Z_k(z_k) = Z_k$ and $P_k(z_k) = P_k$. For $n = 2$, due to the rule-base (4.32) we have

$$\begin{aligned} S_2(z_1, z_2 \mid q_{00}, \ldots, q_{22}) = {} & N_1N_2q_{00}/D_2 + Z_1N_2q_{10}/D_2 + P_1N_2q_{20}/D_2 \\ & + N_1Z_2q_{01}/D_2 + Z_1Z_2q_{11}/D_2 + P_1Z_2q_{21}/D_2 \\ & + N_1P_2q_{02}/D_2 + Z_1P_2q_{12}/D_2 + P_1P_2q_{22}/D_2. \end{aligned}$$

But $D_2 = \prod_{k=1}^{2}(N_k(z_k) + Z_k(z_k) + P_k(z_k)) = 1$. Thus,

$$\begin{aligned} S_2(z_1, z_2 \mid q_{00}, \ldots, q_{22}) = {} & N_2(N_1q_{00} + Z_1q_{10} + P_1q_{20}) \\ & + Z_2(N_1q_{01} + Z_1q_{11} + P_1q_{21}) \\ & + P_2(N_1q_{02} + Z_1q_{12} + P_1q_{22}) \end{aligned}$$

and S_2 is the same as in (4.67), i.e. for $n = 2$ the Theorem 4.15 is true.

According to the rule-base decomposition (4.55) for $n = k+1 \geq 3$ we obtain

$$\begin{aligned} S_{k+1} &= N_{k+1}\left(N_1 N_2 \ldots N_k q_{0,0,\ldots,0,0} + \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,0}\right)/D_{k+1} \\ &+ Z_{k+1}\left(N_1 N_2 \ldots N_k q_{0,0,\ldots,0,1} + \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,1}\right)/D_{k+1} \\ &+ P_{k+1}\left(N_1 N_2 \ldots N_k q_{0,0,\ldots,0,2} + \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,2}\right)/D_{k+1}, \end{aligned}$$

where the denominator $D_{k+1} = \prod_{i=1}^{k+1}\left(N_i\left(z_i\right) + Z_i\left(z_i\right) + P_i\left(z_i\right)\right) = 1$. Knowing that $D_k = 1$ for $k = 1, 2, \ldots$ we have

$$\begin{aligned} S_{k+1}\left(\mathbf{z}_{k+1} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}\right) &= N_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,0}\right) \\ &+ Z_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,1}, \ldots, q_{2,\ldots,2,1}\right) \\ &+ P_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,2}, \ldots, q_{2,\ldots,2,2}\right), \end{aligned}$$

where

$$\begin{aligned} S_k\left(\mathbf{z}_k \mid q_{0,0,\ldots,0,0}, \ldots, q_{2,2,\ldots,2,2}\right) &= N_1 N_2 \ldots N_k q_{0,0,\ldots,0,0} \\ &+ Z_1 N_2 \ldots N_k q_{1,0,\ldots,0,0} \\ &+ P_1 N_2 \ldots N_k q_{2,0,\ldots,0,0} \\ &+ \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,2}\ , \end{aligned}$$

$$\begin{aligned} S_k\left(\mathbf{z}_k \mid q_{0,0,\ldots,0,1}, \ldots, q_{2,2,\ldots,2,1}\right) &= N_1 N_2 \ldots N_k q_{0,0,\ldots,0,1} \\ &+ Z_1 N_2 \ldots N_k q_{1,0,\ldots,0,1} \\ &+ P_1 N_2 \ldots N_k q_{2,0,\ldots,0,1} \\ &+ \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,1}\ , \end{aligned}$$

$$\begin{aligned} S_k\left(\mathbf{z}_k \mid q_{0,0,\ldots,0,2}, \ldots, q_{2,2,\ldots,2,2}\right) &= N_1 N_2 \ldots N_k q_{0,0,\ldots,0,2} \\ &+ Z_1 N_2 \ldots N_k q_{1,0,\ldots,0,2} \\ &+ P_1 N_2 \ldots N_k q_{2,0,\ldots,0,2} \\ &+ \ldots + P_1 P_2 \ldots P_k q_{2,2,\ldots,2,2}\ . \end{aligned}$$

Thus,

$$\begin{aligned} S_{k+1}\left(\mathbf{z}_{k+1} \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,2}\right) &= N_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,0}, \ldots, q_{2,\ldots,2,0}\right) \\ &+ Z_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,1}, \ldots, q_{2,\ldots,2,1}\right) \\ &+ P_{k+1} S_k\left(\mathbf{z}_k \mid q_{0,\ldots,0,2}, \ldots, q_{2,\ldots,2,2}\right). \end{aligned}$$

This finishes the proof of Theorem 4.15. □

The above Theorem can be used for rather large rule-bases. For $n = 3$ inputs, taking into account (4.66), it can be graphically interpreted as shown in Fig. 4.7. In the case of the TS system with n inputs, the architecture can be viewed as n-layer neural network with linear activation functions f for all neurons, where $f\left(input\right) = input$. In the layer number k, $(k = 1, \ldots, n)$, the

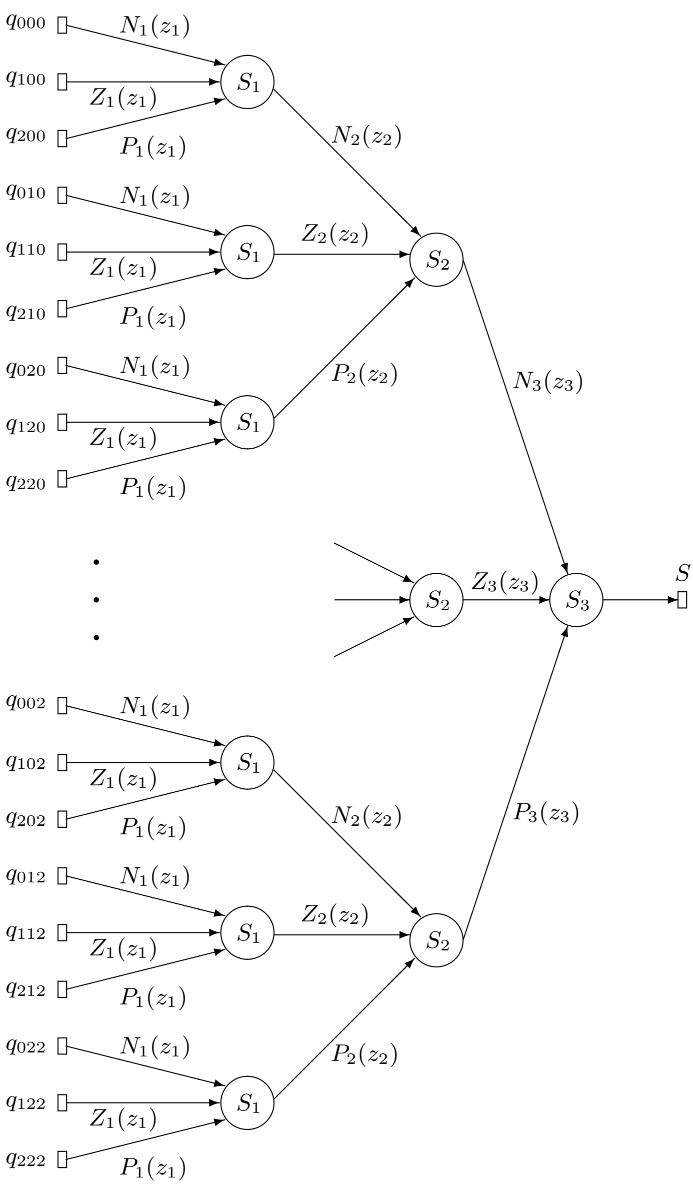


Fig. 4.7 Graphic interpretation of Theorem 4.15 for a TS system with $n = 3$ inputs and the output $S = S_3(z_1, z_2, z_3 | q_{000}, \ldots, q_{222})$

network contains exactly the same neurons S_k and every neuron has three inputs and the same weights, namely $N_k(z_k)$, $Z_k(z_k)$ and $P_k(z_k)$.

A generalization of the Theorem 4.15 for MIMO systems is straightforward and will be omitted. A computational architecture of the recursion for MIMO P2-TS systems as a generalization of (4.57) can be easily drawn, similarly to the one of Fig. 4.7, as well.

4.8 Summary

We considered the TS systems which use the second degree polynomials as the membership functions of fuzzy sets for the inputs. It was shown that it is not possible to obtain any second degree polynomial function, to which a TS rule-based system is equivalent, on the assumption that only two complementary membership functions as the second degree polynomials are defined for the input variables. However, three quadratic membership functions suffice to model every second degree polynomial function.

For the considered zero-order TS system, we defined for every input variable the set of three highly interpretable normalized membership functions as the second degree polynomials (N, Z and P). They contain one free design parameter. The TS systems that use such fuzzy sets were called P2-TS systems and they were thoroughly investigated. One of theorems says that the crisp output of the MISO P2-TS system in the vertex of the hypercuboid D^n is exactly the same as the appropriate conclusion of the fuzzy rule contained in the rule-base.

For the P2-TS systems both the generator and the fundamental matrix were defined. The fundamental matrix and its inverse are important, since they enable one to establish an exact relationship between the consequents of the "If-then" rules and the parameters that define the crisp function, to which the rule-based system is equivalent. Therefore, the procedure of how to compute the fundamental matrix and its inverse was given.

Examples 4.12-4.14 show that P2-TS systems have highly interpretable rule-bases when we use individual fuzzy rules or the metarules.

The P2-TS system with n-inputs, which normally contains a complete and noncontradictory set of fuzzy rules, consists of 3^n individual fuzzy rules. Thus, the curse of dimensionality problem is much more serious for the P2-TS systems than the one for the P1-TS systems. Therefore, we developed the recursive procedures for the computation of both the inverse of the fundamental matrix and the crisp output of the P2-TS systems. Theorem 4.10 and its generalization say that we do not need to inverse large matrices to obtain the crisp output of the P2-TS systems. As a result of these theorems, the curse of dimensionality in P2-TS systems was substantially weakened. Although we considered the MISO systems, all the results can be easily generalized for the MIMO case.

After this chapter we are able to thoroughly generalize the results for the TS systems with the membership functions that are polynomials of the degree $d \geq 3$. However, we should realize that the number of complete and noncontradictory rules will rapidly grow and the analysis will become more and more complicated. Both P1- and P2-TS systems are able to model a large class of real nonlinear processes. Therefore, if it is not necessary, we should not complicate our models in the engineering practice.

Chapter 5
Comprehensive Study and Applications of P1-TS Systems

This chapter focuses mainly on the P1-TS systems introduced in Chapter 2, as the simplest and most transparent among fuzzy rule-based systems with polynomial membership functions. They are highly interpretable and therefore they seem to be particularly important from the engineering point of view. In order to show that there are quite a lot of applications of P1-TS systems, many examples of exact modeling of conventional systems will be given, especially in relation to nonlinear dynamical processes modeling and control. We prefer to use the analytical and systematic approach to the synthesis and analysis of the models. Thanks to such approach the comparison of the methods developed in this book with others will be facilitated. Symbolic quantities will be mainly used to ensure the generality of outcomes. Seldom, if ever, will numerical data be taken, to increase transparency of the examples.

The P1-TS systems with two and more inputs will be comprehensively investigated in subsequent sections, considering interpretability issue. It will be exemplified that by using a multi-valued logic for highly nonlinear dynamical process, one can design an acceptable control algorithm expressed by the P1-TS system fuzzy rules. In this way a connection between P1-TS systems and classical combinational logic systems will be established. Next, the fuzzy rule-based systems with inputs and outputs from the unity intervals will be discussed in the context of generalized operators such as triangular norms (t-norms), t-conorms, implications, etc. The connection between fuzzy rule-based systems and Boolean algebra will become apparent. The highly interpretable rule-bases will be constructed for systems with three and more inputs not only for abstract processes, but also for real dynamical plants, e.g. a NARX model, fuzzy J-K flip-flop, Euler equations for a rigid body, Chen's attractor, the human immunodeficiency virus, magnetic suspension system, low order atmospheric circulation process and induction motor. We will exemplify that the theory of P1-TS systems can be used to transform some complicated control algorithms, formerly obtained with the use of Boolean logic, into the fuzzy domain.

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 101–182.
springerlink.com

The theory of P1-TS systems will also be used for analytical design of an optimal PID controller, working in the closed-loop control system containing a (linear and nonlinear) second order plant. Such the controller in the form of P1-TS system will be optimal with respect to typical requirements for automatic control systems. Next, we will show that using our systematic approach, the so called "controller with variable gains" introduced by Ying [205], [206] can be easily obtained.

In the last sections, exact modeling of single input dynamical systems will be investigated. Similarly as in Section 3.4 it will be assumed that nonlinear dynamical system is a collection of linear dynamical subprocesses. However, in Section 3.4 the inference was concerned with the structure parameters represented by matrices describing local linear models. In contrast to that approach, the nonlinear model of the whole system will be inferred according to the original Takagi-Sugeno inference method. Based on this inference, the class of dynamical systems to which the rule-based system is equivalent will be identified. Finally, conclusions about modeling of nonlinear dynamical systems with the use of P1-TS ones, and theorem concerning equivalence between MIMO linear dynamical systems described by state space equations and P1-TS systems will be formulated.

Because we will mainly investigate P1-TS systems, references will often be made to the outcomes of Chapter 2.

5.1 P1-TS Systems with Two Inputs

5.1.1 General Case

Consider the P1-TS fuzzy system with two inputs $z_1 \in [-\alpha_1, \beta_1]$ and $z_2 \in [-\alpha_2, \beta_2]$, and the output S (see Chapter 2). The fuzzy rules (2.36) have the consequents constituting the vector $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ and can be equivalently written in Table 2.1. The generator and the fundamental matrix are given by (2.37) and (2.38), respectively. According to Theorem 2.4 and the equation (2.45) we have

$$S(z_1, z_2) = \mathbf{g}^T(z_1, z_2) \underbrace{\left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{q}}_{\boldsymbol{\theta}} = \theta_{00} + \theta_{10} z_1 + \theta_{01} z_2 + \theta_{11} z_1 z_2, \tag{5.1}$$

where in general, the components of the vector $\boldsymbol{\theta}$ are explicitly given by

$$\theta_{00} = \frac{(q_1\beta_1 + q_2\alpha_1)\beta_2 + (q_3\beta_1 + q_4\alpha_1)\alpha_2}{V_2}, \tag{5.2}$$

$$\theta_{10} = \frac{(q_2 - q_1)\beta_2 + (q_4 - q_3)\alpha_2}{V_2}, \tag{5.3}$$

Table 5.1 Look-up-table for the P1-TS fuzzy system from Example 5.1

$z_1 \downarrow$ \ $z_2 \rightarrow$	N_2	P_2
N_1	q_1	q_1
P_1	q_2	q_2

$$\theta_{01} = \frac{(q_3 - q_1)\,\beta_1 + (q_4 - q_2)\,\alpha_1}{V_2}\,, \tag{5.4}$$

$$\theta_{11} = \frac{q_1 - q_2 - q_3 + q_4}{V_2}\,, \tag{5.5}$$

where $V_2 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2)$ is the area of the rectangle D^2. The same result we obtain using Theorem 3.6 on recursion.

Example 5.1. Assume that the rules are given in Table 5.1. They say that system output is independent of the second input. This fact can be expressed by two metarules:

$$\left.\begin{array}{l} R_1: \text{ If } z_1 \text{ is } N_1, \text{ then } S = q_1, \\ R_2: \text{ If } z_1 \text{ is } P_1, \text{ then } S = q_2. \end{array}\right\}$$

From (5.1) we obtain

$$S(z_1, z_2) = \frac{q_1\beta_1 + q_2\alpha_1}{\alpha_1 + \beta_1} + \frac{q_2 - q_1}{\alpha_1 + \beta_1} z_1,$$

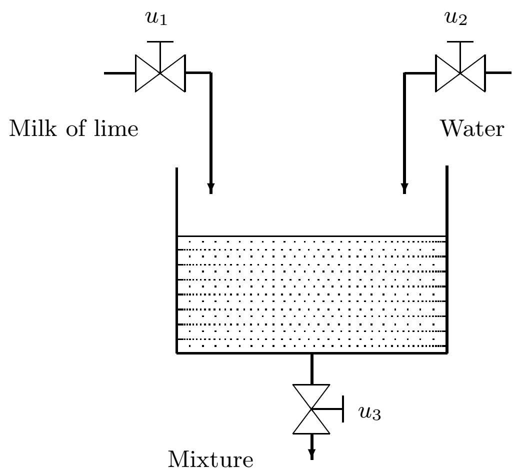
i.e. $S = S(z_1)$, indeed. The above result confirms the facts expressed by the rules.

5.1.2 A Simple Controller Design for a Milk of Lime Blending Tank

As an introduction to the next section let us consider a milk of lime blending tank (MLBT), whose objective is to produce a uniform flow of milk of lime (calcium hydroxide) [135]. The system can be described as a stirred tank (see Fig. 5.1), where a suspension of lime is mixed with water to decrease the dissolution density. The aim of the control system is to keep the density at the output at prescribed values despite the disturbances. The level of liquid in the tank is maintained within a specified range to avoid flow obstruction due to the high density of the dissolution. The mathematical model of the MLBT comprises two nonlinear state space equations corresponding to mass balances [190]:

$$\left\{\begin{array}{l} \dot{x}_1 = \eta_1 \dfrac{(1 - \alpha x_1)(p_1 - x_1)}{(1 - \alpha p_1)\, a x_2} u_1 - \eta_2 \dfrac{(1 - \alpha x_1)\, x_1}{a x_2} u_2\,, \\ \dot{x}_2 = \dfrac{1}{a}\left(\eta_1 u_1 + \eta_2 u_2 - \eta_3 \sqrt{\dfrac{x_2 - b p_2}{1 - \alpha x_1} u_3}\right), \end{array}\right. \tag{5.6}$$

Fig. 5.1 Milk of lime blending tank



where u_1, u_2, u_3 are input signals to manipulate the valves, x_1 is the concentration of the mixture at the output [1/kg], x_2 - the level of liquid in the tank [m], p_1 - the concentration of input lime suspension, p_2 - the back pressure and b is a constant $(b = 1)$. The parameters $\eta_1 = 5, \eta_2 = 3.6$ and $\eta_3 = 9.285$ are the valve constants, $a = 10$ [m^2] - the tank area and $\alpha = 0.573$ relates the concentration of the lime solution to the density of dry lime. Finally, the output variables are the specific gravity of the mixture at the output, $y_1 = 1/(1 - \alpha x_1)$ and the level of liquid, $y_2 = x_2$. Because of one-to-one mapping between x_i and y_i we assume that x_i are measurable state variables.

We assume that the control signal constraints are $u_i \in \left[u_i^L, u_i^H\right] = [0.25, 0.95]$, and the nominal positions of the valves in the steady state are $u_i = u_i^0 = 0.5$ for $i = 1, 2, 3$. Furthermore, the output valve signal is assumed to be constant $u_3(t) = u_3^0$. The steady state is given by the controls u_1^0, u_2^0 and u_3^0 as follows

$$x_1^0 = \frac{p_1}{1 + \dfrac{\eta_2 u_2^0}{\eta_1 u_1^0}(1 - \alpha p_1)} \in [-\alpha_1, \beta_1] = [0.01, 0.275281] \ , \tag{5.7}$$

$$x_2^0 = p_2 + \frac{\left(\eta_1 u_1^0 + \eta_2 u_2^0\right)^3}{\eta_3^2 \eta_2 u_2^0 u_3^0 \left(1 + \dfrac{\eta_1 u_1^0}{\eta_2 u_2^0 (1 - \alpha p_1)}\right)} \in [-\alpha_2, \beta_2] = [0.1, 0.682669] \, . \tag{5.8}$$

Our goal is to obtain two control signals $u_1(t)$ and $u_2(t)$ that stabilize the equilibrium point $\left(x_1^0, x_2^0\right) = (0.180538, 0.474573)$.

Table 5.2 Control actions for the the milk of lime blending tank

Concentration x_1	Mixture level x_2	Valve position u_1	Valve position u_2
N_1	N_2	u_1^H	u_2^H
P_1	N_2	u_1^L	u_2^H
N_1	P_2	u_1^H	u_2^L
P_1	P_2	u_1^L	u_2^L

Observe that for the above process, a very simple system of the control rules may be obtained using Boolean logic, as described in Table 5.2. It says that the valve delivering milk is opened, if the concentration x_1 is low, and it is closed if x_1 is high. Analogously, the valve delivering water is opened, if the mixture level x_2 is low, and it is closed if x_2 is high. Switching (or binary) control signal $u_j(t) \in \{u_j^L, u_j^H\}$ can be easily obtained according to the Table 5.2 as follows (see Fig. 5.2):

$$u_i(t) = \begin{cases} u_i^H & \text{iff } x_i(t) \le x_i^0 \\ u_i^L & \text{iff } x_i(t) > x_i^0, \quad i = 1, 2. \end{cases} \tag{5.9}$$

By simulations one can check that the closed-loop system by the switching control signal (5.9) is stable. However, such strategy which is based on Boolean logic has an essential drawback, since there is a huge number of on/off switchings. To avoid this phenomenon, first of all we assume that the "on/off" valves controlling the plant will be replaced by the analog servo-valves. Next, we define the control signals according to the same rules as in Table 5.2, but implementing a smooth (linear) control as shown in Fig. 5.2:

$$u_1(x_1, x_2) = u_1(x_1) = u_1^0 + k_1\left(x_1 - x_1^0\right), \tag{5.10}$$

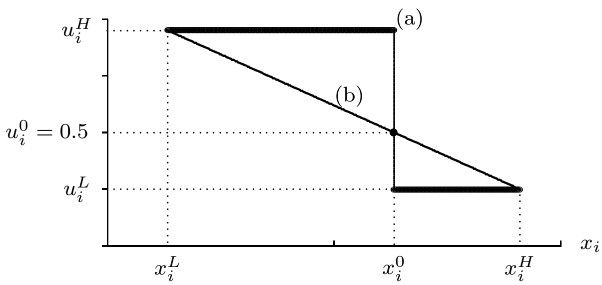


Fig. 5.2 (a) - $u_i(x_i)$ as a switching control function, (b) - $u_i(x_i)$ as a linear state feedback

$$u_2(x_1, x_2) = u_2(x_2) = u_2^0 + k_2\left(x_2 - x_2^0\right). \tag{5.11}$$

From the interpretation of the rules in Table 5.2 it follows that the gains k_1 and k_2 are negative. For simplicity we assume that $\left(u_i^H - u_i^0\right)/\left(x_i^0 - x_i^L\right) = \left(u_i^0 - u_i^L\right)/\left(x_i^H - x_i^0\right)$ and we choose $k_i = -\left(u_i^H - u_i^0\right)/\left(x_i^0 - x_i^L\right)$, for $i = 1, 2$. Thus, $k_1 = -2.59259$ and $k_2 = -1.16667$.

The fuzzy metarules for the P1-TS system performing a controller function that stabilizes the plant described by (5.6) in the equilibrium point $\left(x_1^0, x_2^0\right)$, can be simply formulated as follows

$$\left.\begin{array}{l} R_1: \text{ If } x_1 \text{ is } N_1, \text{ then } u_1 = u_1^H, \\ R_2: \text{ If } x_1 \text{ is } P_1, \text{ then } u_1 = u_1^L, \\ R_3: \text{ If } x_2 \text{ is } N_2, \text{ then } u_2 = u_2^H, \\ R_4: \text{ If } x_2 \text{ is } P_2, \text{ then } u_2 = u_2^L. \end{array}\right\}$$

where N_1 and P_1 denote "low" and "high" concentration of the mixture, whereas N_2 and P_2 - "low" and "high" level of liquid in the tank, respectively. The membership functions of fuzzy sets $N_i(x_i)$ and $P_i(x_i)$ are given by (2.11)-(2.12), assuming the intervals for x_i as in (5.7)-(5.8) for $i = 1, 2$.

The above example shows that for a highly nonlinear dynamical process, in some cases one can obtain an acceptable control algorithm expressed by the fuzzy rules for the P1-TS system, by using a multi-valued (fuzzy) logic. The method of the controller design that works according to multi-valued logic is simple and clear, especially if we analyze the fuzzy rules for the

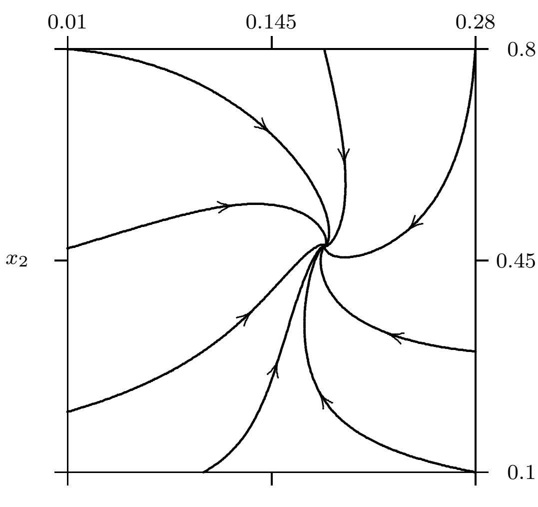


Fig. 5.3 Phase plane of the milk of lime blending tank described by (5.6), when the control signals are linear state feedback (5.10)-(5.11)

P1-TS system. The design method is almost the same as the heuristic design procedure for the combinational logic system that functions as a controller in the closed-loop. Such methods are widely known and used in practice for switching control algorithms synthesis, that are implemented in embedded hardware devices or software components e.g. for programmable logic controllers (PLCs) as the real-time direct digital control systems.

5.1.3 P1-TS Systems with Inputs and Outputs from the Unity Interval

The intervals for the input and output variables of the rule-based system can be always transformed (normalized) into the unity interval $[0,1]$, which can make the interpretation of variables very clear in many cases. For example, every control action $u_i \in \left[u_i^L, u_i^H\right]$ can be written in the form $u_i = u_i^L + \lambda\left(u_i^H - u_i^L\right)$, where $\lambda \in [0,1]$. For the controller designed in Section 5.1.2 this means that the ith valve is opened in the 100λ per cent.

The rule-based P1-TS systems in which both the inputs and outputs take the values from the interval $[0,1]$ are special class of systems. We will call them "logical systems", since

- the labels of fuzzy sets N_k are interpreted as *almost false* or *near zero*, and
- the labels of fuzzy sets P_k are interpreted as *almost true* or *near one.*

Such systems process information expressed in continuous, multi-valued logic and they are interesting from the practical point of view. Observe that logical interpretation coincides with the formerly obtained algebraic results. According to Theorem 3.15, if the conclusions q_v of "If-then" rules take the values from the unity interval $[0,1]$, then the output of the MISO P1-TS system belongs to the same interval:

$$0 \leq S(\mathbf{z}) \leq 1, \qquad \forall\, \mathbf{z} \in [0,1]^n, \tag{5.12}$$

since $\min\{q_1, q_2, \ldots, q_{2^n}\} = 0$ and $\max\{q_1, q_2, \ldots, q_{2^n}\} = 1$.

The look-up-tables, which are equivalent to "If-then" rules or metarules can be viewed as generalized Karnaugh maps. Such maps were developed for minimization of Boolean functions $f : \{0,1\}^n \to \{0,1\}$ [85], [9], [19], [115]. Karnaugh maps enable one simple and natural logical interpretation of the function f to which a given P1-TS system is equivalent - in all vertices of the unity hypercube $[0,1]^n$. A rough interpretation of f is also possible and proves useful in the points situated near these vertices, i.e. in such points of the unity hypercube which have a small entropy [110].

By convention we will assign "0" to the labels N_k and "1" to P_k. In order to convert the look-up-tables into "classical" Karnaugh maps, we should order the labels describing antecedents of rules so that they will be described by the Gray code (see e.g. Example 3.8) [158].

Assume the inputs z_1 and z_2 of the P1-TS system to be from the unity square and the consequents of the rules are allowed to be either 0, or 1. For all $(z_1, z_2) \in [0,1]^2$ let us denote the operations in *multi-valued logic* as follows:

- three basic operations:
  - *strong negation* (or *complement*)

$$\mathfrak{n}(z_1) = 1 - z_1 \ , \tag{5.13}$$

  - *t-norm* (*generalized AND*): $\mathfrak{t}(z_1, z_2)$,
  - *t-conorm* (*generalized OR*): $\mathfrak{s}(z_1, z_2)$,
- and additionally:
  - *implication* function: $\mathfrak{i}(z_1, z_2)$,
  - *equivalence* function: $\mathfrak{e}(z_1, z_2)$.

Let us denote by $S_{q_1 q_2 q_3 q_4}$ the output of the P1-TS system in which the consequents of the fuzzy rules are binary vectors

$$\mathbf{q} = [q_1, q_2, q_3, q_4]^T \in \{0,1\}^4 \, . \tag{5.14}$$

From (5.1) we obtain

$$\begin{aligned} S_{q_1 q_2 q_3 q_4} &= \mathbf{g}^T(z_1, z_2)\left(\boldsymbol{\Omega}^T\right)^{-1}\mathbf{q} \\ &= q_1(1 - z_1 - z_2 + z_1 z_2) + q_2(z_1 - z_1 z_2) + q_3(z_2 - z_1 z_2) + q_4 z_1 z_2 . \end{aligned} \tag{5.15}$$

According to (5.14) there are exactly 16 functions of two variables, which can be viewed as the multi-valued logic functions:

1. $S_{0000}(z_1, z_2) = 0$ constant zero
2. $S_{0001}(z_1, z_2) = z_1 z_2 = \mathfrak{t}(z_1, z_2)$ t-norm
3. $S_{0010}(z_1, z_2) = z_2 - z_1 z_2 = \mathfrak{n}(\mathfrak{i}(z_2, z_1))$ negation of implication
4. $S_{0011}(z_1, z_2) = z_2$ variable z_2
5. $S_{0100}(z_1, z_2) = z_1 - z_1 z_2 = \mathfrak{n}(\mathfrak{i}(z_1, z_2))$ negation of implication
6. $S_{0101}(z_1, z_2) = z_1$ variable z_1
7. $S_{0110}(z_1, z_2) = z_1 + z_2 - 2z_1 z_2 = \mathfrak{n}(\mathfrak{e}(z_1, z_2))$ negation of equivalence
8. $S_{0111}(z_1, z_2) = z_1 + z_2 - z_1 z_2 = \mathfrak{s}(z_1, z_2)$ t-conorm
9. $S_{1000}(z_1, z_2) = 1 - z_1 - z_2 + z_1 z_2 = \mathfrak{n}(\mathfrak{s}(z_1, z_2))$ negation of t-conorm
10. $S_{1001}(z_1, z_2) = 1 - z_1 - z_2 + 2z_1 z_2 = \mathfrak{e}(z_1, z_2)$ equivalence
11. $S_{1010}(z_1, z_2) = 1 - z_1 = \mathfrak{n}(z_1)$ negation of variable z_1
12. $S_{1011}(z_1, z_2) = 1 - z_1 + z_1 z_2 = \mathfrak{i}(z_1, z_2)$ implication
13. $S_{1100}(z_1, z_2) = 1 - z_2 = \mathfrak{n}(z_2)$ negation of z_2
14. $S_{1101}(z_1, z_2) = 1 - z_2 + z_1 z_2 = \mathfrak{i}(z_2, z_1)$ implication
15. $S_{1110}(z_1, z_2) = 1 - z_1 z_2 = \mathfrak{n}(\mathfrak{t}(z_1, z_2))$ negation of t-norm
16. $S_{1111}(z_1, z_2) = 1$ constant one

Observe that the function

$$\mathfrak{t}(z_1, z_2) = z_1 z_2 \tag{5.16}$$

is a special case of the t-norm, called *probabilistic t-norm*,

$$\mathfrak{s}(z_1, z_2) = z_1 + z_2 - z_1 z_2 \tag{5.17}$$

is a special case of the t-conorm, called *probabilistic t-conorm*, and

$$\mathfrak{i}(z_1, z_2) = 1 - z_2 + z_1 z_2 \tag{5.18}$$

is a special case of the implication function, known as *Reichenbach's implication* [38].

Based on strong negation, t-norm and t-conorm one can easily define a *fuzzy algebra*. However, we omit this problem and make some remarks instead. It is well known that e.g. *de Morgan's laws* are satisfied in the fuzzy algebra. One can check that for the above basic operations (5.13), (5.16) and (5.17), the following equations

$$\mathfrak{n}(\mathfrak{t}(\mathfrak{n}(z_1), \mathfrak{n}(z_2))) = z_1 + z_2 - z_1 z_2 = \mathfrak{s}(z_1, z_2),$$
$$\mathfrak{n}(\mathfrak{s}(\mathfrak{n}(z_1), \mathfrak{n}(z_2))) = z_1 z_2 = \mathfrak{t}(z_1, z_2),$$

are satisfied for every point (z_1, z_2) from the square $[0,1]^2$. This means that $(\mathfrak{t}, \mathfrak{s})$ is $\mathfrak{n}$-*dual pair of operators* [38]. Not all features hold, that are known from Boolean algebra. For example

$$\mathfrak{t}(z, \mathfrak{n}(z)) = z(1-z) > 0 \qquad \text{for} \quad z \in (0,1), \tag{5.19}$$
$$\mathfrak{s}(z, \mathfrak{n}(z)) = 1 - z(1-z) < 1 \qquad \text{for} \quad z \in (0,1). \tag{5.20}$$

This means that the Aristotelean *noncontradiction law* and *excluded middle law* do not hold. Therefore, for example the equivalence function in Boolean logic satisfies

$$\mathfrak{e}(z_1, z_2) = \mathfrak{t}(\mathfrak{i}(z_1, z_2), \mathfrak{i}(z_2, z_1)), \qquad \text{for} \quad z \in \{0,1\},$$

but in the case of our operators

$$\mathfrak{e}(z_1, z_2) = \mathfrak{t}(\mathfrak{i}(z_1, z_2), \mathfrak{i}(z_2, z_1)) - \mathfrak{t}(\mathfrak{t}(z_1, \mathfrak{n}(z_1)), \mathfrak{t}(z_2, \mathfrak{n}(z_2))),$$

for $z_1, z_2 \in [0,1]$. The additional term $\mathfrak{t}(\mathfrak{t}(z_1, \mathfrak{n}(z_1)), \mathfrak{t}(z_2, \mathfrak{n}(z_2))) \in [0, 1/16]$; it is zero in the vertices of the square $[0,1]^2$ only, and takes the value $1/16$ for $z_1 = z_2 = 1/2$. Similarly, the absorption law in Boolean logic

$$\mathfrak{s}(z_1, \mathfrak{t}(z_1, z_2)) = z_1, \qquad \forall z_1, z_2 \in \{0,1\},$$

holds, but in our system

$$\mathfrak{s}\left(z_1, \mathfrak{t}\left(z_1, z_2\right)\right) = z_1 + \mathfrak{t}\left(\mathfrak{t}\left(z_1, \mathfrak{n}\left(z_1\right)\right), z_2\right), \qquad \forall z_1, z_2 \in [0,1].$$

Obviously, all above operations are exactly the same as well-known classical binary ones, in the vertices of the square $[0,1]^2$.

We discussed the P1-TS logical systems with only two variables in order to have a connection with Boolean algebra. The same systems with more than two inputs will be applied for modeling real systems further on.

5.2 P1-TS Fuzzy Systems with Three Inputs

5.2.1 General Case

Consider the P1-TS fuzzy system with three inputs $z_k \in [-\alpha_k, \beta_k]$ for $k = 1,2,3$ and the output S (see Chapter 2). The fuzzy rules are given in Table 5.3. The generator and the fundamental matrix are given by (2.40) and (2.41), respectively. According to Theorem 2.4 and equation (2.45) we have

$$\begin{aligned} S\left(z_1, z_2, z_3\right) &= \mathbf{g}^T\left(z_1, z_2, z_3\right) \underbrace{\left(\boldsymbol{\Omega}^T\right)^{-1}\left[q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\right]^T}_{\boldsymbol{\theta}} \\ &= \sum_{(i,j,k)\in\{0,1\}^3} \theta_{ijk} z_1^i z_2^j z_3^k \,, \end{aligned} \tag{5.21}$$

where the coefficients θ_{ijk} of the vector $\boldsymbol{\theta}$ that correspond to the generator $\mathbf{g}\left(z_1, z_2, z_3\right)$ are given explicitly as follows

$$\begin{aligned} \theta_{000} = &\frac{\left(q_1\beta_1 + q_2\alpha_1\right)\beta_2 + \left(q_3\beta_1 + q_4\alpha_1\right)\alpha_2}{V_3}\beta_3 \\ &+ \frac{\left(q_5\beta_1 + q_6\alpha_1\right)\beta_2 + \left(q_7\beta_1 + q_8\alpha_1\right)\alpha_2}{V_3}\alpha_3 \,, \end{aligned} \tag{5.22}$$

$$\theta_{100} = \frac{q_2\beta_2 - q_1\beta_2 - q_3\alpha_2 + q_4\alpha_2}{V_3}\beta_3 + \frac{q_6\beta_2 - q_5\beta_2 - q_7\alpha_2 + q_8\alpha_2}{V_3}\alpha_3 \,, \tag{5.23}$$

$$\theta_{010} = \frac{q_3\beta_1 - q_2\alpha_1 - q_1\beta_1 + q_4\alpha_1}{V_3}\beta_3 + \frac{q_7\beta_1 - q_6\alpha_1 - q_5\beta_1 + q_8\alpha_1}{V_3}\alpha_3 \,, \tag{5.24}$$

Table 5.3 Look-up-table for the P1-TS fuzzy system with $n = 3$ inputs - a general case

$z_1, z_2 \setminus z_3 \rightarrow$ $\downarrow$	N_3	P_3
N_1N_2	q_1	q_5
N_1P_2	q_3	q_7
P_1P_2	q_4	q_8
P_1N_2	q_2	q_6

$$\theta_{110} = \frac{q_1 - q_2 - q_3 + q_4}{V_3}\beta_3 + \frac{q_5 - q_6 - q_7 + q_8}{V_3}\alpha_3 \ , \tag{5.25}$$

$$\theta_{001} = \frac{(q_5 - q_1)\,\beta_1 + (q_6 - q_2)\,\alpha_1}{V_3}\beta_2 + \frac{(q_7 - q_3)\,\beta_1 + (q_8 - q_4)\,\alpha_1}{V_3}\alpha_2 \ , \tag{5.26}$$

$$\theta_{101} = \frac{(q_1 - q_2 - q_5 + q_6)\,\beta_2 + (q_3 - q_4 - q_7 + q_8)\,\alpha_2}{V_3} \ , \tag{5.27}$$

$$\theta_{011} = \frac{(q_1 - q_3 - q_5 + q_7)\,\beta_1 + (q_2 - q_4 - q_6 + q_8)\,\alpha_1}{V_3} \ , \tag{5.28}$$

$$\theta_{111} = \frac{-q_1 + q_2 + q_3 - q_4 + q_5 - q_6 - q_7 + q_8}{V_3} \ , \tag{5.29}$$

and $V_3 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2)(\alpha_3 + \beta_3)$ is the volume of the cuboid D^3.

5.2.2 Examples of Highly Interpretable P1-TS Systems with Three Inputs

Several examples will be given further on, but we begin with a simple theoretical one. As usual, the symbols will be used to preserve the generality of the results.

Example 5.2. Consider the following complete and noncontradictory system of fuzzy metarules:

$$\left.\begin{aligned} &M_1 : \text{If } z_1 \text{ is } N_1, \text{ then } S = p, \\ &M_2 : \text{If } z_1 \text{ is } P_1 \text{ and } z_3 \text{ is } N_3, \text{ then } S = q, \\ &M_3 : \text{If } z_1 \text{ is } P_1 \text{ and } z_3 \text{ is } P_3, \text{ then } S = r. \end{aligned}\right\} \tag{5.30}$$

Equivalently, it corresponds to eight fuzzy rules given in Table 5.4. According to the rules, the system output S does not depend on the second input variable. This is especially clear, if we view the look-up-table as (generalized)

Table 5.4 a) Look-up-table of the TS fuzzy system from Example 5.2, b) Description of the same table using binary Gray code

a)

$z_1, z_2 \setminus z_3 \rightarrow$ ↓	N_3	P_3
N_1N_2	p	p
N_1P_2	p	p
P_1P_2	q	r
P_1N_2	q	r

b)

$z_1, z_2 \setminus z_3 \rightarrow$ ↓	0	1
00	p	p
01	p	p
11	q	r
10	q	r

Karnaugh map as shown in Table 5.4 b. Conclusions of the rules are: $q_1 = p$, $q_2 = q$, $q_3 = p$, $q_4 = q$, $q_5 = p$, $q_6 = r$, $q_7 = p$ and $q_8 = r$.

Our goal is to check whether the logical interpretation of the rules coincides with the crisp system output. From (5.22)-(5.29) we obtain

$$S(z_1, z_2, z_3) = \theta_{000} + \theta_{100} z_1 + \theta_{001} z_3 + \theta_{101} z_1 z_3,$$

where

$$\begin{aligned}
\theta_{000} &= (\alpha_1 + \beta_1)^{-1} (\alpha_3 + \beta_3)^{-1} (p\beta_1\alpha_3 + p\beta_1\beta_3 + q\alpha_1\beta_3 + r\alpha_1\alpha_3), \\
\theta_{100} &= (\alpha_1 + \beta_1)^{-1} (\alpha_3 + \beta_3)^{-1} (q\beta_3 - p\beta_3 - p\alpha_3 + r\alpha_3), \\
\theta_{010} &= 0, \\
\theta_{110} &= 0, \\
\theta_{001} &= (\alpha_1 + \beta_1)^{-1} (\alpha_3 + \beta_3)^{-1} (r - q)\,\alpha_1, \\
\theta_{101} &= (\alpha_3 + \beta_3)^{-1} (\alpha_1 + \beta_1)^{-1} (r - q), \\
\theta_{011} &= 0, \\
\theta_{111} &= 0.
\end{aligned}$$

This means that independently of all constants: p, q, r, α_1, α_2, α_3, β_1, β_2, and β_3, the output of the considered P1-TS system does not depend on the input z_2, indeed. The result agrees with our expectation.

Example 5.3. Suppose we need to obtain the rules for the P1-TS system with the inputs $[z_1, z_2, z_3]^T \in D^3$, which is equivalent to the following multivariate polynomial

$$f_0(z_1, z_2, z_3) = z_1 z_2 (1 - z_3), \qquad z_k \in [-\alpha_k, \beta_k], \quad k = 1, 2, 3. \tag{5.31}$$

The function (5.31) is a special case of (2.26) with $\boldsymbol{\theta} = [0, 0, 0, 1, 0, 0, 0, -1]^T$. Taking into account $\boldsymbol{\Omega}$ given by (2.41) and the equation (2.44), for the given $\boldsymbol{\theta}$ we obtain

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \boldsymbol{\Omega}^T \boldsymbol{\theta} = \begin{bmatrix} \alpha_2\alpha_1(\alpha_3 + 1) \\ -\beta_1\alpha_2(\alpha_3 + 1) \\ -\beta_2\alpha_1(\alpha_3 + 1) \\ \beta_2\beta_1(\alpha_3 + 1) \\ \alpha_2\alpha_1(1 - \beta_3) \\ \beta_1\alpha_2(\beta_3 - 1) \\ \beta_2\alpha_1(\beta_3 - 1) \\ \beta_2\beta_1(1 - \beta_3) \end{bmatrix}. \tag{5.32}$$

The system of 8 fuzzy rules is the same as in (2.39) or Table 5.3 with the consequents of the rules (5.32). For example if $[z_1, z_2, z_3]^T \in [1, 2] \times [3, 4] \times [0, 1]$, then knowing the function (5.31) we can immediately write the system, which consists of four individual rules and one metarule:

R_1 : If z_1 is about 1 and z_2 is about 3 and z_3 is about 0, then $S = 3$,
R_2 : If z_1 is about 2 and z_2 is about 3 and z_3 is about 0, then $S = 6$,
R_3 : If z_1 is about 1 and z_2 is about 4 and z_3 is about 0, then $S = 4$,
R_4 : If z_1 is about 2 and z_2 is about 4 and z_3 is about 0, then $S = 8$
M : If z_3 is about 1, then $S = 0$.

The above rules result from (5.32) by $\alpha_1 = -1$, $\beta_1 = 2$, $\alpha_2 = -3$, $\beta_2 = 4$, $\alpha_3 = 0$ and $\beta_3 = 1$. Observe that they all have a clear interpretation for any location of the cuboid D^3 in the space $\mathbb{R}^3$.

Example 5.4. Consider the discrete-time NARX model (Nonlinear AutoRegressive with the eXtra input) considered in [208]

$$\begin{aligned} y(k+1) &= c_0 + c_1 y(k) + c_2 y(k-1) + c_3 u(k) + c_4 y(k) y(k-1) \\ &\quad + c_5 y(k) u(k) + c_6 y(k-1) u(k) + c_7 y(k) y(k-1) u(k), \end{aligned} \tag{5.33}$$

in which $y(k)$, $y(k-1) \in [-L_1, L_1]$ and $u(k) \in [-L_2, L_2]$. Our goal is to show that this system can be easily modeled using P1-TS system.

The inputs and the output of the P1-TS system which should exactly model the difference equation (5.33) are shown in Fig. 5.4. In order to make a simple comparison with the result of [208] we define the vector $\boldsymbol{\theta} = \mathbf{c} = [c_0, c_1, c_2, c_4, c_3, c_5, c_6, c_7]^T$. The fuzzy rules are given by Table 5.5. The result in [208] is as follows

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \lambda_1 & \lambda_1 & \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 & -\lambda_1 & -\lambda_1 \\ \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 & \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 \\ \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 \\ \lambda_3 & \lambda_3 & -\lambda_3 & -\lambda_3 & -\lambda_3 & -\lambda_3 & \lambda_3 & \lambda_3 \\ \lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & \lambda_4 \\ \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 \\ \lambda_5 & -\lambda_5 & -\lambda_5 & \lambda_5 & -\lambda_5 & \lambda_5 & \lambda_5 & -\lambda_5 \end{bmatrix} \begin{bmatrix} V_1' \\ V_2' \\ V_3' \\ V_4' \\ V_5' \\ V_6' \\ V_7' \\ V_8' \end{bmatrix}.$$

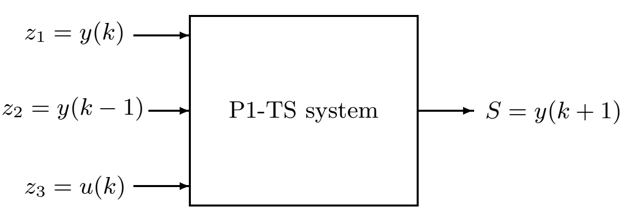


Fig. 5.4 NARX model from [208] (p. 112-114) considered in Example 5.4

Table 5.5 Look-up-table for the TS fuzzy system from Example 5.4.

$z_1, z_2 \setminus z_3 \rightarrow$ $\downarrow$	N_3	P_3
N_1N_2	V_8'	V_7'
N_1P_2	V_6'	V_5'
P_1P_2	V_2'	V_1'
P_1N_2	V_4'	V_3'

On the other hand, according to Tables 5.3 and 5.5 the vector of conclusions of the rules is $\mathbf{q} = [V_8', V_4', V_6', V_2', V_7', V_3', V_5', V_1']^T$. For the system generator (2.40) and according to (5.33) we have

$$S = \mathbf{c}^T \mathbf{g}(\mathbf{z}) = [c_0, c_1, c_2, c_4, c_3, c_5, c_6, c_7]\,[1, z_1, z_2, z_1z_2, z_3, z_1z_3, z_2z_3, z_1z_2z_3]^T.$$

From (2.41) by $\alpha_k = \beta_k = L_1$ for $k = 1, 2$ and $\alpha_3 = \beta_3 = L_2$ we obtain the following fundamental matrix

$$\boldsymbol{\Omega} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -L_1 & L_1 & -L_1 & L_1 & -L_1 & L_1 & -L_1 & L_1 \\ -L_1 & -L_1 & L_1 & L_1 & -L_1 & -L_1 & L_1 & L_1 \\ L_1^2 & -L_1^2 & -L_1^2 & L_1^2 & L_1^2 & -L_1^2 & -L_1^2 & L_1^2 \\ -L_2 & -L_2 & -L_2 & -L_2 & L_2 & L_2 & L_2 & L_2 \\ L_1L_2 & -L_1L_2 & L_1L_2 & -L_1L_2 & -L_1L_2 & L_1L_2 & -L_1L_2 & L_1L_2 \\ L_1L_2 & L_1L_2 & -L_1L_2 & -L_1L_2 & -L_1L_2 & -L_1L_2 & L_1L_2 & L_1L_2 \\ -L_1^2L_2 & L_1^2L_2 & L_1^2L_2 & -L_1^2L_2 & L_1^2L_2 & -L_1^2L_2 & -L_1^2L_2 & L_1^2L_2 \end{bmatrix}. \tag{5.34}$$

Let us take the notations

$$\lambda_1 = \frac{1}{8L_1}, \quad \lambda_2 = \frac{1}{8L_2}, \quad \lambda_3 = \frac{1}{8L_1^2}, \quad \lambda_4 = \frac{1}{8L_1L_2}, \quad \lambda_5 = \frac{1}{8L_1^2L_2}. \tag{5.35}$$

Now we obtain the coefficients of the vector $\boldsymbol{\theta}$ for the consequents of the rules given in Table 5.5. According to (2.44) the vector of coefficients of the system output is $S = \boldsymbol{\theta}^T \mathbf{g}(\mathbf{z})$, where

$$\boldsymbol{\theta}^T = \mathbf{q}^T \boldsymbol{\Omega}^{-1} = [\theta_0, \theta_1, \theta_2, \theta_4, \theta_3, \theta_5, \theta_6, \theta_7] = [V_8', V_4', V_6', V_2', V_7', V_3', V_5', V_1']\,\boldsymbol{\Omega}^{-1},$$

where $\boldsymbol{\Omega}$ is given by (5.34). Equivalently by using notation (5.35) we obtain

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_4 \\ \theta_3 \\ \theta_5 \\ \theta_6 \\ \theta_7 \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ -\lambda_1 & \lambda_1 & -\lambda_1 & \lambda_1 & -\lambda_1 & \lambda_1 & -\lambda_1 & \lambda_1 \\ -\lambda_1 & -\lambda_1 & \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 & \lambda_1 & \lambda_1 \\ \lambda_3 & -\lambda_3 & -\lambda_3 & \lambda_3 & \lambda_3 & -\lambda_3 & -\lambda_3 & \lambda_3 \\ -\lambda_2 & -\lambda_2 & -\lambda_2 & -\lambda_2 & \lambda_2 & \lambda_2 & \lambda_2 & \lambda_2 \\ \lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & \lambda_4 \\ \lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & \lambda_4 \\ -\lambda_5 & \lambda_5 & \lambda_5 & -\lambda_5 & \lambda_5 & -\lambda_5 & -\lambda_5 & \lambda_5 \end{bmatrix} \begin{bmatrix} V_8' \\ V_4' \\ V_6' \\ V_2' \\ V_7' \\ V_3' \\ V_5' \\ V_1' \end{bmatrix} .$$

This means that the results obtained using the theory of P1-TS systems and in [208] are the same. Next observe that the NARX model becomes a linear ARX one without offset if, and only if, the conclusions of the rules are as follows

$$\mathbf{q} = \begin{bmatrix} V_8' \\ V_4' \\ V_6' \\ V_2' \\ V_7' \\ V_3' \\ V_5' \\ V_1' \end{bmatrix} = \mathbf{\Omega}^T \begin{bmatrix} 0 \\ c_1 \\ c_2 \\ 0 \\ c_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -L_1c_1 - L_1c_2 - L_2c_3 \\ L_1c_1 - L_1c_2 - L_2c_3 \\ -L_1c_1 + L_1c_2 - L_2c_3 \\ L_1c_1 + L_1c_2 - L_2c_3 \\ -L_1c_1 - L_1c_2 + L_2c_3 \\ L_1c_1 - L_1c_2 + L_2c_3 \\ -L_1c_1 + L_1c_2 + L_2c_3 \\ L_1c_1 + L_1c_2 + L_2c_3 \end{bmatrix} ,$$

and the above result agrees with the one obtained in [208], as well. The only difference follows from the notation and sequence of the rules, but this is of no importance, as was shown in Section 2.7. It should be stressed that our approach that uses matrices is a very simple and systematic one.

Recently, a great deal of research and development has been directed toward designing and implementation of the "fuzzy computer" components [7], [26], [84], [120]. There are various types of fuzzy logic circuits. The most simple are combinational logic systems based on fuzzy gates. More complicated are fuzzy sequential circuits, to which belong various types of fuzzy flip-flops, that are useful in many applications, e.g. in hardware implementation of fuzzy Petri nets [60], [55], [56], [102], [104], [105], [106].

Example 5.5. Let us consider a fuzzy J-K flip-flop which was originally developed in [60]. It is known that for its synthesis one can use various t-norms and s-norms. All proposed fuzzy flip-flops are the generalized form of the ordinary (binary) J-K flip-flop [9], with the symbol shown in Fig. 5.5 a. Their fuzzy truth-tables are different but they include the binary truth-table of the

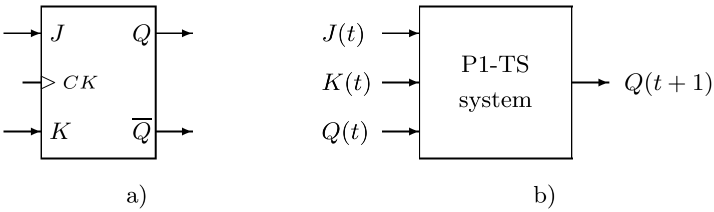


Fig. 5.5 a) Symbol of a J-K flip-flop. b) P1-TS system as a fuzzy J-K flip-flop which works according to the discrete-time state equation (5.37).

Table 5.6 Truth table of the conventional JK flip-flop

$J(t)$	$K(t)$	$Q(t)$	$Q(t+1)$
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	0

conventional J-K flip-flop. A conventional J-K flip-flop is a system operating in the domain of two-valued logic. It is used to memorize a single bit of information and can be unambiguously described as shown in Table 5.6. Assume that both the flip-flop inputs, and its present state constitute three inputs for the P1-TS system, whereas the next flip-flop state $Q(t+1)$ is the P1-TS system output

$$z_1 = J(t), \quad z_2 = K(t), \quad z_3 = Q(t), \quad S = Q(t+1).$$

We allow the signals to take not only values from the bivalent set $\{0,1\}$, but from the whole interval $[0,1]$. Thus, $[-\alpha_i, \beta_i] = [0,1]$ for $i = 1,2,3$ and therefore $Q(t+1) \in [0,1]$. According to Table 5.6 we formulate the following system of fuzzy rules in the matrix form

$$\text{If} \quad [z_1, z_2, z_3] \quad \text{is} \quad \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ P_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix}, \quad \text{then} \quad S \quad \text{is} \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5.36}$$

The vector containing the consequents of the rules is the same as the last column in Table 5.6. According to Theorem 2.10 and taking a generator $\mathbf{g}(z_1, z_2, z_3)$ from (2.40), we have

$$Q(t+1) = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \\ z_3 \\ z_1 z_3 \\ z_2 z_3 \\ z_1 z_2 z_3 \end{bmatrix}^T \left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \right)^{-1} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Thus,

$$\begin{aligned} Q(t+1) &= z_1 + z_3 - z_1 z_3 - z_2 z_3 \\ &= J(t)(1 - Q(t)) + (1 - K(t))\, Q(t) \\ &= J(t)\,\overline{Q(t)} + \overline{K(t)}Q(t), \qquad t = 0, 1, 2, \ldots \end{aligned} \tag{5.37}$$

In [60] the same result was obtained in an entirely different way, where (5.37) was called *the fundamental equation of an algebraic fuzzy flip-flop*. In the cited work, based on this equation, both discrete-mode and continuous-mode circuits have been presented.

Example 5.6. Consider the rule-based system with the inputs $\omega_1(t)$, $\omega_2(t)$ and $\omega_3(t)$ which are components of the angular velocity vector $\boldsymbol{\omega}$ along the principal axes and $\boldsymbol{\omega}$ is a point of the cube $D^3 = [-\omega_{\max}, \omega_{\max}]^3$, where $\omega_{\max}$ is a maximal angular velocity. Every input ω_k has assigned two linear membership functions of fuzzy sets $N_k(\omega_k)$ and $P_k(\omega_k) = 1 - N_k(\omega_k)$ for $k = 1, 2, 3$. Thus, N_k is interpreted as *negative* and P_k as *positive* angular velocity ω_k about kth axis. The outputs of the TS system are torques that measure the tendency of a force to rotate the object about particular axes $S_1 = J_1\dot{\omega}_1(t)$, $S_2 = J_2\dot{\omega}_2(t)$ and $S_3 = J_3\dot{\omega}_3(t)$, where J_1, J_2 and J_3 are the principal moments of inertia. Suppose that the system consists of the following 8 fuzzy rules

$$\text{If } [\omega_1, \omega_2, \omega_3] \text{ is } \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ P_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix}, \text{ then } [S_1, S_2, S_3] \text{ is } \begin{bmatrix} A_1 & B_1 & C_1 \\ A_1 & B_2 & C_2 \\ A_2 & B_1 & C_2 \\ A_2 & B_2 & C_1 \\ A_2 & B_2 & C_1 \\ A_2 & B_1 & C_2 \\ A_1 & B_2 & C_2 \\ A_1 & B_1 & C_1 \end{bmatrix}, \tag{5.38}$$

where

$$A_1 = \omega_{\max}^2 (J_2 - J_3) + u_1, \qquad A_2 = -\omega_{\max}^2 (J_2 - J_3) + u_1, \tag{5.39}$$

$$B_1 = \omega_{\max}^2 (J_3 - J_1) + u_2, \qquad B_2 = -\omega_{\max}^2 (J_3 - J_1) + u_2, \tag{5.40}$$

$$C_1 = \omega_{\max}^2 (J_1 - J_2) + u_3, \qquad C_2 = -\omega_{\max}^2 (J_1 - J_2) + u_3, \tag{5.41}$$

and $u_k = u_k(t)$ are the relative torque inputs applied about the principal axes, which are assumed to be crisp signals. Therefore they are considered to be parameters but not the system inputs. The rules are clear and can be written in the form of metarules. For example, two of them say that the torque $S_1 = A_1$ if the remaining angular speeds ω_2 and ω_3 are at the same time either *negative* or *positive*. The torque $S_1 = A_2$ if the remaining angular speeds ω_2 and ω_3 are in opposition to each other, i.e. the pair (ω_2, ω_3) is either N_2P_3 or P_2N_3. The conclusions A_1 and A_2 do not depend on the angular velocity ω_1. We can analyze all the rules in a similar way. The remaining facts concerning the torques S_k can be explained according to the fuzzy rules.

The problem is "What does an exact conventional model of this system look like?" In order to give an answer to this question we compute the vector of system outputs

$$\mathbf{S} = \mathbf{g}^T (\omega_1(t), \omega_2(t), \omega_3(t)) \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{Q}$$

for $n = 3$, by the matrix $\mathbf{Q}$ of the consequents of the rules given in (5.38). The generator $\mathbf{g}$ for the input variables ω_1, ω_2 and ω_3 can be obtained from (2.40) and the inverse of the fundamental matrix for $\alpha_k = \beta_k = \omega_{\max}$, $(k = 1, 2, 3)$, is given by (B.11) in Appendix B. We immediately obtain the following equation

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} (J_2 - J_3)\,\omega_2\omega_3 + u_1 \\ (J_3 - J_1)\,\omega_1\omega_3 + u_2 \\ (J_1 - J_2)\,\omega_1\omega_2 + u_3 \end{bmatrix}.$$

This means that the considered rule-based P1-TS system is equivalent to

$$\begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{5.42}$$

In other words the fuzzy rules of the P1-TS system in (5.38) are equivalent to the Euler equations for a rigid body, e.g. a rotating rigid spacecraft [89], [111], [153].

Example 5.7. Consider the chaotic system investigated in [121], [220], developed by Chen [23]. In the original form the system is described by the following three nonlinear differential equations

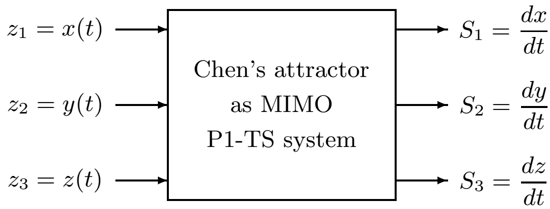


Fig. 5.6 The inputs and the outputs of the MIMO P1-TS system modeling the Chen's attractor from Example 5.7

$$\begin{cases} \dot{x} = a\,(y - x)\,, \\ \dot{y} = (c - a)\,x - xz + cy, \\ \dot{z} = xy - bz, \end{cases} \tag{5.43}$$

where the constants are $a = 35$, $b = 3$, and $c = 28$. We will show that this system can be exactly described by the zero-order P1-TS system in which the inputs are $z_1 = x\,(t) \in [-\alpha_1, \beta_1]$, $z_2 = y\,(t) \in [-\alpha_2, \beta_2]$, and $z_3 = z\,(t) \in [-\alpha_3, \beta_3]$, and the outputs are $S_1 = \dot{x}\,(t)$, $S_2 = \dot{y}\,(t)$ and $S_3 = \dot{z}\,(t)$, as shown in Fig. 5.6.

For system generator (2.40) the equations (5.43) can be equivalently written as

$$\begin{bmatrix} \dot{x}, \, \dot{y}, \, \dot{z} \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ y \\ xy \\ z \\ xz \\ yz \\ xyz \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ -a & c-a & 0 \\ a & c & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -b \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{g}^T\,(x, y, z)\,\mathbf{\Theta}. \tag{5.44}$$

According to the formula (2.52) for the MIMO P1-TS systems, the fundamental matrix (2.41) and (5.44) we obtain

$$\mathbf{Q} = \mathbf{\Omega}^T \mathbf{\Theta} = \begin{bmatrix} a\,(\alpha_1 - \alpha_2) & \alpha_1\,(a - c - \alpha_3) - c\alpha_2 & b\alpha_3 + \alpha_1\alpha_2 \\ -a\,(\alpha_2 + \beta_1) & \beta_1\,(c - a + \alpha_3) - c\alpha_2 & b\alpha_3 - \alpha_2\beta_1 \\ a\,(\alpha_1 + \beta_2) & \alpha_1\,(a - c - \alpha_3) + c\beta_2 & b\alpha_3 - \alpha_1\beta_2 \\ a\,(\beta_2 - \beta_1) & \beta_1\,(c - a + \alpha_3) + c\beta_2 & b\alpha_3 + \beta_1\beta_2 \\ a\,(\alpha_1 - \alpha_2) & \alpha_1\,(a - c + \beta_3) - c\alpha_2 & \alpha_1\alpha_2 - b\beta_3 \\ -a\,(\alpha_2 + \beta_1) & \beta_1\,(c - a - \beta_3) - c\alpha_2 & -\,(b\beta_3 + \alpha_2\beta_1) \\ a\,(\alpha_1 + \beta_2) & \alpha_1\,(a - c + \beta_3) + c\beta_2 & -\,(b\beta_3 + \alpha_1\beta_2) \\ a\,(\beta_2 - \beta_1) & \beta_1\,(c - a - \beta_3) + c\beta_2 & \beta_1\beta_2 - b\beta_3 \end{bmatrix}. \tag{5.45}$$

The column vectors of this matrix are the conclusions of the fuzzy rules, assigned to the outputs S_1, S_2 and S_3, respectively. Thus, for all points (x, y, z) of the cuboid D^3 the considered chaotic system (5.43) is equivalent to the P1-TS system defined by 8 fuzzy rules, or equivalently - by one rule in the matrix form

$$\text{If} \quad [x, y, z] \quad \text{is} \quad \mathbf{M}, \quad \text{then} \quad [\dot{x}, \dot{y}, \dot{z}] \quad \text{is} \quad \mathbf{Q}, \tag{5.46}$$

where the antecedents matrix is the same as in (5.38) and the consequents matrix is given by (5.45). All consequents of the fuzzy rules are real numbers depending on the constants a, b, c and constraints for the variables x, y and z.

Example 5.8. Several models of virus dynamics can be found in literature [198], [29]. Let us consider the human immunodeficiency virus (HIV) infection and the elementary modeling of the immune system when it is subject to HIV infection. First we show the derivation of one of the simplest traditional models concerning the viral infection dynamics, next we obtain the same model in the form of P1-TS system.

A) *Conventional model in the form of differential equations* [198], [29].
The immune system is mainly based on the so-called CD4 cells and the CD8 cells. The CD4 cells act as markers, they mark and identify the undesirable agents as viruses, bacteria, etc. The CD8 cells act as killers. However the CD8 cells kill only agents that have been marked beforehand by some CD4 cell. A virus attacks the basis of the immune system by infecting CD4 cells. Infected CD4 cells act as host cells and they produce new virions. An elementary model may be derived as follows. Let us denote by $T(t)$, $I(t)$ and $V(t)$ the population of healthy cells (hepatocytes susceptible to infection), infected cells, and free viruses, respectively. Assume that new CD4 cells $T(t)$ are produced at a constant rate λ, die at per capita rate d, and become infected at a rate proportional (with constant k) to both the virus concentration $V(t)$ and cell concentration $T(t)$:

$$\dot{T}(t) = \lambda - dT(t) - kV(t)T(t). \tag{5.47}$$

Infected liver cells (hepatocytes) are assumed to die at constant rate δ per cell and are produced at the above mentioned rate $kV(t)T(t)$:

$$\dot{I}(t) = -\delta I(t) + kV(t)T(t). \tag{5.48}$$

Upon infection, hepatocytes produce the virus at rate p per infected cell, and the virus is cleared at rate c per virion:

$$\dot{V}(t) = -pI(t) - cV(t). \tag{5.49}$$

B) *The model in the form of P1-TS system.*
Assume that the P1-TS system has three inputs $z_1 = T(t)$, $z_2 = I(t)$ and $z_3 = V(t)$ and three outputs $S_1 = \dot{T}(t)$, $S_2 = \dot{I}(t)$ and $S_3 = \dot{V}(t)$. The input vector is a point of the cuboid $D^3 = [0, \beta_1] \times [0, \beta_2] \times [0, \beta_3]$. Every

input has assigned two membership functions of fuzzy sets; $P_k(z_k) = z_k/\beta_k$ and $N_k(z_k) = 1 - z_k/\beta_k$ for $k = 1, 2, 3$. Thus, N_k denotes *small* (or *near zero*) and P_k - *big* (or *almost maximal*). For the output vector $\mathbf{S} = \left(\dot{T}(t), \dot{I}(t), \dot{V}(t)\right)$, according to the equations (5.47)-(5.49) we formulate the following system of fuzzy rules in the matrix form:

$$\text{If} \quad [T, I, V] \quad \text{is} \quad \mathbf{M}, \quad \text{then} \quad \mathbf{S} \quad \text{is} \quad \mathbf{Q},$$

where the antecedents matrix $\mathbf{M}$ is the same as in (5.38) and the consequents matrix

$$\mathbf{Q} = \begin{bmatrix} \lambda & 0 & 0 \\ \lambda - d\beta_1 & 0 & 0 \\ \lambda & -\delta\beta_2 & -p\beta_2 \\ \lambda - d\beta_1 & -\delta\beta_2 & -p\beta_2 \\ \lambda & 0 & -c\beta_3 \\ \lambda - \beta_1(d + k\beta_3) & k\beta_1\beta_3 & -c\beta_3 \\ \lambda & -\delta\beta_2 & -(c\beta_3 + p\beta_2) \\ \lambda - \beta_1(d + k\beta_3) & k\beta_1\beta_3 - \delta\beta_2 & -(c\beta_3 + p\beta_2) \end{bmatrix}.$$

Observe that both fuzzy sets and all fuzzy rules can be easily interpreted. For example the first two fuzzy rules say that, if the number of infected cells (I) and free viruses (V) is *small*, then independently of the healthy cells population size (T), both I and V are constant (their derivatives are zero).

5.3 Examples of P1-TS Systems with Four and More Inputs

General formulas for the P1-TS systems with four and more inputs are not difficult to obtain by using symbolic computations using the appropriate software, e.g. Maple, Mathematica, MuPAD, etc. [30]. However, all matrices become huge and take up a lot of space. Therefore, instead of the general case we will consider several examples. Let us begin with a simple one.

Example 5.9. The P1-TS system with four inputs $z_k \in [-\alpha_k, \beta_k] = [0, b]$, $(b > 0)$, for $k = 1, 2, 3, 4$ and one output S, is given by the following metarules:

$$\begin{aligned} M_1 &: \text{ If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2 \text{ and } z_4 \text{ is } N_4, \text{ then } S = k, \\ M_2 &: \text{ If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2 \text{ and } z_4 \text{ is } P_4, \text{ then } S = k, \\ M_3 &: \text{ If } z_3 \text{ is } P_3, \text{ then } S = k, \\ M_4 &: \text{ otherwise } S = 0. \end{aligned}$$

Our goal is to compute the crisp system output and explain the results. The above metarules we rewrite in Table 5.7, where $q_1, \ldots, q_{16}$ denote the

Table 5.7 Look-up-table for the P1-TS fuzzy system considered in Example 5.9

$z_1, z_2 \setminus z_3, z_4 \rightarrow$ ↓	N_3N_4	N_3P_4	P_3P_4	P_3N_4
N_1N_2	$q_1 = 0$	$q_9 = 0$	$q_{13} = k$	$q_5 = k$
N_1P_2	$q_3 = 0$	$q_{11} = k$	$q_{15} = k$	$q_7 = k$
P_1P_2	$q_4 = 0$	$q_{12} = 0$	$q_{16} = k$	$q_8 = k$
P_1N_2	$q_2 = k$	$q_{10} = 0$	$q_{14} = k$	$q_6 = k$

subsequent consequent of the rules. Next, for $n = 4$ we compute the fundamental matrix, according to (B.16)-(B.20) given in Appendix B

$$\mathbf{\Omega} = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&b&0&b&0&b&0&b&0&b&0&b&0&b&0&b \\ 0&0&b&b&0&0&b&b&0&0&b&b&0&0&b&b \\ 0&0&0&b^2&0&0&0&b^2&0&0&0&b^2&0&0&0&b^2 \\ 0&0&0&0&b&b&b&b&0&0&0&0&b&b&b&b \\ 0&0&0&0&0&b^2&0&b^2&0&0&0&0&0&b^2&0&b^2 \\ 0&0&0&0&0&0&b^2&b^2&0&0&0&0&0&0&b^2&b^2 \\ 0&0&0&0&0&0&0&b^3&0&0&0&0&0&0&0&b^3 \\ 0&0&0&0&0&0&0&0&b&b&b&b&b&b&b&b \\ 0&0&0&0&0&0&0&0&0&b^2&0&b^2&0&b^2&0&b^2 \\ 0&0&0&0&0&0&0&0&0&0&b^2&b^2&0&0&b^2&b^2 \\ 0&0&0&0&0&0&0&0&0&0&0&b^3&0&0&0&b^3 \\ 0&0&0&0&0&0&0&0&0&0&0&0&b^2&b^2&b^2&b^2 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&b^3&0&b^3 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&b^3&b^3 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&b^4 \end{bmatrix}.$$

From Table 5.7, we read the successive conclusions of the rules

$$\mathbf{q} = [0, k, 0, 0, k, k, k, k, 0, 0, k, 0, k, k, k, k]^T.$$

According to equation (2.44) we obtain

$$\boldsymbol{\theta} = b^{-3}\left[0, b^2k, 0, -bk, b^2k, -bk, 0, k, 0, -bk, bk, 0, 0, k, -k, 0\right]^T,$$

and for the generator (B.15) given in Appendix B, the system output is

$$\begin{aligned} S(z_1, z_2, z_3, z_4) &= \boldsymbol{\theta}^T \mathbf{g}(z_1, z_2, z_3, z_4) \\ &= \frac{k}{b^3}\left(z_1(z_3 - b)(z_4 - b) + b^2 z_3 + z_2(z_1 - z_4)(z_3 - b)\right). \end{aligned} \tag{5.50}$$

For $b = k = 1$ we have to do with the P1-TS fuzzy system, which models a function defined in the multi-valued logic

Table 5.8 Look-up-table for the P1-TS fuzzy system as Karnaugh map

$z_1, z_2 \setminus z_3, z_4 \rightarrow$ ↓	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	0	0	1	1
10	1	0	1	1

$$S(z_1, z_2, z_3, z_4) = z_3 + z_1 (z_3 - 1)(z_4 - 1) + z_2 (z_3 - 1)(z_1 - z_4) \qquad (5.51)$$

and Table 5.7 can be viewed as the generalized Karnaugh map (see Table 5.8).

One can check that the function (5.51) takes the following values in the vertices of the hypercube $[0,1]^4$:

$$\begin{array}{llll} S(0,0,0,0) = 0, & S(0,0,0,1) = 0, & S(0,0,1,0) = 1, & S(0,0,1,1) = 1, \\ S(0,1,0,0) = 0, & S(0,1,0,1) = 1, & S(0,1,1,0) = 1, & S(0,1,1,1) = 1, \\ S(1,0,0,0) = 1, & S(1,0,0,1) = 0, & S(1,0,1,0) = 1, & S(1,0,1,1) = 1, \\ S(1,1,0,0) = 0, & S(1,1,0,1) = 0, & S(1,1,1,0) = 1, & S(1,1,1,1) = 1. \end{array}$$

Thus, the above values correspond with the ones given in Table 5.8. Now the interpretation of the metarules, individual rules, and the function to which the P1-TS system is equivalent are extremely simple.

In some cases of the nonlinear systems, it is desirable to transform the original system variables into other ones. To exemplify this idea let us consider a magnetic suspension system.

Example 5.10. Magnetic suspension systems (see Fig. 5.7) are a familiar setup that is receiving increasing attention in applications where it is essential to reduce friction force caused by mechanical contact. Such systems are commonly encountered in high-speed trains and magnetic bearings, as well as in gyroscopes and accelerometers. The conventional model in the form of differential equations derived e.g. in [89], [134] is as follows

$$\left.\begin{aligned} \dot{x}_1 &= x_2 \;, \\ \dot{x}_2 &= g - \frac{k}{m} x_2 - \frac{\lambda \mu x_3^2}{2m(1+\mu x_1)^2} \;, \\ \dot{x}_3 &= \frac{1+\mu x_1}{\lambda} \left(-R x_3 + \frac{\lambda \mu x_2 x_3}{(1+\mu x_1)^2} + v \right) . \end{aligned}\right\} \qquad (5.52)$$

The state variables are $x_1 \geq 0$ - the vertical (downward) position of the ball optically measured [200] from a reference point [m], $x_2 = \dot{x}_1$ [m / s] is

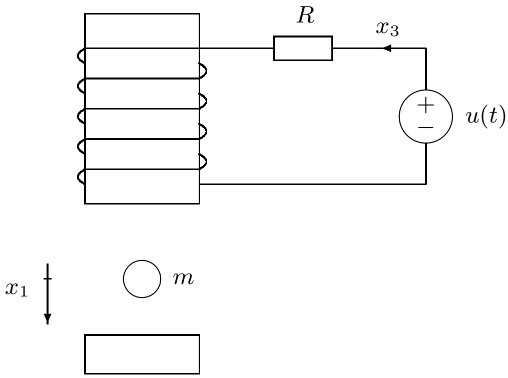


Fig. 5.7 Magnetic suspension system

the ball speed and x_3 is the electric current of the electromagnet [A]. The system constants are g - the acceleration due to gravity $\left[\mathrm{m}/\mathrm{s}^2\right]$, m - the mass of the ball [kg], k - the viscous friction coefficient [N / (m / s)] and R - the series resistance of the circuit [Ω]. The quantities μ [1/ m] and λ [H] are positive constants such that the inductance of the electromagnet depends on the position x_1 of the ball as follows

$$L\left(x_1\right) = \frac{\lambda}{1+\mu x_1}. \tag{5.53}$$

The control signal $u\left(t\right)$ is the voltage. We assume real restrictions on the original system variables. The vertical position of the ball $0 \le x_1^L \le x_1\left(t\right) \le x_1^H$, where $x_1\left(t\right) = 0$ if the ball is next to the coil, the ball speed $-x_2^H \le x_2\left(t\right) \le x_2^H$ and the current in the electric circuit $-x_3^H \le x_3\left(t\right) \le x_3^H$. In order to obtain a P1-TS system let us define new state variables

$$\begin{bmatrix} w_1\left(t\right) \\ w_2\left(t\right) \\ w_3\left(t\right) \end{bmatrix} = \begin{bmatrix} x_1\left(t\right) \\ x_2\left(t\right) \\ x_3\left(t\right) / \left(1+\mu x_1\left(t\right)\right) \end{bmatrix}, \qquad \forall\, t \ge 0. \tag{5.54}$$

Thus, $w_3\left(t\right)$ is the electric current related to the ball position. Observe that the transformation (5.54) is one-to-one. Thus, knowing the signals $w_1\left(t\right)$, $w_2\left(t\right)$ and $w_3\left(t\right)$, we immediately obtain the solution $x_1\left(t\right)$, $x_2\left(t\right)$ and $x_3\left(t\right)$ of the original system (5.52) and vice-versa. Suppose we change the variables according to (5.54). Thus,

$$\dot{w}_1 = \dot{x}_1 = x_2 = w_2,$$

and

$$\dot{w}_2 = \dot{x}_2 = g - \frac{k}{m}x_2 - \frac{\lambda\mu}{2m}\left(\frac{x_3}{1+\mu x_1}\right)^2 = g - \frac{k}{m}w_2 - \frac{\lambda\mu}{2m}w_3^2.$$

Taking into account the third variable from (5.54) we have

$$\dot{w}_3 = \frac{\dot{x}_3 (1 + \mu x_1) - x_3 \mu \dot{x}_1}{(1 + \mu x_1)^2}.$$

According to the third equation of (5.52) we obtain

$$\begin{aligned}\dot{w}_3 &= \frac{1}{(1 + \mu x_1)} \frac{(1 + \mu x_1)}{\lambda} \left(-R x_3 + \frac{\lambda \mu x_2 x_3}{(1 + \mu x_1)^2} + u \right) - \frac{x_3 \mu x_2}{(1 + \mu x_1)^2} \\ &= \frac{1}{\lambda} (-R x_3 + u).\end{aligned}$$

From the equality $x_3 = (1 + \mu x_1) w_3$ we finally obtain the system of the differential equations

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{bmatrix} = \begin{bmatrix} w_2 \\ g - \frac{k}{m} w_2 - \frac{\lambda \mu}{2m} w_3^2 \\ -\frac{R}{\lambda} w_3 - \frac{R}{\lambda} \mu w_1 w_3 + \frac{u}{\lambda} \end{bmatrix}, \tag{5.55}$$

which is equivalent to the original one (5.52).

One can express the system (5.55) by 27 highly interpretable fuzzy rules for the P2-TS system with the inputs w_1, w_2 and w_3. In order to reduce the number of rules, we will show that the same system can be exactly described by the zero-order P1-TS system with four inputs and three outputs. The inputs and the outputs of the P1-TS system we define as shown in Fig. 5.8:

- $z_1 = w_1(t) \in [-\alpha_1, \beta_1]$, where $\alpha_1 \leq 0$ and $\beta_1 = x_1^H$,
- $z_2 = w_2(t) \in [-\alpha_2, \beta_2]$, where $\alpha_2 = \beta_2 = x_2^H > 0$,
- $z_3 = w_3(t) \in [-\alpha_3, \beta_3]$, by $-\alpha_3 = -x_3^H / \left(1 + \mu x_1^L\right)$ and $\beta_3 = x_3^H / \left(1 + \mu x_1^L\right)$,
- $z_4 = w_3^2(t) \in [-\alpha_4, \beta_4]$, where $\alpha_4 = 0$ and $\beta_4 = x_3^H / \left(1 + \mu x_1^L\right)^2$,
- $S_1 = \dot{w}_1(t)$, $S_2 = \dot{w}_2(t)$ and $S_3 = \dot{w}_3(t)$.

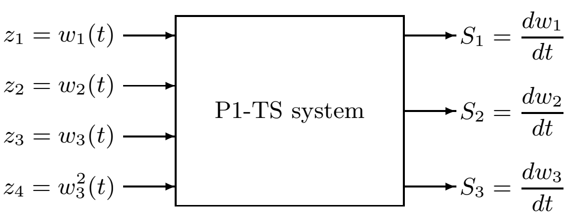


Fig. 5.8 Inputs and outputs of the P1-TS as a model of the magnetic suspension system

We assume that the control action u is viewed as a crisp function and no fuzzy sets are assigned to u. The system generator $\mathbf{g}(z_1, z_2, z_3, z_4)$ is given by (B.15) in Appendix B. Thus,

$$\mathbf{g}^T\left(w_1, w_2, w_3, w_3^2\right) = [1, w_1, w_2, w_1 w_2, w_3, w_1 w_3, w_2 w_3, w_1 w_2 w_3, w_3^2, w_1 w_3^2, w_2 w_3^2, w_1 w_2 w_3^2, w_3^3, w_1 w_3^3, w_2 w_3^3, w_1 w_2 w_3^3].$$

For such generator the equations (5.55) can be equivalently written as

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{bmatrix} = \boldsymbol{\Theta}^T \mathbf{g}\left(w_1, w_2, w_3, w_3^2\right), \tag{5.56}$$

where

$$\boldsymbol{\Theta}^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0\,0 \\ g & 0 & -\dfrac{k}{m} & 0 & 0 & 0 & 0 & 0 & -\dfrac{\lambda\mu}{2m} & 0\,0\,0\,0\,0\,0\,0 \\ \dfrac{u}{\lambda} & 0 & 0 & 0 & -\dfrac{R}{\lambda} & -\dfrac{R\mu}{\lambda} & 0 & 0 & 0 & 0\,0\,0\,0\,0\,0\,0 \end{bmatrix}. \tag{5.57}$$

For $n = 4$ the fundamental matrix $\boldsymbol{\Omega}$ is given by (B.16)-(B.20) in Appendix B. According to the formula (2.52) and the matrix of function coefficients (5.57), we immediately obtain the matrix $\mathbf{Q}$ of the consequents of the fuzzy rules, containing three column vectors as the conclusions of the fuzzy rules, assigned to the outputs S_1-S_3. By $\alpha_1 = \alpha_4 = 0$ we obtain the following system of fuzzy rules

$$\text{If } [z_1, z_2, z_3, z_4] \text{ is } \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \\ P_1 & N_2 & N_3 & N_4 \\ N_1 & P_2 & N_3 & N_4 \\ P_1 & P_2 & N_3 & N_4 \\ N_1 & N_2 & P_3 & N_4 \\ P_1 & N_2 & P_3 & N_4 \\ N_1 & P_2 & P_3 & N_4 \\ P_1 & P_2 & P_3 & N_4 \\ N_1 & N_2 & N_3 & P_4 \\ P_1 & N_2 & N_3 & P_4 \\ N_1 & P_2 & N_3 & P_4 \\ P_1 & P_2 & N_3 & P_4 \\ N_1 & N_2 & P_3 & P_4 \\ P_1 & N_2 & P_3 & P_4 \\ N_1 & P_2 & P_3 & P_4 \\ P_1 & P_2 & P_3 & P_4 \end{bmatrix}, \text{ then } [\dot{w}_1, \dot{w}_2, \dot{w}_3] \text{ is } \begin{bmatrix} -\alpha_2 & B_1 & C_1 \\ -\alpha_2 & B_1 & C_2 \\ \beta_2 & B_2 & C_1 \\ \beta_2 & B_2 & C_2 \\ -\alpha_2 & B_1 & C_3 \\ -\alpha_2 & B_1 & C_4 \\ \beta_2 & B_2 & C_3 \\ \beta_2 & B_2 & C_4 \\ -\alpha_2 & B_3 & C_1 \\ -\alpha_2 & B_3 & C_2 \\ \beta_2 & B_4 & C_1 \\ \beta_2 & B_4 & C_2 \\ -\alpha_2 & B_3 & C_3 \\ -\alpha_2 & B_3 & C_4 \\ \beta_2 & B_4 & C_3 \\ \beta_2 & B_4 & C_4 \end{bmatrix}, \tag{5.58}$$

where

$$B_1 = \frac{k\alpha_2 + gm}{m}, \quad B_2 = \frac{-k\beta_2 + gm}{m}, \tag{5.59}$$

$$B_3 = \frac{2k\alpha_2 + 2gm - \lambda\mu\beta_4}{2m}, \quad B_4 = \frac{-2k\beta_2 + 2gm - \lambda\mu\beta_4}{2m}, \tag{5.60}$$

$$C_1 = \frac{u + R\alpha_3}{\lambda}, \quad C_2 = \frac{u + R\alpha_3 + R\mu\beta_1\alpha_3}{\lambda}, \tag{5.61}$$

$$C_3 = \frac{u - R\beta_3}{\lambda}, \quad C_4 = \frac{u - R\beta_3 - R\mu\beta_1\beta_3}{\lambda}. \tag{5.62}$$

Thus, the consequents of the fuzzy rules depend on the system constants g, m, k, R, the crisp control action $u = u(t)$ and the location of the hypercuboid D^4 in the space $\mathbb{R}^4$.

As one can see, the magnetic suspension system described by the differential equations (5.52) with original variables (x_1, x_2, x_3) is equivalent to (5.55) by new variables (w_1, w_2, w_3) associated with the original ones by (5.54). This highly nonlinear system is exactly modeled by the P1-TS system defined by 16 fuzzy rules.

Example 5.11. Taking into account Example 5.10 we will show that we can simplify the modeling process by combining various models. First of all, it is not necessary to model the first equation of (5.55) at all, since it is too simple. The second equation of (5.55) can be modeled by P2-TS system with two inputs w_2 and w_3, since it has a very simple interpretation. The third equation of (5.55) can be modeled by P1-TS system with two inputs w_1 and w_3, because it has a very simple interpretation and a low number of fuzzy rules (see Fig. 5.9).

For the input variables w_2 and w_3 the membership functions of fuzzy sets are $N_k(w_k)$, $Z_k(w_k)$ and $P_k(w_k)$, given by (4.24)-(4.26), with $\sigma_k = (-\alpha_k + \beta_k)/2$ for $k = 2, 3$, assuming some parameters λ_k for the quadratic membership functions, say $\lambda_2 = \lambda_3 = 1$. According to (4.33) we have the generator

$$\mathbf{g}_2(w_2, w_3) = \left[1, w_2, w_2^2, w_3, w_2w_3, w_2^2w_3, w_3^2, w_2w_3^2, w_2^2w_3^2\right]^T, \tag{5.63}$$

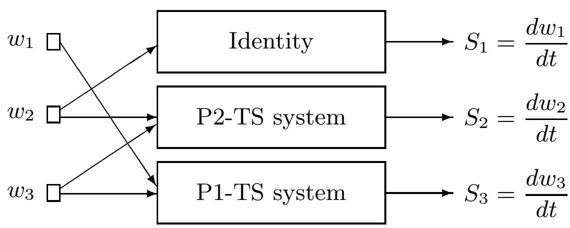


Fig. 5.9 The architecture of the rule-based system from Example 5.11

and on the basis of (4.50) after changing the names of the boundaries of intervals from $(\alpha_1, \alpha_2, \beta_1, \beta_2)$ into $(\alpha_2, \alpha_3, \beta_2, \beta_3)$, we obtain the fundamental matrix for the P2-TS system

$$\mathbf{\Omega}_2^T = \begin{bmatrix} 1 & -\alpha_2 & \alpha_2^2 & -\alpha_3 & \alpha_2\alpha_3 & -\alpha_2^2\alpha_3 & \alpha_3^2 & -\alpha_2\alpha_3^2 & \alpha_2^2\alpha_3^2 \\ 1 & \sigma_2 & -\alpha_2\beta_2 & -\alpha_3 & -\sigma_2\alpha_3 & \alpha_2\alpha_3\beta_2 & \alpha_3^2 & \sigma_2\alpha_3^2 & -\alpha_2\alpha_3^2\beta_2 \\ 1 & \beta_2 & \beta_2^2 & -\alpha_3 & -\alpha_3\beta_2 & -\alpha_3\beta_2^2 & \alpha_3^2 & \alpha_3^2\beta_2 & \alpha_3^2\beta_2^2 \\ 1 & -\alpha_2 & \alpha_2^2 & \sigma_3 & -\alpha_2\sigma_3 & \alpha_2^2\sigma_3 & -\alpha_3\beta_3 & \alpha_2\alpha_3\beta_3 & -\alpha_2^2\alpha_3\beta_3 \\ 1 & \sigma_2 & -\alpha_2\beta_2 & \sigma_3 & \sigma_2\sigma_3 & -\alpha_2\beta_2\sigma_3 & -\alpha_3\beta_3 & -\sigma_2\alpha_3\beta_3 & \alpha_2\alpha_3\beta_2\beta_3 \\ 1 & \beta_2 & \beta_2^2 & \sigma_3 & \beta_2\sigma_3 & \beta_2^2\sigma_3 & -\alpha_3\beta_3 & -\alpha_3\beta_2\beta_3 & -\alpha_3\beta_2^2\beta_3 \\ 1 & -\alpha_2 & \alpha_2^2 & \beta_3 & -\alpha_2\beta_3 & \alpha_2^2\beta_3 & \beta_3^2 & -\alpha_2\beta_3^2 & \alpha_2^2\beta_3^2 \\ 1 & \sigma_2 & -\alpha_2\beta_2 & \beta_3 & \sigma_2\beta_3 & -\alpha_2\beta_2\beta_3 & \beta_3^2 & \sigma_2\beta_3^2 & -\alpha_2\beta_2\beta_3^2 \\ 1 & \beta_2 & \beta_2^2 & \beta_3 & \beta_2\beta_3 & \beta_2^2\beta_3 & \beta_3^2 & \beta_2\beta_3^2 & \beta_2^2\beta_3^2 \end{bmatrix}. \tag{5.64}$$

For the output variable $\dot{w}_2$, the vector of the function coefficients is given by

$$\boldsymbol{\theta} = \left[g, \ -\frac{k}{m}, \ 0, \ 0, \ 0, \ 0, \ -\frac{\lambda\mu}{2m}, \ 0, \ 0 \right]^T.$$

For the given $\mathbf{\Omega}_2$ and $\boldsymbol{\theta}$, according to (4.37) we obtain the conclusions of the fuzzy rules; $\mathbf{q} = \mathbf{\Omega}_2^T \boldsymbol{\theta}$. Finally, we obtain the following system of rules for the P2-TS subsystem

$$\text{If } [w_2, w_3] \text{ is } \begin{bmatrix} N_1 & N_2 \\ Z_1 & N_2 \\ P_1 & N_2 \\ N_1 & Z_2 \\ Z_1 & Z_2 \\ P_1 & Z_2 \\ N_1 & P_2 \\ Z_1 & P_2 \\ P_1 & P_2 \end{bmatrix}, \text{ then } [\dot{w}_2] = \begin{bmatrix} g + k\alpha_2/m - \lambda\mu\alpha_3^2/(2m) \\ g - k\sigma_2/m - \lambda\mu\alpha_3^2/(2m) \\ g - k\beta_2/m - \lambda\mu\alpha_3^2/(2m) \\ g + k\alpha_2/m + \lambda\mu\alpha_3\beta_3/(2m) \\ g - k\sigma_2/m + \lambda\mu\alpha_3\beta_3/(2m) \\ g - k\beta_2/m + \lambda\mu\alpha_3\beta_3/(2m) \\ g + k\alpha_2/m - \lambda\mu\beta_3^2/(2m) \\ g - k\sigma_2/m - \lambda\mu\beta_3^2/(2m) \\ g - k\beta_2/m - \lambda\mu\beta_3^2/(2m) \end{bmatrix},$$

that exactly models the second equation of (5.55). The values α_k and β_k for $k = 2, 3$ are the same as in Example 5.10. A model building for the third equation in (5.55) in the form of the P1-TS system is simple and will be skipped.

In the next sections we will consider examples of modeling more complicated dynamical systems.

5.3.1 Low Order Atmospheric Circulation Model

Consider a low order atmospheric circulation model described in [122], [174] in the form of the following three nonlinear differential equations

$$\begin{cases} \dot{x} = -ax - y^2 - z^2 + aF, \\ \dot{y} = -y + xy - bxz + G, \\ \dot{z} = -z + bxy + xz, \end{cases} \tag{5.65}$$

where x represents the strength of the globally averaged westerly current, and y, z are the strength of the cosine and sine phases of a chain of superposed waves. The unit of the variable t is equal to the damping time of the waves, estimated to be five days. The terms in F and G represent thermal forcing terms, and the parameter b stands for the strength of the advection of the waves by the westerly current. Here, F, G are treated as control parameters, with $a = 1/4$ and $b = 4$ [174].

We will show that this system can be exactly described by the zero-order 5-input and 3-output P1-TS system, in which the inputs are $z_1 = x(t) \in [-\alpha_1, \beta_1]$, $z_2 = y(t) \in [-\alpha_2, \beta_2]$, $z_3 = y^2(t) \in [-\alpha_3, \beta_3]$, by $-\alpha_3 = \min\{\alpha_2^2, \beta_2^2\}$, $\beta_3 = \max\{\alpha_2^2, \beta_2^2\}$, $z_4 = z \in [-\alpha_4, \beta_4]$, and $z_5 = z^2 \in [-\alpha_5, \beta_5]$, where $-\alpha_5 = \min\{\alpha_4^2, \beta_4^2\}$, $\beta_5 = \max\{\alpha_4^2, \beta_4^2\}$. Additionally we add two controls $u_1 = aF$ and $u_2 = G$. There are three outputs of the system: $S_1 = \dot{x}(t)$, $S_2 = \dot{y}(t)$ and $S_3 = \dot{z}(t)$ as shown in Fig. 5.10.

The generator for the P1-TS system with $n = 5$ inputs is as follows

$$\begin{aligned} \mathbf{g}(z_1, z_2, z_3, z_4, z_5) = [&1, z_1, z_2, z_1z_2, z_3, z_1z_3, z_2z_3, z_1z_2z_3, z_4, z_1z_4, z_2z_4, \\ &z_1z_2z_4, z_3z_4, z_1z_3z_4, z_2z_3z_4, z_1z_2z_3z_4, z_5, z_1z_5, \\ &z_2z_5, z_1z_2z_5, z_3z_5, z_1z_3z_5, z_2z_3z_5, z_1z_2z_3z_5, z_4z_5, \\ &z_1z_4z_5, z_2z_4z_5, z_1z_2z_4z_5, z_3z_4z_5, z_1z_3z_4z_5, \\ &z_2z_3z_4z_5, z_1z_2z_3z_4z_5]. \end{aligned} \tag{5.66}$$

Thus, the equations (5.65) can be equivalently written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \boldsymbol{\Theta}^T \mathbf{g}\left(x, y, y^2, z, z^2\right), \tag{5.67}$$

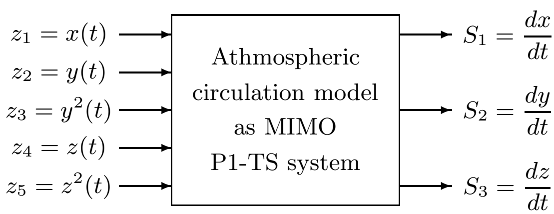


Fig. 5.10 The inputs and the outputs of the MIMO P1-TS system modeling the athmospheric circulation

where

$$\mathbf{\Theta}^T = \begin{bmatrix} u_1 & -a & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 \\ u_2 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -b & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & b & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{5.68}$$

The fundamental matrix contains 1024 elements and can be obtained using recursion (2.43):

$$\mathbf{\Omega}_5 = \begin{bmatrix} \mathbf{\Omega}_4 & \mathbf{\Omega}_4 \\ -\alpha_5 \mathbf{\Omega}_4 & \beta_5 \mathbf{\Omega}_4 \end{bmatrix} \in \mathbb{R}^{32\times 32}, \tag{5.69}$$

where $\mathbf{\Omega}_4$ is given by (B.16)-(B.20) in Appendix B. According to the formula for MIMO P1-TS systems (2.52), the fundamental matrix for $n = 5$ and (5.68) we obtain the matrix

$$\mathbf{Q} = \mathbf{\Omega}^T \mathbf{\Theta} = \begin{bmatrix} \mathbf{Q}_a \\ \mathbf{Q}_b \\ \mathbf{Q}_c \\ \mathbf{Q}_d \\ \mathbf{Q}_e \\ \mathbf{Q}_f \\ \mathbf{Q}_g \\ \mathbf{Q}_h \end{bmatrix}, \tag{5.70}$$

where

$$\mathbf{Q}_a = \begin{bmatrix} u_1 + \alpha_3 + \alpha_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 + b\alpha_1\alpha_2 \\ u_1 + \alpha_3 + \alpha_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 - b\alpha_2\beta_1 \\ u_1 + \alpha_3 + \alpha_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 - b\alpha_1\beta_2 \\ u_1 + \alpha_3 + \alpha_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.71}$$

$$\mathbf{Q}_b = \begin{bmatrix} u_1 - \beta_3 + \alpha_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 + b\alpha_1\alpha_2 \\ u_1 - \beta_3 + \alpha_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 - b\alpha_2\beta_1 \\ u_1 - \beta_3 + \alpha_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 - b\alpha_1\beta_2 \\ u_1 - \beta_3 + \alpha_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.72}$$

$$\mathbf{Q}_c = \begin{bmatrix} u_1 + \alpha_3 + \alpha_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 + b\alpha_1\alpha_2 \\ u_1 + \alpha_3 + \alpha_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 - b\alpha_2\beta_1 \\ u_1 + \alpha_3 + \alpha_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 - b\alpha_1\beta_2 \\ u_1 + \alpha_3 + \alpha_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.73}$$

$$\mathbf{Q}_d = \begin{bmatrix} u_1 - \beta_3 + \alpha_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 + b\alpha_1\alpha_2 \\ u_1 - \beta_3 + \alpha_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 - b\alpha_2\beta_1 \\ u_1 - \beta_3 + \alpha_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 - b\alpha_1\beta_2 \\ u_1 - \beta_3 + \alpha_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 + b\beta_1\beta_2 \end{bmatrix} \tag{5.74}$$

$$\mathbf{Q}_e = \begin{bmatrix} u_1 + \alpha_3 - \beta_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 + b\alpha_1\alpha_2 \\ u_1 + \alpha_3 - \beta_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 - b\alpha_2\beta_1 \\ u_1 + \alpha_3 - \beta_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 - b\alpha_1\beta_2 \\ u_1 + \alpha_3 - \beta_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.75}$$

$$\mathbf{Q}_f = \begin{bmatrix} u_1 - \beta_3 - \beta_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 + b\alpha_1\alpha_2 \\ u_1 - \beta_3 - \beta_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 - b\alpha_2\beta_1 \\ u_1 - \beta_3 - \beta_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 - b\alpha_1\alpha_4, \; \alpha_4 + \alpha_1\alpha_4 - b\alpha_1\beta_2 \\ u_1 - \beta_3 - \beta_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 + b\beta_1\alpha_4, \; \alpha_4 - \beta_1\alpha_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.76}$$

$$\mathbf{Q}_g = \begin{bmatrix} u_1 + \alpha_3 - \beta_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 + b\alpha_1\alpha_2 \\ u_1 + \alpha_3 - \beta_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 - b\alpha_2\beta_1 \\ u_1 + \alpha_3 - \beta_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 - b\alpha_1\beta_2 \\ u_1 + \alpha_3 - \beta_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 + b\beta_1\beta_2 \end{bmatrix}, \tag{5.77}$$

$$\mathbf{Q}_h = \begin{bmatrix} u_1 - \beta_3 - \beta_5 + a\alpha_1, \; u_2 + \alpha_2 + \alpha_1\alpha_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 + b\alpha_1\alpha_2 \\ u_1 - \beta_3 - \beta_5 - a\beta_1, \; u_2 + \alpha_2 - \alpha_2\beta_1 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 - b\alpha_2\beta_1 \\ u_1 - \beta_3 - \beta_5 + a\alpha_1, \; u_2 - \beta_2 - \alpha_1\beta_2 + b\alpha_1\beta_4, \; -\beta_4 - \alpha_1\beta_4 - b\alpha_1\beta_2 \\ u_1 - \beta_3 - \beta_5 - a\beta_1, \; u_2 - \beta_2 + \beta_1\beta_2 - b\beta_1\beta_4, \; -\beta_4 + \beta_1\beta_4 + b\beta_1\beta_2 \end{bmatrix}. \tag{5.78}$$

The above submatrices contain three column vectors as the conclusions of the fuzzy rules, assigned to the outputs S_1, S_2 and S_3, respectively. Thus, the considered system (5.65) is equivalent to the P1-TS one defined by 32 fuzzy rules in the matrix form

$$\text{If} \quad \left[x, y, y^2, z, z^2\right] \quad \text{is} \quad \mathbf{M}, \quad \text{then} \quad [\dot{x}, \dot{y}, \dot{z}] \quad \text{is} \quad \mathbf{Q}, \tag{5.79}$$

where the antecedents matrix $\mathbf{M}$ containing the labels of the fuzzy sets has the following structure

$$\mathbf{M} = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 & N_5 \\ P_1 & N_2 & N_3 & N_4 & N_5 \\ N_1 & P_2 & N_3 & N_4 & N_5 \\ P_1 & P_2 & N_3 & N_4 & N_5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix}_{32\times 5}, \tag{5.80}$$

and the consequents matrix is given by (5.70)-(5.78). All consequents of the fuzzy rules are real numbers depending on the given constants a, b, the control parameters F and G, and "$-\alpha_k$" and "β_k", $(k = 1, \ldots, 5)$, as stated above.

5.3.2 Induction Motor Model

Using the theory of P1-TS systems we can exactly model some class of highly nonlinear fifth-order dynamical systems. Consider the following continuous-time model of an induction motor developed in [133], which includes both electrical and mechanical dynamics under the assumption of linear magnetic circuits

$$\frac{di_{sa}}{dt} = \frac{MR_r}{\sigma L_s L_r^2}\psi_{ra} + \frac{n_p M}{\sigma L_s L_r}\omega\psi_{rb} - \left(\frac{M^2R_r + L_r^2R_s}{\sigma L_s L_r^2}\right) i_{sa} + \frac{1}{\sigma L_s}u_{sa} \ ,$$

$$\frac{di_{sb}}{dt} = -\frac{n_p M}{\sigma L_s L_r}\omega\psi_{ra} + \frac{MR_r}{\sigma L_s L_r^2}\psi_{rb} - \left(\frac{M^2R_r + L_r^2R_s}{\sigma L_s L_r^2}\right) i_{sb} + \frac{1}{\sigma L_s}u_{sb} \ ,$$

$$\frac{d\psi_{ra}}{dt} = -\frac{R_r}{L_r}\psi_{ra} - n_p\omega\psi_{rb} + \frac{R_r}{L_r}Mi_{sa} \ ,$$

$$\frac{d\psi_{rb}}{dt} = n_p\omega\psi_{ra} - \frac{R_r}{L_r}\psi_{rb} + \frac{R_r}{L_r}Mi_{sb} \ ,$$

$$\frac{d\omega}{dt} = \frac{n_p M}{JL_r}\left(\psi_{ra}i_{sb} - \psi_{rb}i_{sa}\right) - \frac{T_L}{J} \ ,$$

where the subscripts s and r stand for stator and rotor, (a,b) denote the components of a vector with respect to a fixed stator reference frame. The meanings of the symbols are as follows:

i_s - stator current [A],
ψ_s - stator flux linkage [Wb],
i_r - rotor current [A],
u_s - stator voltage input to the machine [V],
ψ_r - rotor flux linkage, (e.g. 1.3 [Wb]) rated,
R_s - stator resistance, (e.g. 0.18 [Ω]),
R_r - rotor resistance, (e.g. 0.15 [Ω]),
M - mutual inductance, (e.g. 0.068 [H]),
L_s - stator inductance, (e.g. 0.0699 [H]),
L_r - rotor inductance, (e.g. 0.0699 [H]),
n_p - number of pole pairs of the induction machine, (e.g. $n_p = 1$),
ω - the angular speed of the rotor, (e.g. 220 [rad / s]) rated,
T_L - load torque, (e.g. 70 [N m]) rated,
J - rotor inertia, (e.g. 0.0586 $\left[\mathrm{kg\,m^2}\right]$).

After substituting the constants

$$\frac{R_r}{L_r} = \frac{1}{T_r}, \quad \sigma = 1 - \frac{M^2}{L_s L_r^2}, \quad K = \frac{M}{\sigma L_s L_r}, \quad \gamma = \frac{R_s}{\sigma L_s} + \frac{R_r M^2}{\sigma L_s L_r^2}, \quad n_p = p,$$

we can write the continuous-time state equations

$$\left.\begin{aligned}
\frac{dz_1}{dt} &= -\gamma z_1(t) + \frac{K}{T_r} z_3(t) + pK z_4(t) z_5(t) + \frac{1}{\sigma L_s} u_{sa}(t) \\
\frac{dz_2}{dt} &= -\gamma z_2(t) - pK z_5(t) z_3(t) + \frac{K}{T_r} z_4(t) + \frac{1}{\sigma L_s} u_{sb}(t) \\
\frac{dz_3}{dt} &= \frac{M}{T_r} z_1(t) - \frac{1}{T_r} z_3(t) - p z_5(t) z_4(t) \\
\frac{dz_4}{dt} &= \frac{M}{T_r} z_2(t) - \frac{1}{T_r} z_4(t) + p z_5(t) z_3(t) \\
\frac{dz_5}{dt} &= \frac{pM}{JL_r}\left(z_3(t) z_2(t) - z_4(t) z_1(t)\right) - \frac{T_L}{J},
\end{aligned}\right\} \tag{5.81}$$

or using Euler discretization of step size h, the discrete-time state equations, as in [17], [109]

$$\begin{aligned}
z_1(k+1) &= z_1(k) + h\left(-\gamma z_1(k) + \frac{K}{T_r} z_3(k) + pK z_4(k) z_5(k) + \frac{u_{sa}(k)}{\sigma L_s}\right), \\
z_2(k+1) &= z_2(k) + h\left(-\gamma z_2(k) - pK z_5(k) z_3(k) + \frac{K}{T_r} z_4(k) + \frac{u_{sb}(k)}{\sigma L_s}\right), \\
z_3(k+1) &= z_3(k) + h\left(\frac{M}{T_r} z_1(k) - \frac{1}{T_r} z_3(k) - p z_5(k) z_4(k)\right), \\
z_4(k+1) &= z_4(k) + h\left(\frac{M}{T_r} z_2(k) + p z_5(k) z_3(k) - \frac{1}{T_r} z_4(k)\right), \\
z_5(k+1) &= z_5(k) + h\left(\frac{pM}{JL_r}\left(z_3(k) z_2(k) - z_4(k) z_1(k)\right) - \frac{T_L}{J}\right).
\end{aligned}$$

The state equations of the induction motor in the continuous or discrete-time form, can be exactly modeled by the MIMO P1-TS rule-based system with the outputs $S_1, \ldots, S_5$. For the continuous model they are the derivatives of state variables (see Fig. 5.11)

$$S_1 = \frac{di_{sa}}{dt}, \qquad S_2 = \frac{di_{sb}}{dt}, \qquad S_3 = \frac{d\psi_{ra}}{dt}, \qquad S_4 = \frac{d\psi_{rb}}{dt}, \qquad S_5 = \frac{d\omega}{dt}.$$

The inputs of the P1-TS system include both all five state variables

$$\begin{aligned}
i_{sa}(t) &= z_1(t) \in [-\alpha_1, \beta_1], \quad i_{sb}(t) = z_2(t) \in [-\alpha_2, \beta_2], \\
\psi_{ra}(t) &= z_3(t) \in [-\alpha_3, \beta_3], \quad \psi_{rb}(t) = z_4(t) \in [-\alpha_4, \beta_4],
\end{aligned}$$

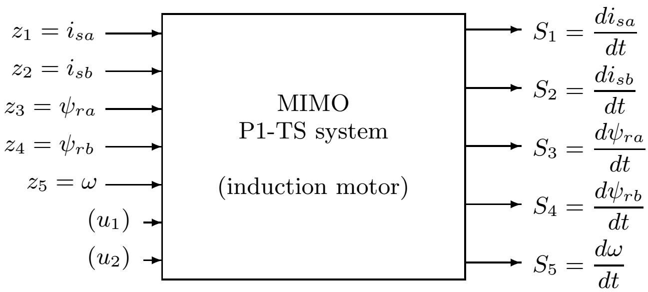


Fig. 5.11 MIMO zero-order P1-TS system as an exact continuous-time model of an induction motor

$$\omega(t) = z_5(t) \in [-\alpha_5, \beta_5],$$

and, additionally, two control signals

$$u_1(t) = (\sigma L_s)^{-1} u_{sa}(t), \qquad u_2(t) = (\sigma L_s)^{-1} u_{sb}(t).$$

In such case the number of rules would be 128. We can substantially reduce the number of rules by reducing the inputs into the state variables, because the state derivatives of the current (velocities of i_{sa} and i_{sb}, respectively) depend linearly on the control signals u_1 and u_2. Thus, formally u_1 and u_2 are not considered as variables of the "If" parts of the fuzzy rules, but as the parameters - in contrast to the variables $z_1, \ldots, z_5$, (Fig. 5.11).

As a result we will consider $2^5 = 32$ fuzzy rules. In order to avoid mistakes we can write down the successive vertices of the hypercuboid D^5:

$$\begin{aligned}
\boldsymbol{\gamma}_{00000} &= \boldsymbol{\gamma}_1 = (-\alpha_1, -\alpha_2, -\alpha_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{10000} &= \boldsymbol{\gamma}_2 = (\beta_1, -\alpha_2, -\alpha_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{01000} &= \boldsymbol{\gamma}_3 = (-\alpha_1, \beta_2, -\alpha_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{11000} &= \boldsymbol{\gamma}_4 = (\beta_1, \beta_2, -\alpha_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{00100} &= \boldsymbol{\gamma}_5 = (-\alpha_1, -\alpha_2, \beta_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{10100} &= \boldsymbol{\gamma}_6 = (\beta_1, -\alpha_2, \beta_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{01100} &= \boldsymbol{\gamma}_7 = (-\alpha_1, \beta_2, \beta_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{11100} &= \boldsymbol{\gamma}_8 = (\beta_1, \beta_2, \beta_3, -\alpha_4, -\alpha_5),\\
\boldsymbol{\gamma}_{00010} &= \boldsymbol{\gamma}_9 = (-\alpha_1, -\alpha_2, -\alpha_3, \beta_4, -\alpha_5),\\
\boldsymbol{\gamma}_{10010} &= \boldsymbol{\gamma}_{10} = (\beta_1, -\alpha_2, -\alpha_3, \beta_4, -\alpha_5),\\
\boldsymbol{\gamma}_{01010} &= \boldsymbol{\gamma}_{11} = (-\alpha_1, \beta_2, -\alpha_3, \beta_4, -\alpha_5),\\
\boldsymbol{\gamma}_{11010} &= \boldsymbol{\gamma}_{12} = (\beta_1, \beta_2, -\alpha_3, \beta_4, -\alpha_5),\\
\boldsymbol{\gamma}_{00110} &= \boldsymbol{\gamma}_{13} = (-\alpha_1, -\alpha_2, \beta_3, \beta_4, -\alpha_5),\\
\boldsymbol{\gamma}_{10110} &= \boldsymbol{\gamma}_{14} = (\beta_1, -\alpha_2, \beta_3, \beta_4, -\alpha_5),
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\gamma}_{01110} &= \boldsymbol{\gamma}_{15} = (-\alpha_1, \beta_2, \beta_3, \beta_4, -\alpha_5)\,, \\
\boldsymbol{\gamma}_{11110} &= \boldsymbol{\gamma}_{16} = (\beta_1, \beta_2, \beta_3, \beta_4, -\alpha_5)\,, \\
\boldsymbol{\gamma}_{00001} &= \boldsymbol{\gamma}_{17} = (-\alpha_1, -\alpha_2, -\alpha_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{10001} &= \boldsymbol{\gamma}_{18} = (\beta_1, -\alpha_2, -\alpha_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{01001} &= \boldsymbol{\gamma}_{19} = (-\alpha_1, \beta_2, -\alpha_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{11001} &= \boldsymbol{\gamma}_{20} = (\beta_1, \beta_2, -\alpha_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{00101} &= \boldsymbol{\gamma}_{21} = (-\alpha_1, -\alpha_2, \beta_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{10101} &= \boldsymbol{\gamma}_{22} = (\beta_1, -\alpha_2, \beta_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{01101} &= \boldsymbol{\gamma}_{23} = (-\alpha_1, \beta_2, \beta_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{11101} &= \boldsymbol{\gamma}_{24} = (\beta_1, \beta_2, \beta_3, -\alpha_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{00011} &= \boldsymbol{\gamma}_{25} = (-\alpha_1, -\alpha_2, -\alpha_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{10011} &= \boldsymbol{\gamma}_{26} = (\beta_1, -\alpha_2, -\alpha_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{01011} &= \boldsymbol{\gamma}_{27} = (-\alpha_1, \beta_2, -\alpha_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{11011} &= \boldsymbol{\gamma}_{28} = (\beta_1, \beta_2, -\alpha_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{00111} &= \boldsymbol{\gamma}_{29} = (-\alpha_1, -\alpha_2, \beta_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{10111} &= \boldsymbol{\gamma}_{30} = (\beta_1, -\alpha_2, \beta_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{01111} &= \boldsymbol{\gamma}_{31} = (-\alpha_1, \beta_2, \beta_3, \beta_4, \beta_5)\,, \\
\boldsymbol{\gamma}_{11111} &= \boldsymbol{\gamma}_{32} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)\,.
\end{aligned}$$

According to (2.48)-(2.50) the system of rules has the following form

$$\text{If } [z_1, z_2, z_3, z_4, z_5] \text{ is } \mathbf{M}, \text{ then } [S_1, S_2, S_3, S_4, S_5] \text{ is } \mathbf{Q}, \tag{5.82}$$

where the antecedents matrix is the same as in (5.80) and the consequents matrix $\mathbf{Q}$ contains 5 columns, where every column $\mathbf{q}_j$ corresponds to the output S_j of the rule-based system

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5] = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} & q_{1,4} & q_{1,5} \\ q_{2,1} & q_{2,2} & q_{2,3} & q_{2,4} & q_{2,5} \\ q_{3,1} & q_{3,2} & q_{3,3} & q_{3,4} & q_{3,5} \\ q_{4,1} & q_{4,2} & q_{4,3} & q_{4,4} & q_{4,5} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ q_{32,1} & q_{32,2} & q_{32,3} & q_{32,4} & q_{32,5} \end{bmatrix}.$$

According to Theorem 2.10 the matrix of crisp outputs $\mathbf{S}(\mathbf{z}) = [S_1, \ldots, S_5]$ can be computed by the formula $\mathbf{S}(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{Q}$, where the system generator $\mathbf{g}(\mathbf{z})$ is given by (5.66) and $\mathbf{Q}$ by (2.52). The fundamental matrix $\boldsymbol{\Omega}$ for $n = 5$ can be computed recursively according to (5.69). The matrix $\boldsymbol{\Theta}$ contains 5 columns. Each column is associated with the appropriate crisp system output S_j as follows

$$\mathbf{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4, \boldsymbol{\theta}_5] \in \mathbb{R}^{32\times 5},$$
$$\boldsymbol{\theta}_j = [\theta_{j,00000}, \theta_{j,10000}, \theta_{j,01000}, \ldots, \theta_{j,11111}]^T \in \mathbb{R}^{32}, \qquad j = 1, \ldots, 5, \quad (5.83)$$

where

$$\begin{aligned}\boldsymbol{\theta}_1^T = [&\frac{1}{\sigma L_s} u_{as}(t), -\gamma, \frac{1}{\sigma L_s}, 0, \frac{K}{T_r}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0, 0, 0, 0, 0, pK, 0, 0, 0, 0, 0, 0, 0],\end{aligned}$$

$$\begin{aligned}\boldsymbol{\theta}_2^T = [&\frac{1}{\sigma L_s} u_{bs}(t), 0, -\gamma, 0, 0, 0, 0, 0, \frac{K}{T_r}, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0, -pK, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],\end{aligned}$$

$$\begin{aligned}\boldsymbol{\theta}_3^T = [&0, \frac{M}{T_r}, 0, 0, -\frac{1}{T_r}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0, 0, 0, 0, 0, -p, 0, 0, 0, 0, 0, 0, 0],\end{aligned}$$

$$\begin{aligned}\boldsymbol{\theta}_4^T = [&0, 0, \frac{M}{T_r}, 0, 0, 0, 0, 0, -\frac{1}{T_r}, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],\end{aligned}$$

$$\begin{aligned}\boldsymbol{\theta}_5^T = [&-\frac{T_L}{J}, 0, 0, 0, 0, 0, \frac{pM}{JL_r}, 0, 0, -\frac{pM}{JL_r}, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].\end{aligned}$$

Finally, we give a simple procedure of how to compute the consequents of the fuzzy "If-then" rules as elements $q_{i,j}$ of the matrix $\mathbf{Q}$:

$$q_{i,j} = \mathbf{g}^T(\boldsymbol{\gamma}_i)\,\boldsymbol{\theta}_j^T, \qquad i = 1, \ldots, 32, \qquad j = 1, \ldots, 5.$$

According to the antecedents matrix (5.80) we have the first three rules as follows:

R_1 : If z_1 is N_1 and z_2 is N_2 and z_3 is N_3 and z_4 is N_4 and z_5 is N_5, then
$$\frac{di_{sa}}{dt} = \boldsymbol{\theta}_1^T \mathbf{g}(\boldsymbol{\gamma}_{00000}), \quad \frac{di_{sb}}{dt} = \boldsymbol{\theta}_2^T \mathbf{g}(\boldsymbol{\gamma}_{00000}), \quad \frac{d\psi_{ra}}{dt} = \boldsymbol{\theta}_3^T \mathbf{g}(\boldsymbol{\gamma}_{00000}),$$
$$\frac{d\psi_{rb}}{dt} = \boldsymbol{\theta}_4^T \mathbf{g}(\boldsymbol{\gamma}_{00000}), \quad \frac{d\omega}{dt} = \boldsymbol{\theta}_5^T \mathbf{g}(\boldsymbol{\gamma}_{00000}),$$

R_2 : If z_1 is P_1 and z_2 is N_2 and z_3 is N_3 and z_4 is N_4 and z_5 is N_5, then
$$\frac{di_{sa}}{dt} = \boldsymbol{\theta}_1^T \mathbf{g}(\boldsymbol{\gamma}_{10000}), \quad \frac{di_{sb}}{dt} = \boldsymbol{\theta}_2^T \mathbf{g}(\boldsymbol{\gamma}_{10000}), \quad \frac{d\psi_{ra}}{dt} = \boldsymbol{\theta}_3^T \mathbf{g}(\boldsymbol{\gamma}_{10000}),$$
$$\frac{d\psi_{rb}}{dt} = \boldsymbol{\theta}_4^T \mathbf{g}(\boldsymbol{\gamma}_{10000}), \quad \frac{d\omega}{dt} = \boldsymbol{\theta}_5^T \mathbf{g}(\boldsymbol{\gamma}_{10000}),$$

R_3 : If z_1 is N_1 and z_2 is P_2 and z_3 is N_3 and z_4 is N_4 and z_5 is N_5, then

$$\frac{di_{sa}}{dt} = \boldsymbol{\theta}_1^T \mathbf{g}\left(\boldsymbol{\gamma}_{01000}\right), \quad \frac{di_{sb}}{dt} = \boldsymbol{\theta}_2^T \mathbf{g}\left(\boldsymbol{\gamma}_{01000}\right), \quad \frac{d\psi_{ra}}{dt} = \boldsymbol{\theta}_3^T \mathbf{g}\left(\boldsymbol{\gamma}_{01000}\right),$$
$$\frac{d\psi_{rb}}{dt} = \boldsymbol{\theta}_4^T \mathbf{g}\left(\boldsymbol{\gamma}_{01000}\right), \quad \frac{d\omega}{dt} = \boldsymbol{\theta}_5^T \mathbf{g}\left(\boldsymbol{\gamma}_{01000}\right),$$

$$\vdots$$

and the last rule:

R_{32} : If z_1 is P_1 and z_2 is P_2 and z_3 is P_3 and z_4 is P_4 and z_5 is P_5, then

$$\frac{di_{sa}}{dt} = \boldsymbol{\theta}_1^T \mathbf{g}\left(\boldsymbol{\gamma}_{11111}\right), \quad \frac{di_{sb}}{dt} = \boldsymbol{\theta}_2^T \mathbf{g}\left(\boldsymbol{\gamma}_{11111}\right), \quad \frac{d\psi_{ra}}{dt} = \boldsymbol{\theta}_3^T \mathbf{g}\left(\boldsymbol{\gamma}_{11111}\right),$$
$$\frac{d\psi_{rb}}{dt} = \boldsymbol{\theta}_4^T \mathbf{g}\left(\boldsymbol{\gamma}_{11111}\right), \quad \frac{d\omega}{dt} = \boldsymbol{\theta}_5^T \mathbf{g}\left(\boldsymbol{\gamma}_{11111}\right).$$

The method of modeling the motor as a discrete-time system is the same.

5.3.3 Acclimatization Chamber Model

In the paper [77] a condition-sequence control circuit was described. It is based on Boolean algebra algorithms to solve the complex logical problem existing in temperature-humidity environmental control procedures. The control algorithm for a grafted seedling acclimatization chamber was implemented on programmable logic controller (PLC), where the values of both input and output process variables were from the binary set $\{0, 1\}$. The inputs are time-dependent state-variables produced by the sensors that deliver the binary state concerning indoor and outdoor temperature and humidity and the plant light. The control signals are binary as well, designed for the actuator subsystems, i.e. air conditioner (Y_1), free cooling (Y_2), humidifier (Y_3) and heater (Y_4). Based on Boolean logic methods and using a psychrometric chart of cooling, dehumidifying, heating and humidifying control processes in a grafted seedlings acclimatization chamber, the following control strategies were derived in [77]:

$$Y_1 = x_1 \cdot \overline{x_2} \cdot \left(\overline{x_3} + \overline{x_4}\right) \cdot \left(\overline{x_5} + \overline{x_6}\right) \cdot \left(\overline{x_7} + \overline{x_8}\right) \tag{5.84}$$

$$Y_2 = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \cdot \overline{x_5} \cdot \overline{x_6} \cdot \overline{x_7} \cdot \overline{x_8} \tag{5.85}$$

$$Y_3 = \left(\overline{x_1} + \overline{x_2}\right) \cdot \overline{x_3} \cdot x_4 \cdot \left(\overline{x_5} + \overline{x_6}\right) \cdot \left(\overline{x_7} + \overline{x_8}\right) \tag{5.86}$$

$$Y_4 = \overline{x_1} \cdot \left(\overline{x_7} + \overline{x_8}\right) \cdot \left(A + B + C\right), \tag{5.87}$$

where

$$A = \overline{x_4} \cdot \overline{x_6} \cdot \left(x_2 + x_3 \cdot \overline{x_9}\right), \quad B = x_2 \cdot \overline{x_3} \cdot \left(\overline{x_5} + \overline{x_6}\right), \quad C = x_3 \cdot \overline{x_4} \cdot \overline{x_5} \cdot x_6. \tag{5.88}$$

The operations "$\overline{(\cdot)}$", "$\cdot$" and "+" mean Boolean "*not*", "*and*" and "*or*", respectively.

Unlike the approach used in [77], where the appropriate conditions for the state variables have to be either *true* or *false*, ($x_k \in \{0,1\}$), we can say that the same conditions can be satisfied to some degree, which is the number from the interval $[0,1]$ (see [55], [56]). The decisions Y_k can be true to a certain degree, as well. Thus, we want to extend the control algorithm (5.84)-(5.88) into the fuzzy domain. To do this, according to the results from Section 5.1.3 we assume that

- the inputs $x_k \in \{0,1\}$ are replaced by $z_k \in [0,1]$ as the inputs for the P1-TS system, $(k = 1, \ldots, 9)$,
- the outputs of the P1-TS system are Y_1, Y_2, Y_3 and Y_4,
- the Boolean operations "$\overline{(.)}$", "$\cdot$" and "+" we replace by "n" - the strong negation (5.13), "$\mathfrak{t}$" - the probabilistic t-norm (5.16) and "$\mathfrak{s}$" - the probabilistic t-conorm (5.17), respectively.

Finally we express Boolean functions (5.84)-(5.88) by the following ones

$$\begin{aligned} Y_1 &= \mathfrak{t}\left(z_1, n\left(z_2\right), \mathfrak{s}\left(n\left(z_3\right), n\left(z_4\right)\right), \mathfrak{s}\left(n\left(z_5\right), n\left(z_6\right)\right), \mathfrak{s}\left(n\left(z_7\right), n\left(z_8\right)\right)\right) \\ &= z_1\left(1-z_2\right)\left(1-z_3 z_4\right)\left(1-z_5 z_6\right)\left(1-z_7 z_8\right), \end{aligned} \tag{5.89}$$

$$\begin{aligned} Y_2 &= \mathfrak{t}\left(n\left(z_1\right), n\left(z_2\right), n\left(z_3\right), n\left(z_4\right), n\left(z_5\right), n\left(z_6\right), n\left(z_7\right), n\left(z_8\right)\right) \\ &= \left(1-z_1\right)\left(1-z_2\right)\left(1-z_3\right)\left(1-z_4\right)\left(1-z_5\right)\left(1-z_6\right)\left(1-z_7\right)\left(1-z_8\right), \end{aligned} \tag{5.90}$$

$$\begin{aligned} Y_3 &= \mathfrak{t}\left(\mathfrak{s}\left(n\left(z_1\right), n\left(z_2\right)\right), n\left(z_3\right), z_4, \mathfrak{s}\left(n\left(z_5\right), n\left(z_6\right)\right), \mathfrak{s}\left(n\left(z_7\right), n\left(z_8\right)\right)\right) \\ &= \left(1-z_1 z_2\right)\left(1-z_3\right) z_4\left(1-z_5 z_6\right)\left(1-z_7 z_8\right), \end{aligned} \tag{5.91}$$

$$\begin{aligned} Y_4 &= \mathfrak{t}\left(n\left(z_1\right), \mathfrak{s}\left(n\left(z_7\right), n\left(z_8\right)\right), \mathfrak{s}\left(A, B, C\right)\right) \\ &= \left(1-z_1\right)\left(1-z_7 z_8\right)\left(A+B+C-AB-AC-BC+ABC\right), \end{aligned} \tag{5.92}$$

where

$$\begin{aligned} A &= \mathfrak{t}\left(n\left(z_4\right), n\left(z_6\right), \mathfrak{s}\left(z_2, \mathfrak{t}\left(z_3, n\left(z_9\right)\right)\right)\right) \\ &= \left(1-z_2\right)\left(z_2-z_3\left(1-z_9\right)\right)\left(1-z_4\right)\left(1-z_6\right), \end{aligned} \tag{5.93}$$

$$B = \mathfrak{t}\left(z_2, n\left(z_3\right), \mathfrak{s}\left(n\left(z_5\right), n\left(z_6\right)\right)\right) = z_2\left(1-z_3\right)\left(1-z_5 z_6\right), \tag{5.94}$$

$$C = \mathfrak{t}\left(z_3, n\left(z_4\right), n\left(z_5\right), z_6\right) = z_3\left(1-z_4\right)\left(1-z_5\right) z_6. \tag{5.95}$$

If we apply binary sensors, the inputs are $z_k \in \{0,1\}$ and the P1-TS system works exactly as a binary controller which produces the outputs $Y_1, \ldots, Y_4 \in \{0,1\}$. By replacing the sensors into the input interfaces containing analog or k-bits digital sensors, $(k \geq 2)$ delivering the signals $z_k \in [0,1]$, the controller works smoothly as a multidimensional "fuzzy switching system". Since for the P1-TS system the conclusions of the rules are from the set $\{0,1\}$, according to the assessment (5.12), the outputs Y_j are from the interval $[0,1]$ for $j = 1,2,3,4$. Therefore in the case of the fuzzy switching system, the plant should be equipped with actuators handling analog signals.

The fuzzy metarules can be easily obtained. For example the equations (5.89)-(5.91) generate the following metarules

M_1 : If z_1 is P_1 and z_2 is N_2 and (z_3 is N_3 or z_4 is N_4) and (z_5 is N_5 or z_6 is N_6) and (z_7 is N_7 or z_8 is N_8), then Y_1 is 1, otherwise Y_1 is 0.

M_2 : If z_1 is N_1 and z_2 is N_2 and z_3 is N_3 and z_4 is N_4 and z_5 is N_5 and z_6 is N_6 and z_7 is N_7 and z_8 is N_8, then Y_2 is 1, otherwise Y_2 is 0.

M_3 : If (z_1 is N_1 or z_2 is N_2) and z_3 is N_3 and z_4 is P_4 and z_5 is N_5 and (z_5 is N_5 or z_6 is N_6) and (z_7 is N_7 or z_8 is N_8), then Y_3 is 1, otherwise Y_3 is 0.

The membership functions of fuzzy sets $N_k(z_k)$ and $P_k(z_k)$ have a clear interpretation. For example, if formerly assumed $x_9 = 1$ (or $x_9 = 0$) modeled the situation “the light is on” (or “the light is off”), then for the P1-TS system “z_9 is P_9” (or “z_9 is N_9”) models the situation “the plant light is powerful” (or “the plant light is not powerful”).

5.4 Optimal Fuzzy Control System Design for Second Order Plant

In this section we will consider a PID control system that behaves optimally in some sense. First, we obtain a highly interpretable P1-TS system, exactly modeling conventional PID controller. Next, we will show how to obtain a fuzzy PID controller as the rule-based system satisfying typical engineering requirements formulated for the closed-loop control system.

5.4.1 Highly Interpretable Fuzzy Rules for PID Controller

It is well known that a conventional PID controller is a special kind of the so called PID fuzzy controller (PID-FC for short [95]). Assume that the control action at the instant t is $u(t)$. By $\varepsilon(t)$ we denote the control error, as the difference between the reference signal $w(t)$ and the plant output $y(t)$, as shown in Fig. 5.12:

$$\varepsilon(t) = w(t) - y(t). \tag{5.96}$$

The rule-based P1-TS system has three inputs: $z_1 = \dot{\varepsilon}(t)$, $z_2 = \varepsilon(t)$ and $z_3 = \ddot{\varepsilon}(t)$, where $\dot{\varepsilon}(t)$ is the *speed* of the error and $\ddot{\varepsilon}(t)$ - its *acceleration* [92]. The P1-TS system output S is the derivative of the control action $\dot{u}(t)$. It is

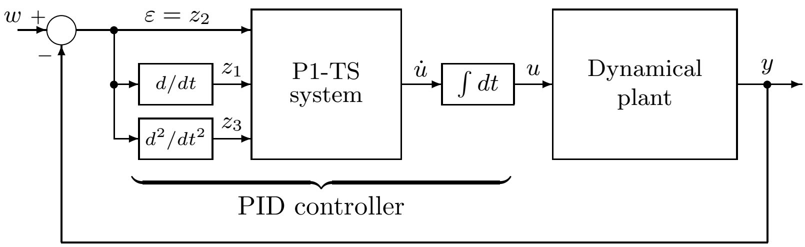


Fig. 5.12 Closed-loop PID control system

apparent that both the error and its derivatives are from intervals containing zero. Therefore we make the next assumption

$$z_k \in [-\alpha_k, \beta_k], \qquad k = 1, 2, 3,$$

where

$$\alpha_1 = \beta_1 = \sup_{t \geq 0} |\dot{\varepsilon}(t)|, \quad \alpha_2 = \beta_2 = \sup_{t \geq 0} |\varepsilon(t)|, \quad \alpha_3 = \beta_3 = \sup_{t \geq 0} |\ddot{\varepsilon}(t)|. \tag{5.97}$$

For the stable closed-lop control system, the cuboid D^3 includes all trajectories $(\dot{\varepsilon}(t), \varepsilon(t), \ddot{\varepsilon}(t))$ for $t \geq 0$. The fuzzy sets for the input variables z_k that are used in the antecedents of the fuzzy rules are N_k and P_k for $k = 1, 2, 3$. The label N_k denotes a *negative* value of kth input variable and P_k - a *positive* one (see Fig. 2.8 in Section 2.2).

Suppose that at every moment $t \geq 0$, the P1-TS system output is the following linear combination of the inputs

$$S = k_p z_1 + T_i^{-1} z_2 + T_d z_3.$$

It follows that

$$u(t) = k_p \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) \, d\tau + T_d \frac{d\varepsilon}{dt} + u(0). \tag{5.98}$$

This means that the P1-TS system together with the integral block produces the same signal as a conventional, continuous-time proportional-integral-derivative (PID) controller. In order to obtain the fuzzy rules for the P1-TS system, we define the following vector of coefficients

$$\boldsymbol{\theta} = [0, k_p, 1/T_i, 0, T_d, 0, 0, 0]^T. \tag{5.99}$$

From (2.30), (2.41) and (5.99) we immediately obtain the consequents of the fuzzy rules

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & -\alpha_3 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & -\alpha_3 & -\beta_1\alpha_3 & \alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & -\alpha_3 & \alpha_1\alpha_3 & -\alpha_3\beta_2 & \alpha_1\alpha_3\beta_2 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & -\alpha_3 & -\beta_1\alpha_3 & -\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & \beta_3 & -\alpha_1\beta_3 & -\alpha_2\beta_3 & \alpha_1\alpha_2\beta_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & \beta_3 & \beta_1\beta_3 & -\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & \beta_3 & -\alpha_1\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_2\beta_3 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & \beta_3 & \beta_1\beta_3 & \beta_2\beta_3 & \beta_1\beta_2\beta_3 \end{bmatrix} \begin{bmatrix} 0 \\ k_p \\ 1/T_i \\ 0 \\ T_d \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 \\ \beta_1 & -\alpha_2 & -\alpha_3 \\ -\alpha_1 & \beta_2 & -\alpha_3 \\ \beta_1 & \beta_2 & -\alpha_3 \\ -\alpha_1 & -\alpha_2 & \beta_3 \\ \beta_1 & -\alpha_2 & \beta_3 \\ -\alpha_1 & \beta_2 & \beta_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix} \begin{bmatrix} k_p \\ 1/T_i \\ T_d \end{bmatrix}. \tag{5.100}$$

Thus, every consequent of the vth fuzzy rule ($v = 1, \ldots, 8$) is a scalar product of the vector containing adjustable parameters $(k_p, 1/T_i, T_d)$ and vth vertex $\boldsymbol{\gamma}_v$ of the cuboid D^3. In other words we obtain the system of fuzzy rules for the P1-TS system, that defines a differentiated output of the conventional PID controller, which consists of 8 fuzzy rules given in Table 5.9, where α_k and β_k are given by (5.97). As one can see, the fuzzy rules are extremely simple for interpretation.

5.4.2 Optimal PID Fuzzy Controller for Linear Second Order Plant

Let us consider the control system shown in Fig. 5.12, containing the dynamical oscillatory plant

$$\frac{d^2 y(t)}{dt^2} + 2\xi\omega_0 \frac{dy(t)}{dt} + \omega_0^2 y(t) = k_0 u(t), \qquad y(0) = y_0,\ \dot{y}(0) = \dot{y}_0, \tag{5.101}$$

Table 5.9 Look-up-table for the P1-TS fuzzy system which produces the derivative of conventional PID controller output

$\dot{\varepsilon}(t), \varepsilon(t) \setminus \ddot{\varepsilon}(t) \rightarrow$ ↓	N_3	P_3
$N_1 N_2$	$-k_p\alpha_1 - T_i^{-1}\alpha_2 - T_d\alpha_3$	$-k_p\alpha_1 - T_i^{-1}\alpha_2 + T_d\beta_3$
$N_1 P_2$	$-k_p\alpha_1 + T_i^{-1}\beta_2 - T_d\alpha_3$	$-k_p\alpha_1 + T_i^{-1}\beta_2 + T_d\beta_3$
$P_1 P_2$	$k_p\beta_1 + T_i^{-1}\beta_2 - T_d\alpha_3$	$k_p\beta_1 + T_i^{-1}\beta_2 + T_d\beta_3$
$P_1 N_2$	$k_p\beta_1 - T_i^{-1}\alpha_2 - T_d\alpha_3$	$k_p\beta_1 - T_i^{-1}\alpha_2 + T_d\beta_3$

where k_0, ξ and ω_0 are known parameters, with $k_0 > 0$, $0 < \xi < 1$ and $\omega_0 > 0$. The reference signal is Heaviside step function

$$w(t) = \begin{cases} w_0 & \text{for } t > 0 \\ 0 & \text{for } t \le 0 \end{cases}. \tag{5.102}$$

Our goal is to design such a rule-based PID controller that guarantees a nonoscillatory step response

$$y(t) = w_0 + (y_0 - w_0)\, e^{-t\lambda}, \tag{5.103}$$

of the closed-loop system with zero initial conditions, where $\lambda > 0$ is the preset parameter. Thus, in the feedback system, the plant output should be governed by the differential equation

$$\frac{dy(t)}{dt} + \lambda y(t) = \lambda w(t), \qquad y(0) = y_0.$$

The control error is defined as $\varepsilon(t) = w(t) - y(t)$. The step response (5.103) has three important features:

- there are no oscillations in the closed-loop system,
- there is no steady state error ($\varepsilon(\infty) = 0$), and
- the plant output y reaches the set-point w_0 as quickly as required (by choosing a sufficiently large parameter λ).

In such sense the controller that guarantees the step response (5.103) is considered to be optimal.

We require that at any instant t in the closed-loop system, the following relation between the system output y and the control error ε

$$y(t) = \lambda \int_0^t \varepsilon(\tau)\, d\tau + y_0, \tag{5.104}$$

must be satisfied. One can check that by zero initial conditions, the equation (5.104) holds, if

$$\lambda \frac{d\varepsilon(t)}{dt} + 2\xi\omega_0\lambda\varepsilon(t) + \omega_0^2\lambda \int_0^t \varepsilon(\tau)\, d\tau = k_0 k_p \varepsilon(t) + \frac{k_0}{T_i}\int_0^t \varepsilon(\tau)\, d\tau + k_0 T_d \frac{d\varepsilon(t)}{dt}$$

is satisfied. From the last equation we obtain optimal - in the sense expressed above, the gain parameters for the PID controller

$$\begin{bmatrix} k_p^* \\ 1/T_i^* \\ T_d^* \end{bmatrix} = \begin{bmatrix} 2\lambda\xi\omega_0/k_0 \\ \lambda\omega_0^2/k_0 \\ \lambda/k_0 \end{bmatrix}. \tag{5.105}$$

According to Table 5.9, the system of highly interpretable fuzzy rules for the P1-TS system that defines the optimal PID controller (5.98), consists of the following fuzzy rules:

R_1 : If $\dot{\varepsilon}$ is negative and ε is negative and $\ddot{\varepsilon}$ is negative,
then $\dot{u} = -2\alpha_1\lambda\xi\omega_0/k_0 - \alpha_2\lambda\omega_0^2/k_0 - \alpha_3\lambda/k_0$,

R_2 : If $\dot{\varepsilon}$ is positive and ε is negative and $\ddot{\varepsilon}$ is negative,
then $\dot{u} = 2\beta_1\lambda\xi\omega_0/k_0 - \alpha_2\lambda\omega_0^2/k_0 - \alpha_3\lambda/k_0$,

R_3 : If $\dot{\varepsilon}$ is negative and ε is positive and $\ddot{\varepsilon}$ is negative,
then $\dot{u} = -2\alpha_1\lambda\xi\omega_0/k_0 + \beta_2\lambda\omega_0^2/k_0 - \alpha_3\lambda/k_0$,

R_4 : If $\dot{\varepsilon}$ is positive and ε is positive and $\ddot{\varepsilon}$ is negative,
then $\dot{u} = 2\beta_1\lambda\xi\omega_0/k_0 + \beta_2\lambda\omega_0^2/k_0 - \alpha_3\lambda/k_0$,

R_5 : If $\dot{\varepsilon}$ is negative and ε is negative and $\ddot{\varepsilon}$ is positive,
then $\dot{u} = -2\alpha_1\lambda\xi\omega_0/k_0 - \alpha_2\lambda\omega_0^2/k_0 + \beta_3\lambda/k_0$,

R_6 : If $\dot{\varepsilon}$ is positive and ε is negative and $\ddot{\varepsilon}$ is positive,
then $\dot{u} = 2\beta_1\lambda\xi\omega_0/k_0 - \alpha_2\lambda\omega_0^2/k_0 + \beta_3\lambda/k_0$,

R_7 : If $\dot{\varepsilon}$ is negative and ε is positive and $\ddot{\varepsilon}$ is positive,
then $\dot{u} = -2\alpha_1\lambda\xi\omega_0/k_0 + \beta_2\lambda\omega_0^2/k_0 + \beta_3\lambda/k_0$,

R_8 : If $\dot{\varepsilon}$ is positive and ε is positive and $\ddot{\varepsilon}$ is positive,
then $\dot{u} = 2\beta_1\lambda\xi\omega_0/k_0 + \beta_2\lambda\omega_0^2/k_0 + \beta_3\lambda/k_0$,

Thus, the conclusions of the optimal fuzzy rules depend on maximal values of the control error, its speed and the acceleration, the values α_k and β_k given by (5.97), the plant parameters k_0, ξ and ω_0, and one design parameter λ.

The above result demonstrates the ability to design an optimal fuzzy logic controller, which satisfies typical engineering requirements when controlling a linear second order plant. The resulting closed-loop system is free of oscillations, has no steady state error and its step response is as quick as required.

5.4.3 PD-Like Optimal Controller for Nonlinear Second Order Plant

Let us consider the closed-loop system shown in Fig. 5.13 that contains the following nonlinear dynamical plant

$$\ddot{y}(t) + a\dot{y}(t)\,y(t) + b\dot{y}(t) + cy(t) = k_0u(t), \qquad y(0) = y_0,\ \dot{y}(0) = \dot{y}_0, \tag{5.106}$$

where $a > 0$, $b > 0$, $c \geq 0$ and $k_0 > 0$ are known constants.

We assume that the P1-TS system functions as a controller and produces the output

$$u(t) = k_1\varepsilon(t) + k_2\dot{\varepsilon}(t) + k_3\varepsilon(t)\,\dot{\varepsilon}(t), \tag{5.107}$$

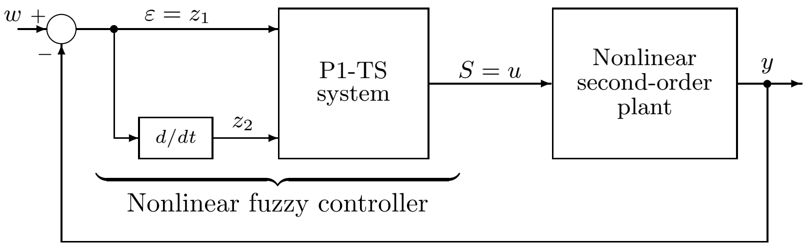


Fig. 5.13 Closed-loop nonlinear PD-like fuzzy control system

where k_1, k_2 and k_3 are the controller gains, $\varepsilon(t) = w(t) - y(t)$ is the control error, $\dot{\varepsilon}(t)$ is its derivative, and $w(t)$ is at least twice differentiable step-like reference signal, which will be defined further on. Our goal is to design such a rule-based controller for which

- the closed-loop system is free of oscillations,
- the response of the feedback system for the step-like reference signal is sufficiently quick, and
- the steady state error is sufficiently small.

If the above conditions are satisfied, we will call the controller an optimal one. We show how to design such a controller in the form of a P1-TS system.

According to (5.106) the control error in the closed-loop system satisfies the following nonlinear, nonstationary differential equation

$$\ddot{\varepsilon}(t) + (k_0 k_3 - a)\,\dot{\varepsilon}(t)\,\varepsilon(t) + 2p(t)\,\dot{\varepsilon}(t) + r(t)\,\varepsilon(t) = f(t)\,, \tag{5.108}$$

where

$$f(t) = \ddot{w}(t) + a\dot{w}(t)\,w(t) + b\dot{w}(t) + cw(t)\,, \tag{5.109}$$

$$2p(t) = b + k_0 k_2 + aw(t)\,, \tag{5.110}$$

$$r(t) = c + k_0 k_1 + a\dot{w}(t)\,. \tag{5.111}$$

Let us choose the controller gains

$$k_3 = \frac{a}{k_0}, \tag{5.112}$$

$$k_2 = k_2(t) = \frac{-aw(t) - b + 2p_0}{k_0}, \tag{5.113}$$

$$k_1 = k_1(t) = \frac{-a\dot{w}(t) - c + p_0^2}{k_0}, \tag{5.114}$$

where k_3 is a constant, k_2 and k_1 vary in time and $p_0 > 0$ is a new design parameter. In such case the error $\varepsilon(t)$ satisfies the following linear stationary differential equation

$$\ddot{\varepsilon}(t) + 2p_0\dot{\varepsilon}(t) + p_0^2\varepsilon(t) = f(t), \qquad \varepsilon(0) = \varepsilon_0, \quad \dot{\varepsilon}(0) = \dot{\varepsilon}_0. \tag{5.115}$$

Thus, the P1-TS system ensures linearization of the nonlinear plant and the control error is of the form

$$\varepsilon(t) = (1 + p_0 t) e^{-p_0 t}\varepsilon_0 + te^{-p_0 t}\dot{\varepsilon}_0 + \int_0^t g_0(t-\tau) f(\tau) d\tau, \tag{5.116}$$

where $f(t)$ is an external signal (5.109) and $g_0(t) = te^{-p_0 t}$. For the step-like functions $w(t)$, the steady-state error $\varepsilon(\infty)$ depends on the parameter p_0 and can be made arbitrarily small. To show this, let us take an example of the reference signal

$$w(t) = w_0 - w_0\left(1 + mt + \frac{m^2t^2}{2}\right)e^{-mt}, \qquad m > 0. \tag{5.117}$$

where $m > 0$. With a sufficiently large m, the reference signal $w(t)$ approximates the step function (5.102), as needed. According to (5.117) and (5.109) we have

$$\begin{aligned} f(t) &= cw_0 - \frac{1}{4}w_0^2t^2m^3a\left(m^2t^2 + 2mt + 2\right)e^{-2mt} \\ &\quad + w_0\left(\frac{1}{2}m^2\left(bm - c + amw_0 - m^2\right)t^2 + m\left(m^2 - c\right)t - c\right)e^{-mt}. \end{aligned}$$

For $w(t)$ given by (5.117), the solution (5.116) is of the form

$$\begin{aligned} \varepsilon(t) &= \frac{cw_0}{p_0^2} + (h_1 + h_2t)e^{-p_0t} + \left(h_3 + h_4t + h_5t^2\right)e^{-mt} \\ &\quad + \left(h_6 + h_7t + h_8t^2 + h_9t^3 + h_{10}t^4\right)e^{-2mt}, \end{aligned} \tag{5.118}$$

where the coefficients $h_1, \ldots, h_{10}$ depend on initial conditions ε_0 and $\dot{\varepsilon}_0$ and the constants a, b, c, m, w_0 and p_0. Thus, the error $\varepsilon(t)$ vanishes without oscillations to the steady-state value

$$\varepsilon(\infty) = \frac{cw_0}{p_0^2}, \tag{5.119}$$

and its decay rate depends only on the given parameter m and a free design parameter p_0. Thus, we can choose the appropriate controller gains k_i that guarantee the formerly formulated requirements for the feedback system.

If the value of the steady-state error is the most important requirement and the quotient

$$\eta = \frac{\varepsilon(\infty)}{w_0}$$

is given, we propose first to choose the parameter

$$p_0 = \sqrt{\frac{c}{\eta}}\,, \tag{5.120}$$

and next, the controller gains k_1, k_2 and k_3, according to (5.112)-(5.114).

Now we show how to build the P1-TS system as a controller that satisfies the above formulated requirements. The input signals of the P1-TS system are $z_1(t) = \varepsilon(t) \in [-\alpha_1, \beta_1]$ and $z_2(t) = \dot{\varepsilon}(t) \in [-\alpha_2, \beta_2]$, where

$$\alpha_1 = \beta_1 = \sup_{t \geq 0} |\varepsilon(t)|\,, \qquad \alpha_2 = \beta_2 = \sup_{t \geq 0} |\dot{\varepsilon}(t)|\,, \tag{5.121}$$

by $\varepsilon(t)$ as in (5.118). Thus, the linear membership functions of fuzzy sets are $N_k(z_k)$ and $P_k(z_k)$, $(k = 1, 2)$, where:

- N_1 and P_1 denote *negative* and *positive* control error,
- N_2 and P_2 - *negative* and *positive* speed of the control error, respectively.

According to (5.107), the output of the rule-based system is $u(t) = \boldsymbol{\theta}^T \mathbf{g}(z_1, z_2)$, where $\boldsymbol{\theta} = [0, k_1, k_2, k_3]^T$ and system generator $\mathbf{g}$ is given by (2.37). The consequents of the fuzzy rules constitute the vector $[q_1, q_2, q_3, q_4]^T$. For the fundamental matrix of the system (2.38), from (2.30) we obtain the consequents of the fuzzy rules

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \boldsymbol{\Omega}^T \boldsymbol{\theta} = \begin{bmatrix} -k_1\alpha_1 - k_2\alpha_2 + k_3\alpha_1\alpha_2 \\ k_1\beta_1 - k_2\alpha_2 - k_3\alpha_2\beta_1 \\ -k_1\alpha_1 + k_2\beta_2 - k_3\alpha_1\beta_2 \\ k_1\beta_1 + k_2\beta_2 + k_3\beta_1\beta_2 \end{bmatrix}.$$

According to (5.113)-(5.114) and (5.120) we obtain

$$k_1(t) = \frac{1}{k_0}\left(-a\dot{w}(t) - c + \frac{c}{\eta}\right), \tag{5.122}$$

$$k_2(t) = \frac{1}{k_0}\left(-aw(t) - b + 2\sqrt{\frac{c}{\eta}}\right)\,. \tag{5.123}$$

Finally, the system of highly interpretable fuzzy rules for the optimal P1-TS system as a controller for the nonlinear plant (5.106) by the reference input $w(t)$ is as follows

R_1 : If the control error is *negative* and its speed is *negative*,
then the control action $u(t) = -k_1(t)\,\alpha_1 - k_2(t)\,\alpha_2 + k_3\alpha_1\alpha_2$,

R_2 : If the control error is *positive* and its speed is *negative*,
then the control action $u(t) = k_1(t)\beta_1 - k_2(t)\alpha_2 - k_3\alpha_2\beta_1$,

R_3 : If the control error is *negative* and its speed is *positive*,
then the control action $u(t) = -k_1(t)\alpha_1 + k_2(t)\beta_2 - k_3\alpha_1\beta_2$,

R_4 : If the control error is *positive* and its speed is *positive*,
then the control action $u(t) = k_1(t)\beta_1 + k_2(t)\beta_2 + k_3\beta_1\beta_2$.

Below we exemplify numerically the design procedure, taking into account the above and simplified consequents of the rules.

Example 5.12. Suppose the control design requirement for the steady-state error in the closed-loop system is $\eta = 0.04$. The nonlinear plant is described by

$$\ddot{y}(t) + 2\dot{y}(t)y(t) + 3\dot{y}(t) + y(t) = u(t)$$

and the reference signal $w(t) = 1 - \left(1 + 10t + 50t^2\right)e^{-10t}$. Thus, $a = 2$, $b = 3$, $c = 1$, $k_0 = 1$, $w_0 = 1$ and $m = 10$. We assume that all trajectories of the system are contained in the rectangle

$$(\varepsilon(t), \dot{\varepsilon}(t)) \in D^2 = [-5, 5] \times [-15, 15].$$

Our task is to obtain the fuzzy control rules for an optimal P1-TS system.

From (5.112) we obtain $k_3 = 2$. According to (5.122)-(5.123), the nonstationary gains are

$$k_1(t) = 24 - 1000t^2e^{-10t},$$

$$k_2(t) = 5 + 2e^{-10t}\left(10t + 50t^2 + 1\right).$$

Thus, the optimal fuzzy rules for the P1-TS system are as follows (see Fig. 5.14 and 5.15):

R_1 : If the control error is *negative* and its speed is *negative*, then
$u(t) = u_1(t) = 5000t^2e^{-10t} - 30e^{-10t}\left(10t + 50t^2 + 1\right) - 45$,

R_2 : If the control error is *positive* and its speed is *negative*, then
$u(t) = u_2(t) = -5000t^2e^{-10t} - 30e^{-10t}\left(10t + 50t^2 + 1\right) - 105$,

R_3 : If the control error is *negative* and its speed is *positive*, then
$u(t) = u_3(t) = 5000t^2e^{-10t} + 30e^{-10t}\left(10t + 50t^2 + 1\right) - 195$,

R_4 : If the control error is *positive* and its speed is *positive*, then
$u(t) = u_4(t) = 30e^{-10t}\left(10t + 50t^2 + 1\right) - 5000t^2e^{-10t} + 345$.

Observe that the control signal $u(t)$ is positive for $t \geq 0$, if both the control error, and its speed are positive; otherwise $u(t) < 0$. Although the main design goal was achieved, we can try to take much simpler, i.e. constant conclusions of the rules, since the reference input w in the steady-state is constant. In other words we substitute the conclusion u_v of the rule R_v for

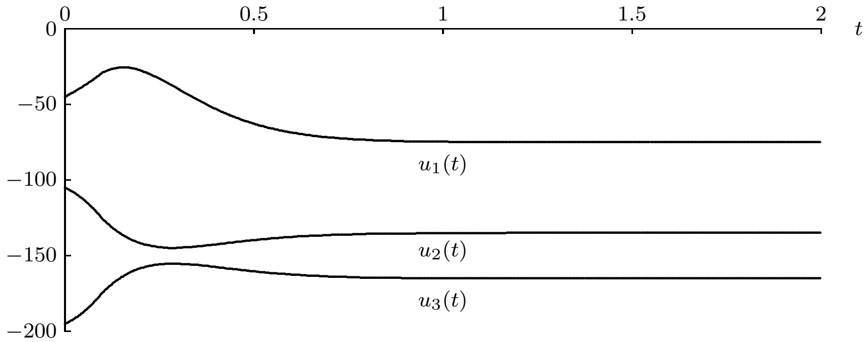


Fig. 5.14 Consequents of the rules R_1-R_3 from Example 5.12

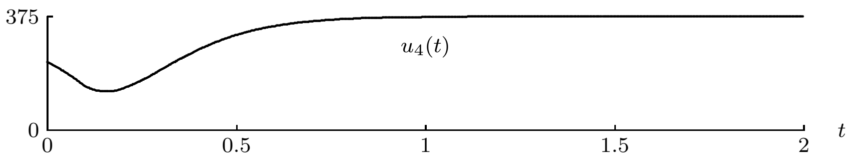


Fig. 5.15 Consequent of the rule R_4 from Example 5.12

$\lim_{t\to\infty} u_v(t)$ for $v = 1, 2, 3, 4$. The resulting quasi-optimal fuzzy rules are as follows:

R_1' : If the control error is *negative* and its speed is *negative*, then $u = -45$,
R_2' : If the control error is *positive* and its speed is *negative*, then $u = -105$,
R_3' : If the control error is *negative* and its speed is *positive*, then $u = -195$,
R_4' : If the control error is *positive* and its speed is *positive*, then $u = 345$.

Fig. 5.16 shows the error plots for initial conditions $(\varepsilon_0, \dot{\varepsilon}_0) = (4, 5) \in D^2$ for two systems of fuzzy rules: R_1 - R_4 and R_1' - R_4'. As one can see, in order to obtain the required closed-loop system behavior, we can simply choose the constant coefficients. The fuzzy rules obtained in this way are quasi-optimal but simple.

5.5 P1-TS System as Controller with Variable Gains

In this section we prove the next useful fact.

Corollary 5.13. *Consider two functions: $f_1 : D^n \to \mathbb{R}$ and the polynomial f_0 given by (2.26) with known coefficients $\theta_{p_1,\ldots,p_n}$, $(p_1, \ldots, p_n) \in \{0, 1\}^n$.*

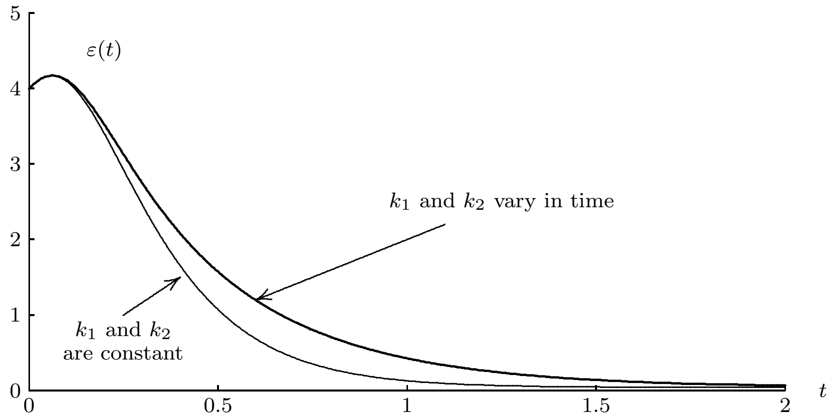


Fig. 5.16 Plots of the control error for the nonstationary and constant consequents of the fuzzy rules considered in Example 5.12

Suppose the output of a P1-TS system is S, its inputs are $z_1, \ldots, z_n$ and this system is defined by 2^n rules (2.13)-(2.15), in which the consequents of the rules are $q_v = k_v f_1(\mathbf{z})$ for $k_v \in \mathbb{R}$, $v \in \{1, 2, \ldots, 2^n\}$ as in (2.16). Then

- *The P1-TS system is equivalent to the function $f_0(\mathbf{z}) \cdot f_1(\mathbf{z})$ for all $\mathbf{z} \in D^n$. For the given function $f_1(\mathbf{z})$ and coefficients k_v, one can find all consequents of the fuzzy rules, according to (2.47), i.e. $\mathbf{q} = [f_0(\boldsymbol{\gamma}_1), \ldots, f_0(\boldsymbol{\gamma}_{2^n})]^T$.*
- *For the given vector q of the consequents of the fuzzy rules, the function to which the P1-TS system is equivalent, is given by*

$$S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{k} f_1(\mathbf{z}), \tag{5.124}$$

where $\mathbf{g}(\mathbf{z})$ is the generator (2.28), $\boldsymbol{\Omega}$ is the fundamental matrix (2.30), and the coefficients k_v constitute the vector $\mathbf{k} = [k_1, k_2, \ldots, k_{2^n}]^T$.

Proof. According to (2.21), the output of the P1-TS system is $S(\mathbf{z}) = \sum_{v=1}^{2^n} h_v q_v$, where $q_v = k_v f_1(\mathbf{z})$. Thus, there exists a collection of real coefficients a_{i_k} and b_{i_k} for $i_k \in \{1, 2\}$ and $k = 1, \ldots, n$ such, that

$$S = \sum_{i_n=1}^{2} \cdots \sum_{i_1=1}^{2} \prod_{k=1}^{n} \left(a_{i_k} z_k + b_{i_k}\right) k_{(i_1, \ldots, i_n)} f_1(\mathbf{z}) = f_0(\mathbf{z}) \cdot f_1(\mathbf{z}),$$

where $f_0(\mathbf{z})$ is the same function as in (2.26). We omit the rest of the proof, since from this point it is practically the same as the proof of Theorem 2.4. □

Example 5.14. Consider the P1-TS system being a special case of the one of Corollary 5.13, in which $f_1(\mathbf{z}) = \mathbf{a}^T \mathbf{z}$ with $\mathbf{a} \in \mathbb{R}^n$ by $n = 3$. Such a system was considered in [205] and [206]. The set of 8 fuzzy rules is as follows

R_1 : If z_1 is N_1 and z_2 is N_2 and z_3 is N_3, then $S = k_8 c$,
R_2 : If z_1 is N_1 and z_2 is N_2 and z_3 is P_3, then $S = k_7 c$,
R_3 : If z_1 is N_1 and z_2 is P_2 and z_3 is N_3, then $S = k_6 c$,
R_4 : If z_1 is N_1 and z_2 is P_2 and z_3 is P_3, then $S = k_5 c$,
R_5 : If z_1 is P_1 and z_2 is N_2 and z_3 is N_3, then $S = k_4 c$,
R_6 : If z_1 is P_1 and z_2 is N_2 and z_3 is P_3, then $S = k_3 c$,
R_7 : If z_1 is P_1 and z_2 is P_2 and z_3 is N_3, then $S = k_2 c$,
R_8 : If z_1 is P_1 and z_2 is P_2 and z_3 is P_3, then $S = c = k_1 \mathbf{a}^T \mathbf{z}$,

where the inputs of the P1-TS system are $z_k \in [-\alpha_k, \beta_k]$, $k = 1, 2, 3$, the output is S and the coefficients k_i are constant for $i = 1, \ldots, 8$. The fuzzy system performs a function of the controller in the closed-loop. This can be easily achieved by the appropriate interpretation of the inputs z_k: they are state variables at instant t, and the output S is the control action (or its derivative, as in Section 5.4.1). Such a system was developed by Ying [206] in another way and was called the *controller with variable gains.* According to (5.124) we obtain

$$S(\mathbf{z}) = \underbrace{\mathbf{g}^T(\mathbf{z})\left(\mathbf{\Omega}^T\right)^{-1}\mathbf{h}k_1}_{w(\mathbf{z})}\mathbf{a}^T\mathbf{z} = w(\mathbf{z})\,\mathbf{a}^T\mathbf{z},$$

where the constant vectors are $\mathbf{h} = [k_8, k_7, k_6, k_5, k_4, k_3, k_2, 1]^T$ and $\mathbf{a}^T = [a_1, a_2, a_3]$. The scalar function $w(\mathbf{z}) = \mathbf{g}^T(\mathbf{z})\left(\mathbf{\Omega}^T\right)^{-1}\mathbf{h}k_1$ determines *variable gains* of the controller. Its role is clear, since $S(\mathbf{z}) = w(\mathbf{z})\,\mathbf{a}^T\mathbf{z}$. The generator $\mathbf{g}(\mathbf{z})$ is known, and therefore we are interested in computing the constant vector $\mathbf{b} = [b_1, b_2, \ldots, b_8]^T = \left(\mathbf{\Omega}^T\right)^{-1}\mathbf{h}k_1$ so that $w(\mathbf{z}) = \mathbf{g}^T(\mathbf{z})\,\mathbf{b}$ holds. After computations we obtain a general form of all components of the vector $\mathbf{b}$ as follows

$$b_1 = \frac{1 + k_2\lambda_3 + k_3\lambda_2 + k_4\lambda_2\lambda_3 + k_5\lambda_1 + k_6\lambda_1\lambda_3 + k_7\lambda_1\lambda_2 + k_8\lambda_1\lambda_2\lambda_3}{(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_2 = \frac{1 + k_2\lambda_3 + k_3\lambda_2 + k_4\lambda_2\lambda_3 - k_5 - k_6\lambda_3 - k_7\lambda_2 - k_8\lambda_2\lambda_3}{\alpha_1(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_3 = \frac{1 + k_2\lambda_3 - k_3 + k_5\lambda_1 - k_4\lambda_3 - k_7\lambda_1 + k_6\lambda_1\lambda_3 - k_8\lambda_1\lambda_3}{\alpha_2(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_4 = \frac{1 - k_2 + k_3\lambda_2 - k_4\lambda_2 + k_5\lambda_1 - k_6\lambda_1 + k_7\lambda_1\lambda_2 - k_8\lambda_1\lambda_2}{\alpha_3(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_5 = \frac{1 + k_2\lambda_3 - k_3 - k_4\lambda_3 - k_5 - k_6\lambda_3 + k_7 + k_8\lambda_3}{\alpha_1\alpha_2(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_6 = \frac{1 - k_2 + k_3\lambda_2 - k_4\lambda_2 - k_5 + k_6 - k_7\lambda_2 + k_8\lambda_2}{\alpha_1\alpha_3(1+\lambda_1)(1+\lambda_2)(1+\lambda_3)}k_1,$$

$$b_7 = \frac{1 - k_2 - k_3 + k_4 + k_5\lambda_1 - k_6\lambda_1 - k_7\lambda_1 + k_8\lambda_1}{\alpha_2\alpha_3\,(1+\lambda_1)\,(1+\lambda_2)\,(1+\lambda_3)} k_1,$$

$$b_8 = \frac{1 - k_2 - k_3 + k_4 - k_5 + k_6 + k_7 - k_8}{\alpha_1\alpha_2\alpha_3\,(1+\lambda_1)\,(1+\lambda_2)\,(1+\lambda_3)} k_1,$$

where $\lambda_k = \beta_k/\alpha_k$ for $k = 1,2,3$. One can check that in the special case, when the constants are $k_1 = 1$, $\alpha_k = \beta_k = L$ for $k = 1,2,3$, we obtain $\lambda_1 = \lambda_2 = \lambda_3 = 1$, i.e. the same outcome as in [206].

5.6 Exact Modeling of Single-Input Dynamical Systems

Our goal in this section is to model a single-input nonlinear dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t))$, where $\mathbf{x}$ is the state vector, u is the scalar control input, ($\mathbf{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$) and $\mathbf{f}$ is the vector function containing multivariate polynomials of the state vector components $x_i = x_i(t)$ and $u = u(t)$. We want to express equivalently the differential equations in the form of the P1-TS system shown in Fig. 5.17. Assume that the rules are

$$R_j: \text{ If } x_1 \text{ is } X_1 \text{ and } \ldots \text{ and } x_n \text{ is } X_n, \text{ then } \dot{\mathbf{x}} = \mathbf{A}_{(j)}\mathbf{x} + \mathbf{b}_{(j)}u, \quad (5.125)$$

where $X_i \in \{N_i, P_i\}$, ($i = 1,\ldots,n$, $j = 1,\ldots,2^n$), $\mathbf{A}_{(j)}$ is a local state matrix and $\mathbf{b}_{(j)}$ is a local control vector from the jth region. Observe that the input $u(t)$ is not contained in the premises of the rules. Thus, we consider a collection of 2^n linear dynamical local models with the matrices

$$\mathbf{A}_{(j)} = \begin{bmatrix} \mathbf{a}_{1,(j)}^T \\ \vdots \\ \mathbf{a}_{n,(j)}^T \end{bmatrix} = \begin{bmatrix} a_{11,(j)} & \cdots & a_{1n,(j)} \\ \vdots & \ddots & \vdots \\ a_{n1,(j)} & \cdots & a_{nn,(j)} \end{bmatrix}, \quad \mathbf{b}_{(j)} = \begin{bmatrix} b_{1,(j)} \\ \vdots \\ b_{n,(j)} \end{bmatrix}, \quad j = 1,\ldots,2^n. \quad (5.126)$$

The rules (5.125) are a special case of those investigated in Section 3.4, where the inference was concerned with the *structure parameters* represented by matrices describing the local models. Therefore the nonlinear model of the whole system was inferred as a weighted sum of linear state equations. In

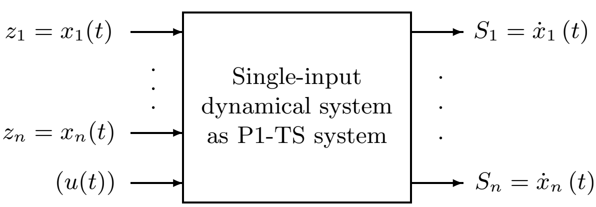


Fig. 5.17 The inputs and the outputs of the P1-TS system defined by the fuzzy rules (5.125)

contrast to that approach, in this section we return to the *original Takagi-Sugeno inference* method.

In every jth region we have a separate fuzzy rule with the following consequent

$$\dot{x}_i = \mathbf{a}_{i,(j)}^T \mathbf{x} + b_{i,(j)} u = \mathbf{M}_i \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \qquad j = 1, \dots, 2^n.$$

For all regions and the i-th system output $S = \dot{x}_i$ we define the following vector of the consequents of the rules

$$\mathbf{q}_i = \begin{bmatrix} q_{1,i} \\ q_{2,i} \\ \vdots \\ q_{2^n,i} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{i,(1)}^T & b_{i,(1)} \\ \mathbf{a}_{i,(2)}^T & b_{i,(2)} \\ \vdots & \vdots \\ \mathbf{a}_{i,(2^n)}^T & b_{i,(2^n)} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \mathbf{M}_i \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \qquad i = 1, \dots, n.$$

This means that every consequent of the rule refers to the same system output $\dot{x}_i$. From the theory of P1-TS systems it follows that the crisp system output inferred from the rule-base is $S = \boldsymbol{\theta}^T \mathbf{g}(x_1, \dots, x_n)$, where $\boldsymbol{\theta}$ is a 2^n-dimensional vector of coefficients and $\mathbf{g}(x_1, \dots, x_n)$ is the system generator. Using the original Takagi-Sugeno inference method, the inferred model for $\dot{x}_i$ is given by

$$\dot{x}_i = \mathbf{q}_i^T \boldsymbol{\Omega}^{-1} \mathbf{g}(\mathbf{x}) = \left[\mathbf{x}^T, u\right] \mathbf{M}_i^T \boldsymbol{\Omega}^{-1} \mathbf{g}(\mathbf{x}) \ , \tag{5.127}$$

where

$$\mathbf{M}_i^T = \begin{bmatrix} \mathbf{a}_{i,(1)} & \mathbf{a}_{i,(2)} & \cdots & \mathbf{a}_{i,(2^n)} \\ b_{i,(1)} & b_{i,(2)} & \cdots & b_{i,(2^n)} \end{bmatrix}, \tag{5.128}$$

for $i = 1, \dots, n$. Thus, for all state variables we can write

$$\dot{\mathbf{x}} = \left(\left[\mathbf{x}^T, u\right] \otimes \underbrace{\left[1, \cdots, 1\right]}_{n} \right) \begin{bmatrix} \mathbf{M}_1^T \\ \vdots \\ \mathbf{M}_n^T \end{bmatrix} \boldsymbol{\Omega}^{-1} \mathbf{g}(\mathbf{x}), \tag{5.129}$$

where $\otimes$ is the Kronecker symbol.

Theorem 5.15. *Suppose we model a dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$, where $\mathbf{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$ and the components of $\mathbf{f}$ belong to a subclass of Kolmogorov-Gabor polynomials, by using P1-TS system with linear membership functions (2.11)-(2.12) for the state variables x_i, $(i = 1, \dots, n)$ and 2^n fuzzy rules in the form (5.125).*

1. Such P1-TS system is equivalent to the nonlinear dynamical system

$$\dot{\mathbf{x}} = \mathbf{W} \mathbf{g}_u(\mathbf{x}), \tag{5.130}$$

where $\mathbf{W} \in \mathbb{R}^{n \times \dim \mathbf{g}_u(\mathbf{x})}$ and $\mathbf{g}_u(\mathbf{x})$ is a modified generator defined by

$$\mathbf{g}_u(\mathbf{x}) = F\left(\left[\begin{array}{c} g_{(1)}(\mathbf{x})\begin{bmatrix}\mathbf{x}\\ u\end{bmatrix} \\ \vdots \\ g_{(2^n)}(\mathbf{x})\begin{bmatrix}\mathbf{x}\\ u\end{bmatrix}\end{array}\right]\right), \tag{5.131}$$

where $g_{(v)}(\mathbf{x})$ is vth component of the generator $\mathbf{g}(\mathbf{x})$, ($v = 1, \ldots, 2^n$), and $F(\mathbf{h})$ is the operator which removes the repeated elements of the vector $\mathbf{h}$, e.g. $F([h_1, h_2, h_1, h_3]) = [h_1, h_2, h_3]$. The elements of the matrix $\mathbf{W}$ depend on boundaries "$-\alpha_i$", "β_i", ($i = 1, \ldots, n$), the elements of $\mathbf{A}_{(j)}$ and $b_{(j)}$, ($j = 1, \ldots, 2^n$).

2. *The length of the generator $\mathbf{g}_u(\mathbf{x})$ is equal to*

$$\dim \mathbf{g}_u(\mathbf{x}) = 2^{n-1}(n+4) - 1. \tag{5.132}$$

Proof. Observe that the components of the generator $\mathbf{g}_u(\mathbf{x})$ are summands contained in the polynomial

$$\left(\sum_{k=1}^{n} x_k + u\right)\prod_{k=1}^{n}(1 + x_k), \tag{5.133}$$

written in the expanded additive form, when substituting in the monomials of the polynomial (5.133) all coefficients by "1". The rest of the first part of the thesis of the theorem follows immediately from the rules (5.125).

Now we prove the formula (5.132). The length of $\mathbf{g}_u(\mathbf{x})$, denoted by $\dim \mathbf{g}_u(\mathbf{x})$, is for $n = 1$ equal to $\dim \mathbf{g}_u(\mathbf{x})|_{\dim \mathbf{x}=1} = 4$. One can check that the following recurrence

$$\dim \mathbf{g}_u(\mathbf{x})|_{\dim \mathbf{x}=k+1} = 2^k + 1 + 2\dim \mathbf{g}_u(\mathbf{x})|_{\dim \mathbf{x}=k}$$

holds. Thus, for $\dim \mathbf{x} = n$ we obtain $\dim \mathbf{g}_u(\mathbf{x}) = 2^{n-1}(n+4) - 1$. □

Example 5.16. The generator (5.131) with $n = 2$ state variables is given by

$$\mathbf{g}_u(x_1, x_2) = \left[\, x_1,\, x_2,\, u,\, x_1^2,\, x_1x_2,\, ux_1,\, x_2^2,\, ux_2,\, x_1^2x_2,\, x_1x_2^2,\, ux_1x_2 \,\right]^T, \tag{5.134}$$

and for $n = 3$ state variables

$$\begin{aligned}\mathbf{g}_u(x_1, x_2, x_3) = [\; & x_1, x_2, x_3, u, x_1^2, x_1x_2, x_1x_3, ux_1, x_2^2, x_2x_3, ux_2,\\ & x_1^2x_2, x_1x_2^2, x_1x_2x_3, ux_1x_2, x_3^2, ux_3, x_1^2x_3, x_1x_3^2,\\ & ux_1x_3, x_2^2x_3, x_2x_3^2, ux_2x_3, x_1^2x_2x_3, x_1x_2^2x_3,\\ & x_1x_2x_3^2, ux_1x_2x_3 \;]^T.\end{aligned} \tag{5.135}$$

The length of $\mathbf{g}_u(\mathbf{x})$ grows faster than $\dim \mathbf{g}(\mathbf{x})$ as shown in Table 5.10.

Table 5.10 Lengths of the generators $\mathbf{g}(\mathbf{x})$ and $\mathbf{g}_u(\mathbf{x})$ for the dynamical P1-TS fuzzy system with n state variables

$n = \dim(\mathbf{x})$	$\dim \mathbf{g}(\mathbf{x})$	$\dim \mathbf{g}_u(\mathbf{x})$
1	2	4
2	4	11
3	8	27
4	16	63
5	32	143
6	64	319
7	128	703
8	256	1535
9	512	3327
10	1024	7167

Example 5.17. Let us consider a two-dimensional nonlinear dynamical system modeled by 4 fuzzy rules:

$$R_1 : \text{If } x_1 \text{ is } N_1 \text{ and } x_2 \text{ is } N_2, \quad \text{then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(1)} & a_{12,(1)} \\ a_{21,(1)} & a_{22,(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{1,(1)} \\ b_{2,(1)} \end{bmatrix} u,$$

$$R_2 : \text{If } x_1 \text{ is } P_1 \text{ and } x_2 \text{ is } N_2, \quad \text{then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(2)} & a_{12,(2)} \\ a_{21,(2)} & a_{22,(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{1,(2)} \\ b_{2,(2)} \end{bmatrix} u,$$

$$R_3 : \text{If } x_1 \text{ is } N_1 \text{ and } x_2 \text{ is } P_2, \quad \text{then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(3)} & a_{12,(3)} \\ a_{21,(3)} & a_{22,(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{1,(3)} \\ b_{2,(3)} \end{bmatrix} u,$$

$$R_4 : \text{If } x_1 \text{ is } P_1 \text{ and } x_2 \text{ is } P_2, \quad \text{then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(4)} & a_{12,(4)} \\ a_{21,(4)} & a_{22,(4)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{1,(4)} \\ b_{2,(4)} \end{bmatrix} u.$$

From the equations (5.127)-(5.128) we obtain

$$\dot{x}_1 = \begin{bmatrix} x_1 & x_2 & u \end{bmatrix} \begin{bmatrix} a_{11,(1)} & a_{11,(2)} & a_{11,(3)} & a_{11,(4)} \\ a_{12,(1)} & a_{12,(2)} & a_{12,(3)} & a_{12,(4)} \\ b_{1,(1)} & b_{1,(2)} & b_{1,(3)} & b_{1,(4)} \end{bmatrix} \boldsymbol{\Omega}^{-1} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}, \tag{5.136}$$

$$\dot{x}_2 = \begin{bmatrix} x_1 \ x_2 \ u \end{bmatrix} \begin{bmatrix} a_{21,(1)} & a_{21,(2)} & a_{21,(3)} & a_{21,(4)} \\ a_{22,(1)} & a_{22,(2)} & a_{22,(3)} & a_{22,(4)} \\ b_{2,(1)} & b_{2,(2)} & b_{2,(3)} & b_{2,(4)} \end{bmatrix} \mathbf{\Omega}^{-1} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}, \tag{5.137}$$

where the fundamental matrix $\mathbf{\Omega}$ is the same as in (2.38). For $\mathbf{g}_u(x_1, x_2)$ given by (5.134) we obtain the following inferred model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} h_{11} \ h_{21} \ h_{31} \ h_{12} \ h_{13} + h_{22} \ h_{32} \ h_{23} \ h_{33} \ h_{14} \ h_{24} \ h_{34} \\ k_{11} \ k_{21} \ k_{31} \ k_{12} \ k_{13} + k_{22} \ k_{32} \ k_{23} \ k_{33} \ k_{14} \ k_{24} \ k_{34} \end{bmatrix} \mathbf{g}_u(x_1, x_2) \tag{5.138}$$

where k_{ij} and h_{ij} can be computed from

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} = \frac{1}{V_2} \begin{bmatrix} a_{11,(1)} & a_{11,(2)} & a_{11,(3)} & a_{11,(4)} \\ a_{12,(1)} & a_{12,(2)} & a_{12,(3)} & a_{12,(4)} \\ b_{1,(1)} & b_{1,(2)} & b_{1,(3)} & b_{1,(4)} \end{bmatrix} \begin{bmatrix} \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \alpha_1\beta_2 & \beta_2 & -\alpha_1 & -1 \\ \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}, \tag{5.139}$$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \end{bmatrix} = \frac{1}{V_2} \begin{bmatrix} a_{21,(1)} & a_{21,(2)} & a_{21,(3)} & a_{21,(4)} \\ a_{22,(1)} & a_{22,(2)} & a_{22,(3)} & a_{22,(4)} \\ b_{2,(1)} & b_{2,(2)} & b_{2,(3)} & b_{2,(4)} \end{bmatrix} \begin{bmatrix} \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \alpha_1\beta_2 & \beta_2 & -\alpha_1 & -1 \\ \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}, \tag{5.140}$$

by $V_2 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2)$.

Theorem 5.18. *The P1-TS system given by the rules (5.125) defines a linear dynamical system if, and only if all local matrices are the same.*

Proof. (Sufficiency) Let us consider a variable x_i for an arbitrarily given index i. From assumption we have $\mathbf{A}_{(1)} = ... = \mathbf{A}_{(2^n)} = \mathbf{A}$ and $\mathbf{b}_{(1)} = \mathbf{b}_{(2)} = ... = \mathbf{b}_{(2^n)} = \mathbf{b}$. Thus,

$$\dot{x}_i = \begin{bmatrix} \mathbf{x}^T \ u \end{bmatrix} \mathbf{M}_i^T \mathbf{\Omega}^{-1} \mathbf{g}(\mathbf{x}), \qquad \mathbf{M}_i^T = \begin{bmatrix} \mathbf{a}_i & \mathbf{a}_i & \cdots & \mathbf{a}_i \\ b_i & b_i & \cdots & b_i \end{bmatrix} \in \mathbb{R}^{(n+1)\times 2^n},$$

and we obtain

$$\mathbf{M}_i^T \mathbf{\Omega}^{-1} = \begin{bmatrix} \mathbf{a}_i & \mathbf{0} & \cdots & \mathbf{0} \\ b_i & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times 2^n}.$$

The columns of $\mathbf{W}$ with numbers $j = n+1, n+2, \ldots, 2^{n-1}(n+4) - 1$ are zero. We conclude that $\dot{x}_i$ inferred from the fuzzy rules (5.125) is as follows

$$\dot{x}_i = \begin{bmatrix} \mathbf{x}^T \mathbf{a}_i + b_i u, \ 0, \ \cdots, \ 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_1 x_2 \ldots x_n \end{bmatrix} = \mathbf{a}_i^T \mathbf{x} + b_i u.$$

The result is valid for any $i \in \{1, \ldots, n\}$. This ends the proof of sufficiency. (Necessity) The necessary condition one can easily prove constructing a counterexample for say $n = 2$. □

Example 5.19. Consider the set of differential equations describing the van de Vusse reaction with the following kinetic reaction scheme:

$$A \overset{k_1}{\rightarrow} B \overset{k_2}{\rightarrow} C,$$
$$A \overset{k_3}{\rightarrow} D,$$

which is carried out in an isothermal, continuously mixed reactor (CSTR) [25], [35]

$$\left.\begin{aligned} \frac{dC_A}{dt} &= -k_1 C_A - k_3 C_A^2 + \frac{F}{V}\left(C_{Af} - C_A\right) \ , \\ \frac{dC_B}{dt} &= k_1 C_A - k_2 C_B - \frac{F}{V} C_B \ , \end{aligned}\right\} \tag{5.141}$$

where C_A and C_B are concentrations of the components A and B, respectively, $F(t) = u(t)$ is the process input and $C_B(t) = y(t)$ is the process output. Our goal is to obtain the fuzzy rules for P1-TS system which exactly models the reactor. For the sake of simplicity we can take the numerical parameters: $k_1 = 50 \left[\mathrm{h}^{-1}\right]$, $k_2 = 100 \left[\mathrm{h}^{-1}\right]$, $k_3 = 10 \left[1/(\mathrm{mol\,h})\right]$, $C_{Af} = 10$ [mol / l] and $V = 1$ [l]. The nominal operation conditions are $C_A = 3.0$ [mol / l], $C_B = 1.12$ [mol / l] and $F = 34.3$ [l / h]. The inputs of the P1-TS system are concentrations $x_1 = C_A \in [-\alpha_1, \beta_1] = [0, 6]$ and $x_2 = C_B = [-\alpha_2, \beta_2] = [0, 2.2]$.

The differential equations (5.141) may be rewritten as follows

$$\left.\begin{aligned} \dot{x}_1 &= -k_1 x_1 - k_3 x_1^2 + c_1 u - c_2 u x_1 \ , \\ \dot{x}_2 &= k_1 x_1 - k_2 x_2 - c_2 x_2 u \ , \end{aligned}\right\} \tag{5.142}$$

where $c_1 = C_{Af}/V = 10 \left[\mathrm{mol}/\mathrm{l}^2\right]$ and $c_2 = 1/V = 1$ [1/l]. According to (5.138)-(5.140) we can write the above differential equations in the following form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \mathbf{W}\left[x_1, x_2, u, x_1^2, x_1 x_2, u x_1, x_2^2, u x_2, x_1^2 x_2, x_1 x_2^2, u x_1 x_2\right]^T ,$$

where

$$\mathbf{W} = \begin{bmatrix} -k_1 & 0 & c_1 & -k_3 & 0 & -c_2 & 0 & 0 & 0 & 0 & 0 \\ k_1 & -k_2 & 0 & 0 & 0 & 0 & 0 & -c_2 & 0 & 0 & 0 \end{bmatrix}.$$

From (5.139)-(5.140) we get the matrices containing coefficients of local linear models

$$\begin{bmatrix} a_{11,(1)} & a_{11,(2)} & a_{11,(3)} & a_{11,(4)} \\ a_{12,(1)} & a_{12,(2)} & a_{12,(3)} & a_{12,(4)} \\ b_{1,(1)} & b_{1,(2)} & b_{1,(3)} & b_{1,(4)} \end{bmatrix} = \begin{bmatrix} \alpha_1 k_3 - k_1 & 0 & c_1 + \alpha_1 c_2 \\ -k_1 - \beta_1 k_3 & 0 & c_1 - \beta_1 c_2 \\ \alpha_1 k_3 - k_1 & 0 & c_1 + \alpha_1 c_2 \\ -k_1 - \beta_1 k_3 & 0 & c_1 - \beta_1 c_2 \end{bmatrix}^T, \quad (5.143)$$

$$\begin{bmatrix} a_{21,(1)} & a_{21,(2)} & a_{21,(3)} & a_{21,(4)} \\ a_{22,(1)} & a_{22,(2)} & a_{22,(3)} & a_{22,(4)} \\ b_{2,(1)} & b_{2,(2)} & b_{2,(3)} & b_{2,(4)} \end{bmatrix} = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 \\ -k_2 & -k_2 & -k_2 & -k_2 \\ c_2\alpha_2 & c_2\alpha_2 & -c_2\beta_2 & -c_2\beta_2 \end{bmatrix}. \quad (5.144)$$

Finally we obtain the following system of fuzzy rules for the P1-TS system which exactly models the van de Vusse reactor (5.141):

R_1 : If x_1 is N_1 and x_2 is N_2, then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 + k_3\alpha_1 & 0 \\ k_1 & -k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_2\alpha_1 \\ c_2\alpha_2 \end{bmatrix} u,$$

R_2 : If x_1 is P_1 and x_2 is N_2, then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 - k_3\beta_1 & 0 \\ k_1 & -k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_1 - c_2\beta_1 \\ c_2\alpha_2 \end{bmatrix} u,$$

R_3 : If x_1 is N_1 and x_2 is P_2, then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 + k_3\alpha_1 & 0 \\ k_1 & -k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_2\alpha_1 \\ -c_2\beta_2 \end{bmatrix} u,$$

R_4 : If x_1 is P_1 and x_2 is P_2, then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -k_1 - k_3\beta_1 & 0 \\ k_1 & -k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} c_1 - c_2\beta_1 \\ -c_2\beta_2 \end{bmatrix} u.$$

Example 5.19 is simple, but for a bigger dimension of the state space, the course of dimensionality problem becomes more vexatious for the P1-TS systems that use the modified generator $\mathbf{g}_u$, than for those which use the generator $\mathbf{g}$. This is evident from Table 5.10. Fortunately, we can use recursion for computation.

Remark 5.20. Theorems 3.6 and 3.7 (on recursion) are valid when we use the original Takagi-Sugeno method of reasoning for the P1-TS systems with the fuzzy rules in the form of (5.125).

Proof. The proof can be constructed analogously to that of Theorem 3.7 and will be omitted. □

Example 5.21. The Rössler system was considered in [74] in the form of differential equations

$$\left.\begin{aligned}\dot{x}_1 &= -x_2 + x_3\ ,\\ \dot{x}_2 &= x_1 + ax_2\ ,\\ \dot{x}_3 &= bx_1 - cx_3 + x_1x_3 + u.\end{aligned}\right\} \tag{5.145}$$

The system was modeled by 2 fuzzy metarules as follows

$$M_1 : \text{If } x_1 \text{ is } N_1, \text{ then } \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & -1 & -1\\ 1 & a & 0\\ b & 0 & -d\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u,$$

$$M_2 : \text{If } x_1 \text{ is } P_1, \text{ then } \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & -1 & -1\\ 1 & a & 0\\ b & 0 & d\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u.$$

Our goal is to show in two ways that the above system of rules defines the model (5.145).

1. Assuming the inputs of the dynamical P1-TS system to be $x_1 \in [-\alpha_1, \beta_1] = [c-d, c+d]$, $x_2 \in [-\alpha_2, \beta_2]$ and $x_3 \in [-\alpha_3, \beta_3]$ and the outputs $S_k = \dot{x}_k$ for $k = 1, 2, 3$, from (5.129) we have

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \left(\begin{bmatrix}x_1 & x_2 & x_3 & u\end{bmatrix} \otimes \begin{bmatrix}1 & 1 & 1\end{bmatrix}\right)\begin{bmatrix}\mathbf{M}_1^T\\ \mathbf{M}_2^T\\ \mathbf{M}_3^T\end{bmatrix}\mathbf{\Omega}^{-1}\mathbf{g}\left(x_1, x_2, x_3\right), \tag{5.146}$$

where

$$\mathbf{M}_1^T = \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \tag{5.147}$$

$$\mathbf{M}_2^T = \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ a & a & a & a & a & a & a & a\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \tag{5.148}$$

$$\mathbf{M}_3^T = \begin{bmatrix}b & b & b & b & b & b & b & b\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ -d & d & -d & d & -d & d & -d & d\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}, \tag{5.149}$$

the generator $\mathbf{g}$ and the fundamental matrix $\mathbf{\Omega}$ are given in (2.40) and (2.41), respectively. According to (5.146) and (5.147)-(5.149) we obtain

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_2 - x_3 \\ x_1 + ax_2 \\ bx_1 + \dfrac{\alpha_1 - \beta_1}{\alpha_1 + \beta_1} dx_3 + \dfrac{2d}{\alpha_1 + \beta_1} x_1 x_3 + u \end{bmatrix}. \tag{5.150}$$

After substitutions $-\alpha_1 = c - d$ and $\beta_1 = c + d$ we conclude that the metarules M_1 and M_2 define exactly the Rössler system (5.145), indeed. It is worth noting that the result (5.150) does not depend on α_2, β_2, α_3 or β_3. This fact agrees with the metarules.

2. Now we will use the recursion in accordance with Remark 5.20. For the linear membership functions of fuzzy sets it can be expressed as follows

$$\begin{aligned} S_n(\mathbf{x} \mid \mathbf{q}_1, \ldots, \mathbf{q}_{2^n}) = {} & N_n(x_n) S_{n-1}(x_1, \ldots, x_{n-1} \mid \mathbf{q}_1, \ldots, \mathbf{q}_{2^{n-1}}) \\ & + P_n(x_n) S_{n-1}(x_1, \ldots, x_{n-1} \mid \mathbf{q}_{2^{n-1}+1}, \ldots, \mathbf{q}_{2^n}), \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is the state vector and the membership functions are $N_n(x_n) = (\beta_n - x_n) / (\alpha_n + \beta_n)$, and $P_n(x_n) = (\alpha_n + x_n) / (\alpha_n + \beta_n)$. For the given metarules M_1 and M_2 we establish the following vectors constituting conclusions for each individual fuzzy rule

$$\mathbf{q}_1 = \mathbf{q}_3 = \mathbf{q}_5 = \mathbf{q}_7 = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & a & 0 & 0 \\ b & 0 & -d & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix},$$

$$\mathbf{q}_2 = \mathbf{q}_4 = \mathbf{q}_6 = \mathbf{q}_8 = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & a & 0 & 0 \\ b & 0 & d & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix}.$$

Using the original Takagi-Sugeno reasoning method, from Theorem 3.6 for $\mathbf{q}_1 = \mathbf{q}_3$, $\mathbf{q}_2 = \mathbf{q}_4$ and knowing that $N_2(x_2) + P_2(x_2) = 1$ we immediately obtain

$$S_2(x_1, x_2 \mid \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = S_1(x_1 \mid \mathbf{q}_1, \mathbf{q}_2).$$

For $\mathbf{q}_1 = \mathbf{q}_5, \mathbf{q}_2 = \mathbf{q}_6, \mathbf{q}_3 = \mathbf{q}_7$ and $\mathbf{q}_4 = \mathbf{q}_8$ by $N_3(x_3) + P_3(x_3) = 1$ we get

$$S_3(x_1, x_2, x_3 \mid \mathbf{q}_1, \ldots, \mathbf{q}_8) = S_2(x_1, x_2 \mid \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = S_1(x_1 \mid \mathbf{q}_1, \mathbf{q}_2).$$

This means that the inferred model can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_k \end{bmatrix} = \frac{\beta_1 - x_1}{\alpha_1 + \beta_1} \begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & a & 0 & 0 \\ b & 0 & -d & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix} + \frac{x_1 + \alpha_1}{\alpha_1 + \beta_1} \begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & a & 0 & 0 \\ b & 0 & d & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ u \end{bmatrix}. \tag{5.151}$$

The above equations are the same as (5.150).

As one can see, recurrence simplifies the derivation of the conventional model of a process for a given system of fuzzy rules.

5.7 Exact Modeling of MIMO Linear Dynamical Systems

Let us consider a MIMO P1-TS system that models a MIMO linear dynamical system. The inputs and the outputs of the MIMO P1-TS system are shown in Fig. 5.18, where $x_1, \ldots, x_n$ are the state variables, $u_1, \ldots, u_m$ are control actions and $y_1, \ldots, y_l$ are the outputs of a modeled linear dynamical system.

According to the theory of P1-TS systems we can formulate the following

Remark 5.22. Suppose the fuzzy sets for all inputs of a MIMO P1-TS system shown in Fig. 5.18 are linear as was defined in (2.11)-(2.12) in Section 2.2, and both state variables and control actions are contained in the premises of the rules. Such a rule-based system is an exact model of

$$\left.\begin{aligned} s\mathbf{x}(t) &= \sum_{\substack{(p_1,p_2,\ldots,p_n)\in\{0,1\}^{n+m}\\ z_k\in\{x_k,u_k\},\ k\in\{1,\ldots,n+m\}}} \theta_{p_1,p_2,\ldots,p_{n+m}} z_1^{p_1} z_2^{p_2} \cdots z_{n+m}^{p_{n+m}}, \\ \mathbf{y}(t) &= \sum_{\substack{(p_1,p_2,\ldots,p_n)\in\{0,1\}^{n+m}\\ z_k\in\{x_k,u_k\},\ k\in\{1,\ldots,n+m\}}} \xi_{p_1,p_2,\ldots,p_{n+m}} z_1^{p_1} z_2^{p_2} \cdots z_{n+m}^{p_{n+m}}, \end{aligned}\right\} \tag{5.152}$$

where

$$s\mathbf{x}(t) = \begin{cases} \dot{\mathbf{x}}(t) & \text{- for a continuos case, } t \geq 0, \\ \mathbf{x}(t+1) & \text{- for a discrete case, } t = 0, 1, 2, \ldots, \end{cases}$$

and the coefficients $\theta_{(\cdot)}$ and $\xi_{(\cdot)}$ are real numbers.

Without loss of generality we will investigate the continuous models. A special case of (5.152) is the linear dynamical system written in the standard form

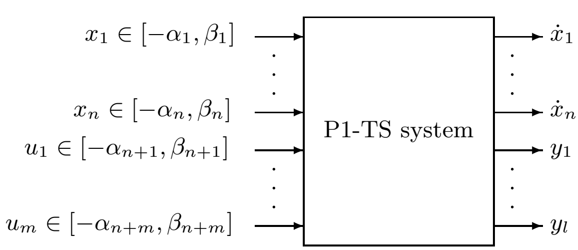


Fig. 5.18 The inputs and the outputs of MIMO P1-TS system which exactly models the dynamical system (5.153)

$$\left.\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\mathbf{u}(t), \quad \mathbf{x}(0) \in D^n,\\ \mathbf{y}(t) &= \mathbf{C}(t)\,\mathbf{x}(t) + \mathbf{D}(t)\,\mathbf{u}(t), \quad t \geq 0,\end{aligned}\right\} \tag{5.153}$$

where $\mathbf{A}(t) : \mathbb{R}_+ \to \mathbb{R}^{n\times n}$ is the state matrix, $\mathbf{B}(t) : \mathbb{R}_+ \to \mathbb{R}^{n\times m}$ is the input matrix, $(m < n)$, $\mathbf{C}(t) : \mathbb{R}_+ \to \mathbb{R}^{l\times n}$ is the output matrix and $\mathbf{D}(t) : \mathbb{R}_+ \to \mathbb{R}^{l\times m}$ is the feedthrough (or feedforward) matrix, $(l \leq n,\ \mathbb{R}_+ = [0,\infty))$. The vectors $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{y}$ are from the hypercuboids: $\mathbf{x}(t) \in D^n$, $\mathbf{u}(t) \in D^m$ and $\mathbf{y}(t) \in D^l$ for $t \geq 0$.

Our goal is to show a simple and effective method of how to obtain an exact model of the linear dynamical system (5.153) in the form of MIMO P1-TS system.

Theorem 5.23. *Let us define the MIMO P1-TS system shown in Fig. 5.18, for which the following conditions are satisfied:*

1. *The inputs are $x_k \in [-\alpha_k, \beta_k]$, $(k = 1,\ldots,n)$ and $u_k \in [-\alpha_{n+k}, \beta_{n+k}]$, $(k = 1,\ldots,m)$ and two linear fuzzy sets for every input are defined by (2.11)-(2.12).*
2. *The outputs are $\dot{x}_k(t)$, $(k = 1,\ldots,n)$ and $y_k(t)$, $(k = 1,\ldots,l)$.*
3. *The TS system is defined by 2^{n+m} fuzzy rules and each single rule is of the form:*

R_v: *If x_1 is A_1 and ... and x_n is A_n and u_1 is A_{n+1}*
and ... and u_m is A_{n+m},
then $\dot{x}_1 = q_{v,1}$, ..., $\dot{x}_n = q_{v,n}$, $y_1 = q_{v,n+1}$, ..., $y_l = q_{v,n+l}$,

where A_i are the labels of the fuzzy sets, $(A_i \in \{N_i, P_i\},\ i = 1,\ldots,n+m)$. Such a rule-based system is an exact model of the linear dynamical system (5.153), if and only if the consequents of the fuzzy rules are computed according to the following equations

$$\begin{aligned}\dot{x}_i &= q_{v,i} = \boldsymbol{\gamma}_v^T \mathbf{r}_i, \quad \text{for} \quad i = 1,\ldots,n,\\ y_i &= q_{v,n+i} = \boldsymbol{\gamma}_v^T \mathbf{r}_{n+i}, \quad \text{for} \quad i = 1,\ldots,l,\\ \boldsymbol{\gamma}_v &\in \Gamma^{n+m},\end{aligned} \tag{5.154}$$

where Γ^{n+m} is the set of vertices of the hypercuboid D^{n+m} and $\mathbf{r}_1,\ldots,\mathbf{r}_{n+l}$ are the row vectors given by

$$\begin{bmatrix}\mathbf{r}_1\\ \vdots\\ \mathbf{r}_{n+l}\end{bmatrix} = \begin{bmatrix}\mathbf{A}(t) & \mathbf{B}(t)\\ \mathbf{C}(t) & \mathbf{D}(t)\end{bmatrix}. \tag{5.155}$$

Equivalently, the same consequents can be expressed in the matrix form

$$\left[\dot{\mathbf{x}}^T, \mathbf{y}^T\right]_v = \mathbf{L}^T \begin{bmatrix}\mathbf{A}(t) & \mathbf{B}(t)\\ \mathbf{C}(t) & \mathbf{D}(t)\end{bmatrix}, \quad v = 1,\ldots,2^{n+m}, \tag{5.156}$$

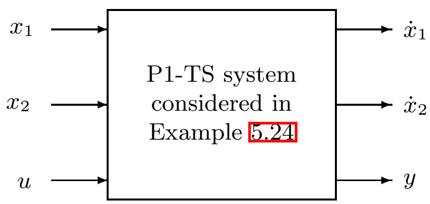


Fig. 5.19 A zero-order TS system as an exact model of a linear second-order dynamical system

where $\left[\dot{\mathbf{x}}^T, \mathbf{y}^T\right]_v = [\dot{x}_1, \ldots, \dot{x}_n, y_1, \ldots, y_l]_v$ *is the output vector of the P1-TS system in the case of the vth fuzzy rule. The matrix* $\mathbf{L} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{2^{n+m}}]$ *is a part of the fundamental matrix* $\boldsymbol{\Omega}$ *of the system. It is constructed based on the generator* $\mathbf{g}(x_1, \ldots, x_n, u_1, \ldots, u_m)$.

Proof. The inputs of the MIMO P1-TS system are components of the vector $\mathbf{z} = [x_1, \ldots, x_n, u_1, \ldots, u_m]^T \in D^{n+m}$. By applying the generator $\mathbf{g}(\mathbf{z})$ to the MIMO P1-TS system, the rest of the proof is straightforward. □

Example 5.24. Consider the following MIMO P1-TS system with the inputs (see Fig. 5.19)

- $x_k \in [-\alpha_k, \beta_k]$ and the fuzzy sets for x_k are $N_k(x_k)$ and $P_k(x_k)$, $(k = 1, 2)$,
- $u \in [-\alpha_3, \beta_3]$, and the fuzzy sets for u are $N_3(u)$ and $P_3(u)$.

The outputs are $\dot{x}_1(t)$, $\dot{x}_2(t)$ and $y(t)$. According to Theorem 5.23 for the generator

$$\mathbf{g}(x_1, x_2, u) = [1, x_1, x_2, x_1x_2, u, ux_1, ux_2, ux_1x_2]^T$$

we immediately obtain the following system of 8 fuzzy rules:

$$\begin{aligned}
R_1: \;& \text{If } x_1 \text{ is } N_1 \text{ and } x_2 \text{ is } N_2 \text{ and } u \text{ is } N_3, \text{ then} \\
& \dot{x}_1 = [-\alpha_1, -\alpha_2, -\alpha_3]\,[a_{11}(t), a_{21}(t), b_1(t)]^T, \\
& \dot{x}_2 = [-\alpha_1, -\alpha_2, -\alpha_3]\,[a_{12}(t), a_{22}(t), b_2(t)]^T, \\
& y = [-\alpha_1, -\alpha_2, -\alpha_3]\,[c_1(t), c_2(t), d(t)]^T, \\
R_2: \;& \text{If } x_1 \text{ is } P_1 \text{ and } x_2 \text{ is } N_2 \text{ and } u \text{ is } N_3, \text{ then} \\
& \dot{x}_1 = [\beta_1, -\alpha_2, -\alpha_3]\,[a_{11}(t), a_{21}(t), b_1(t)]^T, \\
& \dot{x}_2 = [\beta_1, -\alpha_2, -\alpha_3]\,[a_{12}(t), a_{22}(t), b_2(t)]^T, \\
& y = [\beta_1, -\alpha_2, -\alpha_3]\,[c_1(t), c_2(t), d(t)]^T,
\end{aligned}$$

R_3 : If x_1 is N_1 and x_2 is P_2 and u is N_3, then

$$\dot{x}_1 = [-\alpha_1, \beta_2, -\alpha_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [-\alpha_1, \beta_2, -\alpha_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [-\alpha_1, \beta_2, -\alpha_3]\left[c_1(t), c_2(t), d(t)\right]^T,$$

R_4 : If x_1 is P_1 and x_2 is P_2 and u is N_3, then

$$\dot{x}_1 = [\beta_1, \beta_2, -\alpha_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [\beta_1, \beta_2, -\alpha_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [\beta_1, \beta_2, -\alpha_3]\left[c_1(t), c_2(t), d(t)\right]^T,$$

R_5 : If x_1 is N_1 and x_2 is N_2 and u is P_3, then

$$\dot{x}_1 = [-\alpha_1, -\alpha_2, \beta_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [-\alpha_1, -\alpha_2, \beta_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [-\alpha_1, -\alpha_2, \beta_3]\left[c_1(t), c_2(t), d(t)\right]^T,$$

R_6 : If x_1 is P_1 and x_2 is N_2 and u is P_3, then

$$\dot{x}_1 = [\beta_1, -\alpha_2, \beta_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [\beta_1, -\alpha_2, \beta_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [\beta_1, -\alpha_2, \beta_3]\left[c_1(t), c_2(t), d(t)\right]^T,$$

R_7 : If x_1 is N_1 and x_2 is P_2 and u is P_3, then

$$\dot{x}_1 = [-\alpha_1, \beta_2, \beta_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [-\alpha_1, \beta_2, \beta_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [-\alpha_1, \beta_2, \beta_3]\left[c_1(t), c_2(t), d(t)\right]^T,$$

R_8 : If x_1 is P_1 and x_2 is P_2 and u is P_3, then

$$\dot{x}_1 = [\beta_1, \beta_2, \beta_3]\left[a_{11}(t), a_{21}(t), b_1(t)\right]^T,$$
$$\dot{x}_2 = [\beta_1, \beta_2, \beta_3]\left[a_{12}(t), a_{22}(t), b_2(t)\right]^T,$$
$$y = [\beta_1, \beta_2, \beta_3]\left[c_1(t), c_2(t), d(t)\right]^T.$$

Equivalently the same rules can be written in the matrix form

$$\text{If } [x_1, x_2, u] \text{ is } \begin{bmatrix} N_1 & N_2 & N_3 \\ P_1 & N_2 & N_3 \\ N_1 & P_2 & N_3 \\ P_1 & P_2 & N_3 \\ N_1 & N_2 & P_3 \\ P_1 & N_2 & P_3 \\ N_1 & P_2 & P_3 \\ P_1 & P_2 & P_3 \end{bmatrix},$$

$$\text{then } [\dot{x}_1, \dot{x}_2, y] \text{ is } \begin{bmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 \\ \beta_1 & -\alpha_2 & -\alpha_3 \\ -\alpha_1 & \beta_2 & -\alpha_3 \\ \beta_1 & \beta_2 & -\alpha_3 \\ -\alpha_1 & -\alpha_2 & \beta_3 \\ \beta_1 & -\alpha_2 & \beta_3 \\ -\alpha_1 & \beta_2 & \beta_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix} \begin{bmatrix} a_{11}(t) & a_{12}(t) & c_1(t) \\ a_{21}(t) & a_{22}(t) & c_2(t) \\ b_1(t) & b_2(t) & d(t) \end{bmatrix}.$$

The above rules exactly model the second-order dynamical system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} b_1(t) \\ b_2(t) \end{bmatrix} u(t),$$
$$y(t) = \begin{bmatrix} c_1(t) & c_2(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + d(t)\,u(t).$$

with an initial condition $[x_1(0), x_2(0)]^T \in D^2$.

The method of obtaining the fuzzy rules for the given MIMO linear dynamical system described by the differential equations is systematic and extremely simple. The resulting rule-based system can be viewed as an alternative description substituting the differential equations. The same can be said about nonlinear dynamical systems described by (5.152).

5.8 Strong Triangular Fuzzy Partition

A very popular method used for fuzzy modeling is the so called strong triangular fuzzy partition for input variables. Assume that there are J_k fuzzy sets for the input variable z_k which make a strong fuzzy partition of the interval $[m_{k,1}, m_{k,J_k}]$ into $(J_k - 1)$ subintervals

$$[m_{k,1}, m_{k,J_k}] = [m_{k,1}, m_{k,2}] \cup [m_{k,2}, m_{k,3}] \cup \ldots \cup [m_{k,J_k-1}, m_{k,J_k}]$$

as shown in Fig. 5.20. For every z_k from the universe of discourse $[m_{k,1}, m_{k,J_k}]$, there are either exactly two fuzzy sets with nonzero memberships, or exactly one fuzzy set with full membership. Thus, the whole rule-based system is equivalent to $\prod_{k=1}^{n} (J_k - 1)$ distinct P1-TS subsystems $TS_{j_1,\ldots,j_n}$, ($j_k = 1, \ldots, (J_k - 1)$, $k = 1, \ldots, n$). Using the P1-TS system notation, the boundaries of the subsequent subintervals for the input z_k are

$$m_{k,j} = -\alpha_{k,j}, \quad m_{k,j+1} = \beta_{k,j} = -\alpha_{k,j+1}, \qquad j = 1, 2, \ldots, (J_k - 1).$$

Since the membership grades are maximal at the points $m_{k,j}$, the antecedents of the fuzzy rules do refer approximately to these points. Observe that the systems with strong triangular partition do not require a special theory, since

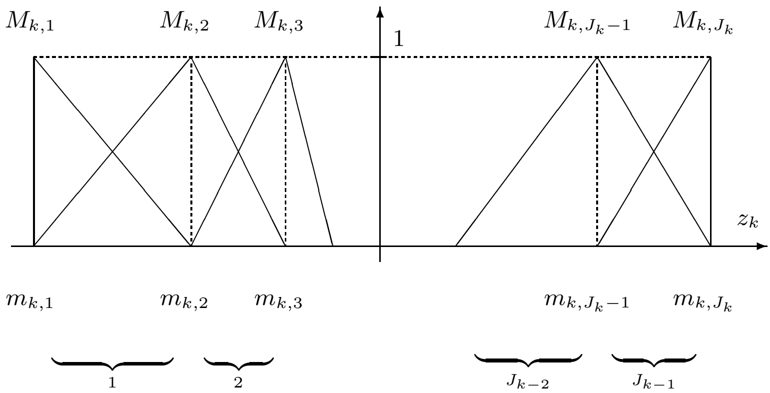


Fig. 5.20 Strong triangular partition

their contribution lies only in a smaller number of the fuzzy rules, in comparison with the P1-TS systems considered till now.

Let us assume that a strong fuzzy partition is applied to a zero-order TS system with n inputs containing the complete and noncontradictory fuzzy rules. By n_{MIN} we denote the minimal number of fuzzy rules, and by n_{REQ} - the number of required fuzzy rules that have to be considered using the theory of P1-TS system. The relationship between the numbers n_{MIN} and n_{REQ} is as follows

$$n_{MIN} = \prod_{k=1}^{n} J_k \leq n_{REQ} = 2^n \prod_{k=1}^{n} (J_k - 1) , \tag{5.157}$$

where $n > 1$ and $\min\{J_1, \ldots, J_n\} \geq 2$. Assuming additionally that the number of fuzzy sets is the same for every input, i.e. $J_k = J$ for $k = 1, \ldots, n$, the pairs of numbers n_{MIN}/n_{REQ} are illustrated in Table 5.11.

Below we will exemplify that for systems with a strong triangular fuzzy partition it would be advisable to use the results obtained for P1-TS

Table 5.11 Minimal number of individual fuzzy rules against required n_{MIN}/n_{REQ} for the TS systems with a strong triangular fuzzy partition

	$J = 2$	$J = 3$	$J = 4$	$J = 5$
$n = 2$	4/4	9/16	16/36	25/64
$n = 3$	8/8	27/64	64/216	125/512
$n = 4$	16/16	81/256	256/1 296	625/4 096
$n = 5$	32/32	243/1 024	1 024/7 776	3125/32 768
$n = 6$	64/64	729/4 096	4 096/46 656	15 625/262 144

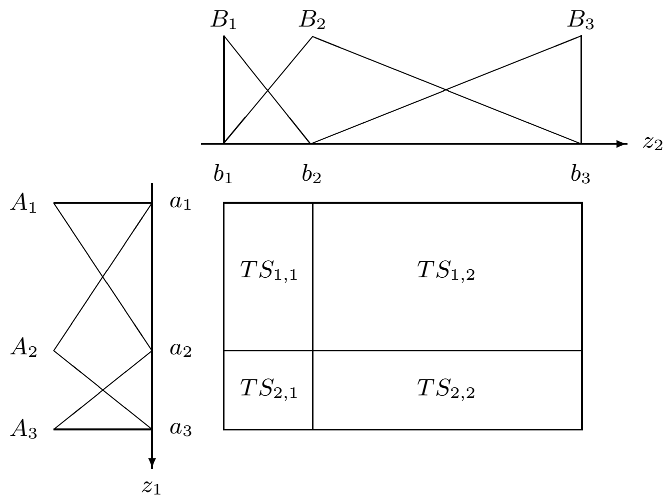


Fig. 5.21 Triangular fuzzy partition for the TS system from Example 5.25

systems. The first example will be abstract, but the second one will refer to the practical design of a simple navigation system for a mobile robot.

Example 5.25. Consider a TS system with the inputs z_1 and z_2, where for each z_k three triangular fuzzy sets are assigned. For the sake of simplicity the membership functions and boundaries of intervals are denoted by (see Fig. 5.21)

$$M_{1,j}(z_1) = A_j(z_1), \; M_{2,j}(z_2) = B_j(z_2), \; m_{1,j} = a_j, \; m_{2,j} = b_j, \; j = 1, 2, 3.$$

The output of the whole TS system is as follows (Fig. 5.21):

$$S(z_1, z_2) = \begin{cases} S_{1,1}(z_1, z_2) & \text{for} \quad (z_1, z_2) \in [a_1, a_2] \times [b_1, b_2] \\ S_{1,2}(z_1, z_2) & \text{for} \quad (z_1, z_2) \in [a_1, a_2] \times [b_2, b_3] \\ S_{2,1}(z_1, z_2) & \text{for} \quad (z_1, z_2) \in [a_2, a_3] \times [b_1, b_2] \\ S_{2,2}(z_1, z_2) & \text{for} \quad (z_1, z_2) \in [a_2, a_3] \times [b_2, b_3] \end{cases},$$

where $S_{i,k}(z_1, z_2)$ is the output of the subsystem $TS_{i,k}$, $(i, k = 1, 2)$. The system of fuzzy rules is given in Table 5.12.

Formally, the crisp output of the (i, k)th P1-TS subsystem is given by

$$S_{i,k}(\mathbf{z}) = \boldsymbol{\theta}_{i,k}^T \mathbf{g}(\mathbf{z}) = \mathbf{q}_{i,k}^T \boldsymbol{\Omega}_{i,k}^{-1} \mathbf{g}(\mathbf{z}), \qquad i, k = 1, 2,$$

where $\mathbf{g}$ is the generator, $\boldsymbol{\theta}_{i,k}$ is the vector of coefficients, $\mathbf{q}_{i,k}$ is the vector of the consequents of the rules, and $\boldsymbol{\Omega}_{i,k}$ is the fundamental matrix of the (i, k)th subsystem. For the sake of simplicity we ignore indices (i, k). Thus,

Table 5.12 Look-up-table for the TS fuzzy system from Example 5.25

	$B_1(z_2)$	$B_2(z_2)$	$B_3(z_2)$
$A_1(z_1)$	q_1	q_4	q_7
$A_2(z_1)$	q_2	q_5	q_8
$A_3(z_1)$	q_3	q_6	q_9

Table 5.13 Subintervals and consequents of the rules of the P1-TS subsystems from Example 5.25

	P1-TS subsystem			
Parameters	$TS_{1,1}$	$TS_{1,2}$	$TS_{2,1}$	$TS_{2,2}$
$[\alpha_1, \beta_1]$	$[-a_1, a_2]$	$[-a_1, a_2]$	$[-a_2, a_3]$	$[-a_2, a_3]$
$[\alpha_2, \beta_2]$	$[-b_1, b_2]$	$[-b_2, b_3]$	$[-b_1, b_2]$	$[-b_2, b_3]$
$\mathbf{q}_{i,k}$	$\mathbf{q}_{1,1} = \begin{bmatrix} q_1 \\ q_4 \\ q_2 \\ q_5 \end{bmatrix}$	$\mathbf{q}_{1,2} = \begin{bmatrix} q_4 \\ q_7 \\ q_5 \\ q_8 \end{bmatrix}$	$\mathbf{q}_{2,1} = \begin{bmatrix} q_2 \\ q_5 \\ q_3 \\ q_6 \end{bmatrix}$	$\mathbf{q}_{2,2} = \begin{bmatrix} q_5 \\ q_8 \\ q_6 \\ q_9 \end{bmatrix}$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{00} \\ \theta_{10} \\ \theta_{01} \\ \theta_{11} \end{bmatrix}, \ \mathbf{g}(\mathbf{z}) = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \end{bmatrix}, \ \boldsymbol{\Omega} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & -\alpha_1 & \beta_1 & \beta_1 \\ -\alpha_2 & \beta_2 & -\alpha_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_1\beta_2 & -\alpha_2\beta_1 & \beta_1\beta_2 \end{bmatrix}.$$

Taking the parameters from Table 5.13 we immediately obtain the output of the first subsystem

$$S_{1,1} = \theta_{00} + \theta_{10} z_1 + \theta_{01} z_2 + \theta_{11} z_1 z_2,$$

where

$$\theta_{00} = \frac{-a_1 b_2 q_2 + a_2 b_2 q_1 + a_1 b_1 q_5 - a_2 b_1 q_4}{(a_2 - a_1)(b_2 - b_1)},$$

$$\theta_{10} = \frac{b_1 (q_4 - q_5) + b_2 (q_2 - q_1)}{(a_2 - a_1)(b_2 - b_1)},$$

$$\theta_{01} = \frac{a_1 (q_2 - q_5) + a_2 (q_4 - q_1)}{(a_2 - a_1)(b_2 - b_1)},$$

$$\theta_{11} = \frac{q_1 - q_2 - q_4 + q_5}{(a_2 - a_1)(b_2 - b_1)}.$$

For the sake of simplicity let us take

$$(a_1, a_2, a_3) = (-a, 0, a), \qquad (b_1, b_2, b_3) = (-b, 0, b). \tag{5.158}$$

The outcome is as follows

$$S_{1,1}(z_1, z_2) = q_5 + \frac{q_5 - q_4}{a} z_1 + \frac{q_5 - q_2}{b} z_2 + \frac{q_1 - q_2 - q_4 + q_5}{ab} z_1 z_2, \quad (5.159)$$

$$S_{1,2}(z_1, z_2) = q_5 + \frac{q_5 - q_4}{a} z_1 + \frac{q_8 - q_5}{b} z_2 + \frac{q_4 - q_5 - q_7 + q_8}{ab} z_1 z_2, \quad (5.160)$$

$$S_{2,1}(z_1, z_2) = q_5 + \frac{q_6 - q_5}{a} z_1 + \frac{q_5 - q_2}{b} z_2 + \frac{q_2 - q_3 - q_5 + q_6}{ab} z_1 z_2, \quad (5.161)$$

$$S_{2,2}(z_1, z_2) = q_5 + \frac{q_6 - q_5}{a} z_1 + \frac{q_8 - q_5}{b} z_2 + \frac{q_5 - q_6 - q_8 + q_9}{ab} z_1 z_2. \quad (5.162)$$

The result is the same as in [51], when substituting symbols for numbers. One can check that for every point $(z_1, z_2) \in [-a, a] \times [-b, b]$ the functions (5.159)-(5.162) are the same, and the whole rule-based system is equivalent to the polynomial

$$S(z_1, z_2) = A + \frac{B - A}{a} z_1 + \frac{C - A}{b} z_2 + \frac{A - B - C + D}{ab} z_1 z_2,$$

if the consequents of the rules from Table 5.12 are $q_1 = 4A - 2B - 2C + D$, $q_2 = 2A - C$, $q_3 = 2B - D$, $q_4 = 2A - B$, $q_5 = A$, $q_6 = B$, $q_7 = 2C - D$, $q_8 = C$ and $q_9 = D$, for any constants A, B, C, D, and nonzero a and b.

Thus, by taking into account all P1-TS subsystems as in Example 5.25 with the use of the methods described in the previous sections, we can easily analyze the whole rule-based system.

A practical application of the TS systems with the strong triangular partition will be exemplified below.

Example 5.26. A navigation system for a mobile robot usually combines obstacle avoidance and goal-seeking behaviors [11]. It can be designed using various approaches like artificial neural networks, genetic algorithms, potential field methods, fuzzy logic or fuzzy-neural systems [11], [16], [47], [170], [197], [199].

Let us consider a simple mobile robot, like Khepera manufactured by K-Team [142], working in an unknown environment. The robot is equipped with two wheels moved by two independent motors coupled with gears and infra-red (IR) proximity sensors, placed as shown in Fig. 5.22. Our goal is to show how to design a simple but efficient sensor-based navigation system for this robot. Based on expert experience and fuzzy logic we will design two independent P1-TS systems as the most important components of the navigation system. In contrast to existing approaches based on fuzzy logic for mobile robot control system design, our method will result in delivering explicit formulas for a fuzzy navigation system.

We assume that the wheels do not skid or float. The robot's kinematics is described by the following differential equations

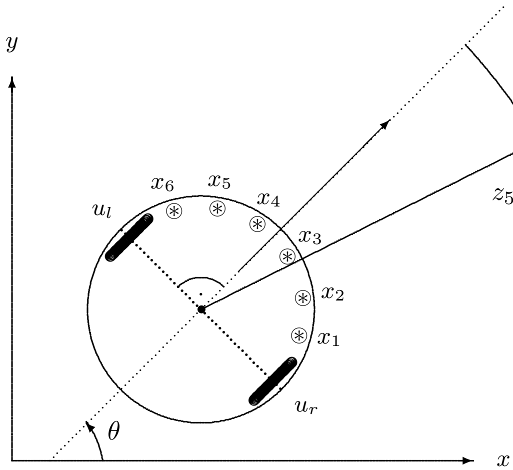


Fig. 5.22 Schematic diagram of a two-wheeled robot for the motion control (⊛ - IR sensor)

$$\dot{\theta} = \frac{1}{2R}\left(u_r - u_l\right), \tag{5.163}$$

$$\dot{x} = \frac{1}{2}\left(u_r + u_l\right)\cos\theta, \tag{5.164}$$

$$\dot{y} = \frac{1}{2}\left(u_r + u_l\right)\sin\theta, \tag{5.165}$$

where x and y are the position of the mobile robot on the ground, θ is the attitude of the robot, R is the displacement from the center of the robot to the center of the wheel, u_l and u_r are control actions for the left- and right-wheeled motor, respectively. In the above equations $\left(u_r + u_l\right)/2$ is the translational (tangential) velocity of the robot and $\left(u_r - u_l\right)/\left(2R\right)$ is its angular velocity.

Obstacle avoidance is one of the most important tasks that the mobile robot should perform independently of the other tasks such as goal-seeking, transporting objects, etc. The control algorithm has to prevent damage to the robot when it moves in an unknown environment. When a robot navigates in an uncertain environment towards the goal position, the two behaviors usually are in conflict with each to other. The navigator should combine obstacle avoidance and goal-seeking behaviors. To do this, we propose to design the behaviors independently and combine them by a soft switching function according to the situation around the robot. The architecture of the navigation system is shown in Fig. 5.23. Every wheel is moved by a DC motor coupled with a gear. We assume that DC motors are controlled by PID controllers working at the lowest control level. The inputs of the controllers are signals $u_{(\cdot)}$ as the number of pulses per second for two wheels; $u_l \in [-C, C]$

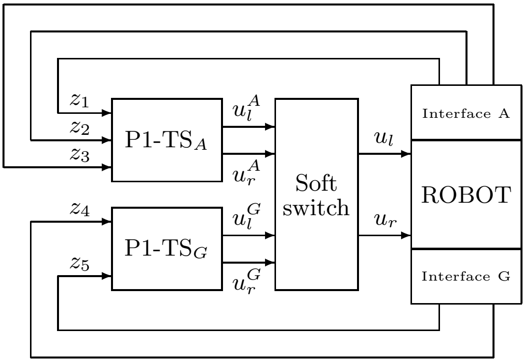


Fig. 5.23 Navigation system for the mobile robot from Fig. 5.22

for the left wheel and $u_r \in [-C, C]$ for the right wheel, where C is the maximal value of pulses per second. Thus, the value of $u_{(\cdot)} = C$ corresponds to the maximal forward wheel speed, whereas $u_{(\cdot)} = -C$ corresponds to the maximal backward wheel speed.

Now we will investigate the obstacle avoidance mode. The sensors x_1, ..., x_6 are placed around the robot and positioned as shown in Fig. 5.22. Every sensor embeds an infra-red emitter and receiver and is coupled with an A/D converter contained in the interface A. It delivers information $x_i \in \{0, 1, 2, \ldots, a\}$, $(i = 1, \ldots, 6)$, about the distance between the robot and an obstacle, where $a = 2^k - 1$, if k-bit A/D converter is used. The signal x_i is a decreasing function of the distance between an obstacle and the robot. The maximal value $x_i = a$ indicates the most dangerous situation, whereas $x_i = 0$ says that there is no "visible" obstacle in the selected direction. In order to reduce the number of rules we define the following three variables for the P1-TS$_A$ system (see Figs. 5.22 and 5.23):

$$z_1 = \max(x_1, x_2), \tag{5.166}$$

$$z_2 = \max(x_5, x_6), \tag{5.167}$$

$$z_3 = \max(x_3, x_4), \tag{5.168}$$

The fuzzy sets for the input variables z_1, z_2 and z_3 for the P1-TS$_A$ system have the following meanings (see Fig. 2.8 in the case $[-\alpha, \beta] = [0, a]$):

N_1 - there *is no* obstacle on the *right*-hand side,

P_1 - there *is* an obstacle on the *right*-hand side,

N_2 - there *is no* obstacle on the *left*-hand side,

P_2 - there *is* an obstacle on the *left*-hand side,

N_3 - there *is no* obstacle on the *front*,

P_3 - there *is* an obstacle on the *front*.

Table 5.14 Look-up-table for the P1-TS$_A$ fuzzy system for the robot working in the obstacle avoidance mode (A)

Rule	z_1 (right)	z_2 (left)	z_3 (front)	Decision	$\left(u_l^A, u_r^A\right)$
R_1	N_1	N_2	N_3	*go ahead*	(C, C)
R_2	P_1	N_2	N_3	*turn left*	$(-C, C)$
R_3	N_1	P_2	N_3	*turn right*	$(C, -C)$
R_4	P_1	P_2	N_3	*go ahead*	(C, C)
R_5	N_1	N_2	P_3	*turn left*	$(-C, C)$
R_6	P_1	N_2	P_3	*turn left*	$(-C, C)$
R_7	N_1	P_2	P_3	*turn right*	$(C, -C)$
R_8	P_1	P_2	P_3	*turn left*	$(-C, C)$

By formulating the fuzzy rules for the obstacle avoidance mode, which should be performed by the P1-TS$_A$ system, the following decisions can be taken: "*go ahead*" or "*turn left*" or "*turn right*". Since every wheel is moved by a low-level controller, every decision corresponds to the control action as a vector containing two components $\left(u_l^A, u_r^A\right) \in \{-C, C\}^2$ as the number of pulses per second for the left and right wheel, respectively. The P1-TS$_A$ system has three inputs $z_1 \in [0, a]$ (right), $z_2 \in [0, a]$ (left) and $z_3 \in [0, a]$ (front) and two outputs $u_l^A \in [-C, C]$ and $u_r^A \in [-C, C]$.

The fuzzy rules are given in Table 5.14. They are defined intuitively and seem to be rather obvious. For example the rule R_2 says that "*If* there is an obstacle on the right-hand side *and* no obstacle on the left-hand side *and* no obstacle on the front, *then* turn left". By $[-\alpha_k, \beta_k] = [0, a]$ for $k = 1, 2, 3$, according to (5.21)-(5.29) we immediately obtain the following control actions:

$$u_l^A = \frac{C}{a^3}\left(a^3 - 2a^2 z_1 - 2a^2 z_3 + 2a z_1 z_2 + 2a z_1 z_3 + 2a z_2 z_3 - 4 z_1 z_2 z_3\right), \tag{5.169}$$

$$u_r^A = \frac{C}{a^2}\left(a^2 - 2a z_2 + 2 z_1 z_2\right), \tag{5.170}$$

with z_1, z_2 and z_3 given by (5.166)-(5.168).

Now let us consider the goal seeking mode. For the P1-TS$_G$ rule-based system responsible for correct robot behavior in the goal seeking mode, we define two inputs z_4 and z_5. The variable $z_4 \in [-\pi, \pi]$ is the angle between the line perpendicular to the robot axle and a distance line between the robot and the goal point, whereas $z_5 \in [0, D]$ is the distance between the robot and the goal point (x_G, y_G). The fuzzy sets for the inputs z_4 and z_5 have the following meanings (see Fig. 5.24):

A_1 - the angle z_4 is *negative*,
A_2 - the angle z_4 is *near zero*,

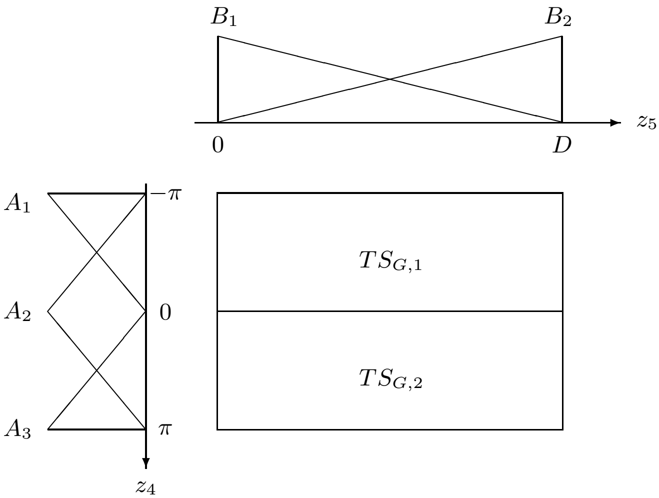


Fig. 5.24 Triangular partition for the P1-TS$_G$ system responsible for the goal-seeking mode

A_3 - the angle z_4 is *positive*,
B_1 - the distance z_5 is *small*,
B_2 - the distance z_5 is *big*.

The fuzzy rules for the P1-TS$_G$ system are given in Table 5.15. They are as transparent as the ones of the obstacle avoidance mode. The rule R_2 says that "If the angle z_4 is *near zero* and the distance z_5 is *small*, then the robot should *go ahead slowly*". This can be achieved by a sufficiently small coefficient $\eta \in (0,1)$, say $\eta = 0.05$. After the partition of the input domain $[-\pi,\pi] \times [0,D]$ as in Fig. 5.24 we obtain two P1-TS subsystems. Without going into details, according to (5.1)-(5.5) by taking the appropriate values for α_k, β_k, $(k = 1,2)$ and q_v, $(v = 1,2,3,4)$, we obtain the following control actions for every point $(z_4, z_5) \in [-\pi,\pi] \times [0,D]$

Table 5.15 Look-up-table for the P1-TS$_G$ fuzzy system for the robot working in the goal seeking mode (G)

Rule	z_4 (angle)	z_5 (distance)	Decision	$\left(u_l^G, u_r^G\right)$
R_1	A_1	B_1	*turn left*	$(-C, C)$
R_2	A_2	B_1	*go ahead slowly*	$(\eta C, \eta C)$
R_3	A_3	B_1	*turn right*	$(C, -C)$
R_4	A_1	B_2	*turn left*	$(-C, C)$
R_5	A_2	B_2	*go ahead*	(C, C)
R_6	A_3	B_2	*turn right*	$(C, -C)$

$$u_l^G = C\left(\eta + \frac{z_4 - \eta\,|z_4|}{\pi} + \frac{1-\eta}{D}z_5 - \frac{1-\eta}{\pi D}\,|z_4|\,z_5\right), \tag{5.171}$$

$$u_r^G = C\left(\eta - \frac{z_4 + \eta\,|z_4|}{\pi} + \frac{1-\eta}{D}z_5 - \frac{1-\eta}{\pi D}\,|z_4|\,z_5\right), \tag{5.172}$$

for the navigation system working in the goal seeking mode. By substituting

$$z_4 = \theta - \arctan\left(\frac{y_G - y}{x_G - x}\right), \qquad z_5 = \sqrt{(x_G - x)^2 + (y_G - y)^2}, \tag{5.173}$$

one obtains explicitly the control signals u_l^G and u_r^G as nonlinear feedback depending on the robot position.

Finally, the two behaviors will be combined. When the mobile robot navigates in an unknown environment, one of these behaviors must be selected at each action step in order to accomplish its goal. Once the rule bases for the P1-TS systems are gathered, the two behaviors can be combined as follows

$$u_l = \rho u_l^A + (1-\rho)\,u_l^G, \qquad u_r = \rho u_r^A + (1-\rho)\,u_r^G, \tag{5.174}$$

where the coefficient $\rho \in [0,1]$ is a constant or $\rho = \max(x_1, \ldots, x_6)/a$.

A series of simulations and experiments was successfully performed using a small Khepera robot [142] to check the effectiveness of the proposed navigation system, under various obstacle configuration, start and goal positions of the robot, initial heading angles, and by choosing various design parameters

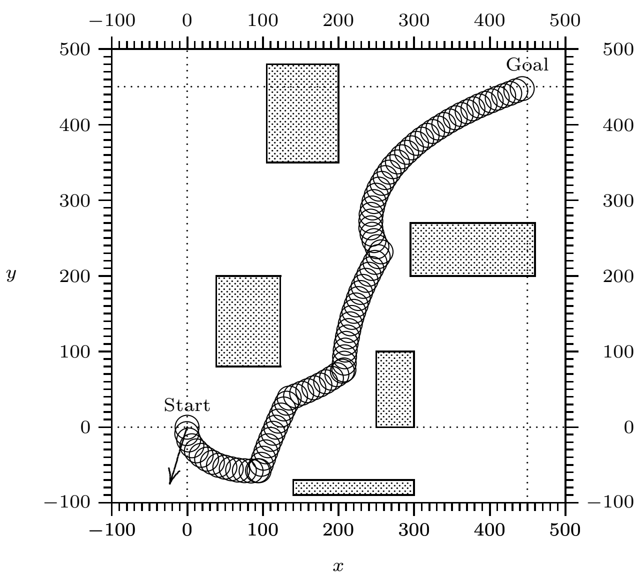


Fig. 5.25 Trajectory of the mobile robot from Example 5.26

ρ and η. Fig. 5.25 shows an example of the mobile robot trajectory for some configuration of the obstacles for $R = 26$ [mm], $C = 10$ [1 / sec], $\eta = 0.05$, $D = 500$ [mm], $\rho = 0.5$, the initial robot position $(\theta(0), x(0), y(0)) = (-0.6\pi, 0, 0)$ and the goal position $(x_G, y_G) = (450, 450)$. As one can see, the mobile robot located at start position "Start" arrives at the goal position "Goal" without colliding with obstacles.

With the proposed navigation strategy, the robot arrives at the given goal position without colliding with obstacles. Observe that explicitly obtained control actions are simple for simulations. What is more, they enable an easy implementation of the navigation strategies in small inexpensive embedded digital systems. The proposed method can be viewed as an initial step in developing further improvements of the navigation system by learning and adaptation.

In the next two sections we will present simple but useful results concerning linearity of the P1-TS systems and the relationship between the first-order and the zero-order P1-TS systems.

5.9 Linearity Condition for P1-TS Systems

Below we give the necessary and sufficient linearity condition for the P1-TS systems.

Corollary 5.27. *Let f be a linear function*

$$f(\mathbf{z}) = \mathbf{r}^T\mathbf{z}, \qquad \mathbf{r} = [r_1, \ldots, r_n]^T \in \mathbb{R}^n, \quad \mathbf{z} = [z_1, \ldots, z_n]^T \in D^n. \quad (5.175)$$

There exists a P1-TS system with the input vector $\mathbf{z}$ and the output S, such that $S(\mathbf{z}) = f(\mathbf{z})$ for all $\mathbf{z} \in D^n$, if and only if the consequents of the fuzzy rules constitute the following vector $\mathbf{q} = [q_1, \ldots, q_{2^n}]^T$:

$$\mathbf{q} = \mathbf{L}^T\mathbf{r}, \quad \mathbf{L} = [\gamma_1, \ldots, \gamma_{2^n}], \quad \gamma_v \in \varGamma^n, \quad v = 1, \ldots, 2^n, \qquad (5.176)$$

where $\varGamma^n$ is the set of vertices of the hypercuboid D^n. Equivalently

$$q_v = q_{(i_1,\ldots,i_n)} = \sum_{k=1}^{n} \left(i_k(\alpha_k + \beta_k) - \alpha_k\right) r_k, \qquad (5.177)$$

by $v \leftrightarrow (i_1, \ldots, i_n) \in \{0,1\}^n$ according to the bijection (2.16).

Proof is given in Appendix C.3.

An advantage of the above condition is that linearity of a P1-TS system can be immediately recognized by a designer without using a generator or a fundamental matrix.

5.10 The First-Order P1-TS Systems

The first-order TS system is able to perform more complex functions than the zero-order one, since the consequents of the fuzzy rules depend on input variables. We will show that based on results obtained for the zero-order P1-TS system, one can easily obtain new results for the first-order system.

Corollary 5.28. *Consider the first order P1-TS system with the input vector* $\mathbf{z} \in D^n$*, in which the consequents* q_v *of the fuzzy rules depend linearly on the inputs:*

$$q_v = q_{v,0} + [q_{v,1}, q_{v,2}, \cdots, q_{v,n}]\,\mathbf{z}\ , \qquad \textit{for} \quad v = 1, 2, \ldots, 2^n. \tag{5.178}$$

1. *Every such a rule-based system is equivalent to the following multivariate polynomial*

$$f_I(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\mathbf{\Omega}^T\right)^{-1} \mathbf{Q} \begin{bmatrix} 1 \\ \mathbf{z} \end{bmatrix}, \tag{5.179}$$

   where the matrix $\mathbf{Q}$ *contains the coefficients* $q_{v,j}$*,* $(v = 1, 2, \ldots, 2^n,\ j = 0, 1, 2, \ldots, n)$ *as in (5.178)*

$$\mathbf{Q} = \begin{bmatrix} q_{1,0} & q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,0} & q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{2^n,0} & q_{2^n,1} & q_{2^n,2} & \cdots & q_{2^n,n} \end{bmatrix}, \tag{5.180}$$

   and $\mathbf{\Omega}$ *is the fundamental matrix corresponding to the generator* $\mathbf{g}$*.*
2. *For the given multivariate polynomial function given by (5.179) there exist infinitely many first-order P1-TS systems performing this function.*

Proof. The subsequent equations (5.178) can be rewritten in the matrix form

$$\mathbf{q}(\mathbf{z}) = \mathbf{Q} \begin{bmatrix} 1 \\ \mathbf{z} \end{bmatrix}.$$

According to the proof of Theorem 2.4, the crisp system output is $S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \left(\mathbf{\Omega}^T\right)^{-1} \mathbf{q} = f_I(\mathbf{z})$. This ends the proof of the first part of Corollary 5.28. The second part we prove by a counterexample. Consider a first-order P1-TS system with two inputs $z_k \in [-\alpha_k, \beta_k]$ for $k = 1, 2$, for which $\mathbf{\Omega}$ is given by (2.38). Let

$$f_I(\mathbf{z}) = r_0 + r_1 z_1 + r_2 z_2 + r_3 z_1 z_2 + r_4 z_1^2 + r_5 z_2^2 + r_6 z_1^2 z_2 + r_7 z_1 z_2^2. \tag{5.181}$$

If the consequents of the rules for this system are

$$\mathbf{q}(\mathbf{z}) = \mathbf{\Omega}^T \begin{bmatrix} r_0 & a & b \\ r_1 - a & r_4 & c \\ r_2 - b & d & r_5 \\ r_3 - c - d & r_6 & r_7 \end{bmatrix} \begin{bmatrix} 1 \\ z_1 \\ z_2 \end{bmatrix},$$

then such rule-based system is equivalent to the function $f_I(\mathbf{z})$ in (5.181) for any values a, b, c and d. This ends the proof of the second part of Corollary 5.28. □

Example 5.29. The system of rules from Example 5.14 which define the controller with variable gains, can be viewed as a special case of the first-order P1-TS system. We will consider it again in the context of Corollary 5.28. One can choose the system generator in many ways. Let us define the following one

$$\mathbf{g}(\mathbf{z}) = [1,\ z_1,\ z_2,\ z_3,\ z_1 z_2,\ z_1 z_3,\ z_2 z_3,\ z_1 z_2 z_3]^T, \tag{5.182}$$

which corresponds to the sequence of rules R_1, R_5, R_3, R_2, R_7, R_6, R_4, R_8 formulated in Example 5.14. Thus, the vector of consequents of the rules is as follows

$$\mathbf{q}(\mathbf{z}) = \begin{bmatrix} 0 & a_1 k_1 k_8 & a_2 k_1 k_8 & a_3 k_1 k_8 \\ 0 & a_1 k_1 k_4 & a_2 k_1 k_4 & a_3 k_1 k_4 \\ 0 & a_1 k_1 k_6 & a_2 k_1 k_6 & a_3 k_1 k_6 \\ 0 & a_1 k_1 k_7 & a_2 k_1 k_7 & a_3 k_1 k_7 \\ 0 & a_1 k_1 k_2 & a_2 k_1 k_2 & a_3 k_1 k_2 \\ 0 & a_1 k_1 k_3 & a_2 k_1 k_3 & a_3 k_1 k_3 \\ 0 & a_1 k_1 k_5 & a_2 k_1 k_5 & a_3 k_1 k_5 \\ 0 & a_1 k_1 & a_2 k_1 & a_3 k_1 \end{bmatrix} \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix}.$$

For the generator (5.182), assuming $\beta_k = \lambda_k \alpha_k$ for $k = 1, 2, 3$ as in Example 5.14, we compute the fundamental matrix $\mathbf{\Omega}$. According to (5.180) after computations we obtain the function (5.179) as follows

$$S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \begin{bmatrix} 0 & a_1 b_1 & a_2 b_1 & a_3 b_1 \\ 0 & a_1 b_2 & a_2 b_2 & a_3 b_2 \\ 0 & a_1 b_3 & a_2 b_3 & a_3 b_3 \\ 0 & a_1 b_4 & a_2 b_4 & a_3 b_4 \\ 0 & a_1 b_5 & a_2 b_5 & a_3 b_5 \\ 0 & a_1 b_6 & a_2 b_6 & a_3 b_6 \\ 0 & a_1 b_7 & a_2 b_7 & a_3 b_7 \\ 0 & a_1 b_8 & a_2 b_8 & a_3 b_8 \end{bmatrix} \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix}. \tag{5.183}$$

One can check that all coefficients b_1, b_2, ..., b_8 contained in the matrix $\mathbf{Q}$ given in (5.183) are exactly the same as formerly obtained in Example 5.14.

Corollary 5.28 can be helpful for the design and analysis of the first-order P1-TS systems. From the designer point of view it seems to be interesting to realize that for a given crisp function from the class of allowable functions (5.179), the process of the fuzzy rules derivation for the first-order P1-TS system can lead to infinitely many solutions.

5.11 Zero-Order TS System with Contradictory Rule-Base

There are several reasons why a fuzzy expert system is not "certain". It may contain *contradictory* rules in the rule-base. According to Section 2.5 two rules are *contradictory* to each other if their consequents are different for the same antecedent. In such a system the rules are of the form

$$\begin{aligned}
&\text{If } P_v,\ \text{then } S = \tilde{q}_v = q_v + \epsilon_v,\\
&\text{If } P_v,\ \text{then } S = \tilde{q}_{v+1} = q_{v+1} + \epsilon_{v+1},\\
&\qquad\vdots\\
&\text{If } P_v,\ \text{then } S = \tilde{q}_{v+k} = q_{v+k} + \epsilon_{v+k},\\
&\qquad \tilde{q}_i \neq \tilde{q}_j, \quad \text{for} \quad i \neq j,
\end{aligned} \tag{5.184}$$

for $v \in \{1, \ldots, 2^n\}$ and ϵ_v is viewed as an "error". The problem is how to obtain the components θ_0, θ_i, $\theta_{i,j}$, ..., $\theta_{1,2,\ldots,n}$ of the vector $\boldsymbol{\theta}$ for the TS system (2.26). Because no additional information on the preferences of values $\tilde{q}_v$ is given, we assume that the consequents of the rules are contaminated by noise with zero mean and some variance σ^2. Now, instead of (2.29) and (2.30), we obtain m boundary conditions

$$\begin{aligned}
S(\boldsymbol{\gamma}_1) &= \boldsymbol{\theta}^T \mathbf{g}(\boldsymbol{\gamma}_1) = \tilde{q}_1 = q_1 + \epsilon_1,\\
&\ \ \vdots\\
S(\boldsymbol{\gamma}_m) &= \boldsymbol{\theta}^T \mathbf{g}(\boldsymbol{\gamma}_m) = \tilde{q}_m = q_m + \epsilon_m, \qquad 2^n < m.
\end{aligned}$$

Using a vector notation $\tilde{\mathbf{q}} = [\tilde{q}_1, \ldots, \tilde{q}_m]^T$ and $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_m]^T$ we have the same equations but in the matrix form

$$\begin{aligned}
\boldsymbol{\epsilon} &= \boldsymbol{\Omega}_e^T \boldsymbol{\theta} - \tilde{\mathbf{q}}\ ,\\
\boldsymbol{\Omega}_e^T &= \left[\mathbf{g}(\boldsymbol{\gamma}_1) \ldots \mathbf{g}(\boldsymbol{\gamma}_m)\right]_{2^n \times m}\ .
\end{aligned} \tag{5.185}$$

The matrix $\boldsymbol{\Omega}_e$ defined by (5.185) will be called a *generalized fundamental matrix* of the P1-TS system. Next, we can easily find such a vector $\tilde{\boldsymbol{\theta}}$ that minimizes the sum of squared errors $\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$

$$\tilde{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{2^n}}{\arg\min} \left\{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}\right\} = \left(\boldsymbol{\Omega}_e^T \boldsymbol{\Omega}_e\right)^{-1} \boldsymbol{\Omega}_e^T \tilde{\mathbf{q}}. \tag{5.186}$$

The unique solution always exists independently of the number of contradictory rules, since

$$\operatorname{rank} \boldsymbol{\Omega}_e = \dim \boldsymbol{\theta}.$$

The system without contradictions is a special case of the rule-based system with contradictions, because of the linguistic interpretation of the fuzzy rules and the problem solution. Observe that we obtain $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$, if there are no contradictions in the rules. This fact seems to be intuitively obvious. However, the vice-versa is not true. To prove this, suppose there are two contradictory rules

$$R_1 : \text{If } P_v, \text{ then } S = q_v + \epsilon,$$

$$R_2 : \text{If } P_v, \text{ then } S = q_v - \epsilon,$$

and we compute $\tilde{\boldsymbol{\theta}}$ according to (5.186). Next, consider the noncontradictory system of rules, in which the corresponding rule is

$$R : \text{If } P_v, \text{ then } S = q_v \ ,$$

and $\boldsymbol{\theta}$ is computed according to (2.30). One can prove that $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$ holds in this case, i.e. both rule-base systems generate the same output.

It should be added that the proposed approach is not unique. Since the consequents in (5.184) may be viewed as intervals, we can obtain another solution, where the vector $\tilde{\boldsymbol{\theta}}$ will contain intervals as its components. However, this interesting problem will not be considered in this book.

Example 5.30. Let us consider a two-input-one-output P1-TS system given by the fuzzy rules

$$\begin{aligned}
&R_1 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = \tilde{q}_1,\\
&R_2 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = \tilde{q}_2,\\
&R_3 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = \tilde{q}_3,\\
&R_4 : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = \tilde{q}_4,\\
&R_5 : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } N_2, \text{ then } S = \tilde{q}_5,\\
&R_6 : \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2, \text{ then } S = \tilde{q}_6,
\end{aligned}$$

where the rules R_1 and R_2, as well as R_3 and R_4 are contradictory ones by $\tilde{q}_1 \neq \tilde{q}_2$ and $\tilde{q}_3 \neq \tilde{q}_4$. One can check that

$$\boldsymbol{\Omega}_e^T = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 \\ 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 \end{bmatrix}.$$

For the given $\tilde{q}^T = [\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_6]$, the estimated vector $\tilde{\boldsymbol{\theta}}$ is

$$\tilde{\boldsymbol{\theta}}^T = \frac{1}{2\left(\alpha_1+\beta_1\right)\left(\alpha_2+\beta_2\right)}\tilde{\mathbf{q}}^T \begin{bmatrix} \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ 2\alpha_1\beta_2 & 2\beta_2 & -2\alpha_1 & -2 \\ 2\alpha_1\alpha_2 & 2\alpha_2 & 2\alpha_1 & 2 \end{bmatrix}.$$

Observe that for noncontradictory rules, when

$$\tilde{q}_1 = \tilde{q}_2 = q_1, \quad \tilde{q}_3 = \tilde{q}_4 = q_2, \quad \tilde{q}_5 = q_3, \quad \tilde{q}_6 = q_4,$$

the result (5.186) reduces to (2.30) and $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

5.12 Summary

The controller synthesis for a milk of lime blending tank considered in Section 5.1.2 suggests that in some cases of highly nonlinear dynamical processes (stable and with one equilibrium point) one can derive a simple and attractive control algorithm expressed by the fuzzy rules for the P1-TS system. By using continuous multi-valued logic we obtained “soft switching” control signals, as opposed to “hard switching” ones that are typical when Boolean logic is applied. There is an observable connection between the heuristic design methods that use conventional (Boolean) logic and the methods that use fuzzy logic. The method of fuzzy rules synthesis for the P1-TS system as a controller is simple and clear. It resembles a heuristic design procedure for a combinational logic system synthesis. The last one is widely used in practice for switching control algorithms designs, intended for the embedded hardware devices or software components for real-time direct digital control systems, e.g. programmable logic controllers (PLCs).

In Section 5.1.3 we considered a special class of P1-TS systems, in which input vectors are points from the unity hypercube $[0,1]^n$. They were called “logical” systems, since according to Theorem 3.15, the output vectors take values from the unity hypercube, as well. Logical systems process information expressed in continuous multi-valued logic. Their look-up-tables describing “If-then” rules or metarules can be viewed as generalized Karnaugh maps. In the vertices of the unity hypercube the Karnaugh maps enable us to interpret the function to which a given P1-TS system is equivalent. This interpretation coincides with formerly obtained algebraic results. We derived all functions of two variables on the assumption that the consequents of the rules take binary values. Among others the probabilistic t-norm and t-conorm, Reichenbach’s implication and the other functions expressed in the continuous multi-valued logic were obtained.

In Section 5.2.2 we considered highly interpretable rule-bases for systems with three and more inputs for both abstract processes and real dynamical plants. It was shown that by using a systematic approach and matrix computations, the fuzzy rules for discrete-time NARX model considered in [208] and fuzzy J-K flip-flop developed in [60] can be easily obtained. In Example 5.5 we proved equivalence between fuzzy rules and Euler equations for a rigid body. Next, the fuzzy rules for nonlinear dynamical processes such as Chen's attractor, human immunodeficiency virus, magnetic suspension system, low order atmospheric circulation process and induction motor, were derived using symbolic computations. The number of similar models could be substantially increased. General formulas for the P1-TS systems with four and more inputs are not difficult to obtain by using the appropriate software specializing in symbolic computations, e.g. Maple, Mathematica, MuPAD, etc. It was exemplified that thanks to recursive procedures described in the previous sections, the curse of dimensionality problem in the rule-based systems can be substantially reduced. In all cases we should try to obtain a small number of rules. To do this, both P1-TS and P2-TS systems can be used in some cases. Sometimes it is desirable to transform original variables into other ones by using a nonlinear mapping, as was shown in Example 5.10.

In Section 5.3.1 a low order atmospheric circulation model described in literature was considered as P1-TS system with five inputs. Next, we showed that P1-TS system can exactly model the induction motor as a highly nonlinear dynamical fifth-order system. Example in Section 5.3.3 shows that some complicated functions describing e.g. the control algorithms, earlier obtained by using Boolean logic methods, can be immediately transformed into the fuzzy domain by applying generalized operators (t-norms, t-conorms and strong negation). The fuzzy rules obtained in this way, have a clear logical interpretation.

In Section 5.4 theory of P1-TS systems was used for the analytical design of the PID controller working in the closed-loop control system for some class of (linear and nonlinear) second order plant. The controller as P1-TS system works “optimally” with respect to typical requirements formulated for automatic control systems (the closed-loop system is free of oscillations, has no steady state error and its step response is as quick as required). In Section 5.4.3 a PD-like rule-based controller was derived for a nonlinear second order plant. The fuzzy controller behaves optimally for a given reference input. The consequents of the fuzzy rules are time-dependent functions. In this way we showed that the “optimal” fuzzy controller in the closed-loop containing a second order dynamical plant can be analytically obtained. Good performance of the fuzzy control system is independent of the particular values of the parameters of the plant.

In Section 5.5 we showed that the “controller with variable gains” introduced by Ying [205], [206], can be immediately obtained using the facts concerning P1-TS systems.

In Section 5.6 we established an exact relationship between the P1-TS systems and some class of multiaffine dynamical systems, in which the derivatives of state variable are from a subclass of Kolmogorov-Gabor polynomials. The antecedents of the fuzzy rules are concerned with the state variables and the consequents are state derivatives depending linearly on the state variables and the control signal. In contrast to Section 3.4, original Takagi-Sugeno inference method for the P1-TS system was used. Theoretical results were exemplified by exact fuzzy modeling of the van de Vusse reaction and Rössler chaotic system. In Remark 5.22 the class of single-input dynamical systems, which can be perfectly modeled by P1-TS systems was formally defined. The proposed method can be easily extended to MIMO systems.

Section 5.7 describes the architecture of the P1-TS system as the fuzzy model of conventional MIMO linear dynamical system.

In Section 5.8 we showed that the idea of TS systems with two linear membership functions of fuzzy sets can be easily extended to the systems with triangular fuzzy partition. It should be added that the triangular membership functions can be substituted by other nonlinear membership functions which are similar to the triangular ones, i.e. they have the same support and the same monotonicity intervals. For such more general systems one can use formerly obtained results on recursion. As a practical example of using the systems with triangular fuzzy partition, a sensor-based navigation system for a mobile robot was presented. The proposed navigator consists of obstacle avoidance and goal seeking behaviors. These are independently designed to accommodate complex environments and combined by the behavior selector in the form of soft switching function. Although the design process of the navigator was based on expert knowledge, the proposed method can be viewed as an initial step in developing further improvements of the navigation system by learning and adaptation.

This chapter ends with supplementary results for P1-TS systems. The outcomes concern with the necessary and sufficient condition of linearity for such rule-based systems (Section 5.9), the first-order P1-TS systems (Section 5.10) and the zero-order systems with contradictory rule-base (Section 5.11). The advantage of the linearity condition is that linearity of the rule-base can be immediately recognized by a designer without using a generator or a fundamental matrix. Corollary 5.28 can be helpful for the design and analysis of the first-order P1-TS systems. From the designer point of view it seems to be interesting to realize that for a given crisp function from the class of allowable functions, the process of the fuzzy rules derivation for the first-order P1-TS system can lead to infinitely many solutions.

In the last section we showed that the system without contradictions is a special case of the rule-based system with contradictions, because of the linguistic interpretation of the fuzzy rules and the problem solution. We introduced a generalized fundamental matrix of the P1-TS system which reduces to the formerly considered fundamental matrix, if the system of the rules is a noncontradictory one. The generalized fundamental matrix can be easily extended to the P2-TS systems, as well.

The above chapter contains many examples concerning exact fuzzy modeling and control of real systems. In this way it was shown that P1-TS systems deserve a special attention not only from the theoretical point of view, but also they should be attractive for practitioners. The results are analytical and therefore cannot be questioned; they reinforce our belief that many successful applications of the fuzzy rule-based systems (especially fuzzy controllers) cannot be a matter of chance.

Chapter 6
Modeling of Multilinear Dynamical Systems from Experimental Data

Since the introduction of fuzzy sets by Zadeh in 1965 [210], many researchers have shown interest in applying this theory to system identification, which is an essential part of any control system design. Rapid development of intelligent control methodologies such as artificial neural networks, fuzzy logic theory, and rule-based expert systems, have provided alternative tools to tackle the problem of system identification [203]. A large number of fuzzy identification techniques have been developed using neural networks, genetic algorithms, clustering techniques, Kalman filtering and other methods, including ad hoc ones [112]. Consequently, fuzzy identification has become a very important area in fuzzy system theory [180]. The main approaches to fuzzy identification are based on linguistic fuzzy modeling, fuzzy relational equation modeling and Takagi-Sugeno modeling [13]. In this chapter we present a new effective method of modeling continuous multilinear dynamical systems using the Takagi-Sugeno fuzzy expert system.

Most of the current fuzzy identification is carried out in discrete-time domain in contrast to continuous-time domain. Continuous-time models are often desired for the control system design, since the designers prefer if the parameters identified in the model have a direct relationship with the physical and chemical parameters of the plant. Thus, the need for a fuzzy model expressed in the continuous time domain arises [203]. In this chapter we will start with a continuous-time model of a dynamical system. However, for the numerical computing we will convert this model into the discrete-time form.

6.1 Problem Statement

Let us consider a multilinear dynamical system described by the following differential equations of variables $z_1(t), \ldots, z_n(t)$, $(t \geq 0)$, with $n2^n$ real coefficients

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 183–197.
springerlink.com

$$
\left.\begin{aligned}
\frac{dz_1}{dt} &= a_0 + \sum_{i=1}^{n} a_i z_i + \sum_{\substack{i,j=1\\ i<j}}^{n} a_{i,j} z_i z_j + \cdots + a_{1,2,\ldots,n} z_1 z_2 \cdots z_n \ , \\
\frac{dz_2}{dt} &= b_0 + \sum_{i=1}^{n} b_i z_i + \sum_{\substack{i,j=1\\ i<j}}^{n} b_{i,j} z_i z_j + \cdots + b_{1,2,\ldots,n} z_1 z_2 \cdots z_n \ , \\
&\vdots \\
\frac{dz_n}{dt} &= h_0 + \sum_{i=1}^{n} h_i z_i + \sum_{\substack{i,j=1\\ i<j}}^{n} h_{i,j} z_i z_j + \cdots + h_{1,2,\ldots,n} z_1 z_2 \cdots z_n \ .
\end{aligned}\right\} \tag{6.1}
$$

Assume that the trajectory of this system is known at the time instants t_1, t_2, ..., t_{K+1}. The available data can be gathered as shown in Table 6.1.

Table 6.1 Learning data for the multilinear dynamical system (6.1)

Time instant	z_1	z_2	$\cdots$	z_n
t_1	$z_1(t_1)$	$z_2(t_1)$	$\cdots$	$z_n(t_1)$
t_2	$z_1(t_2)$	$z_2(t_2)$	$\cdots$	$z_n(t_2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_{K+1}	$z_1(t_{K+1})$	$z_2(t_{K+1})$	$\cdots$	$z_n(t_{K+1})$

Our goal is to develop an identification algorithm such that for the given data set of samples, one obtains a MIMO P1-TS system which optimally models the dynamical system (6.1). As an optimization criterion we assume a standard quality index in the form of sum of squared errors for every variable z_j that comes from the approximation of the derivatives and a measurement noise. To reduce the influence of these disturbances, the number of samples must be sufficiently large.

6.2 Problem Solution

The set of equations (6.1) is equivalent to

$$
\frac{dz_j}{dt} = \mathbf{g}^T(z_1, \ldots, z_n)\,\boldsymbol{\theta}_j, \qquad \boldsymbol{\theta}_j \in \mathbb{R}^{2^n}, \qquad j = 1, 2, \ldots, n, \tag{6.2}
$$

where $\mathbf{g}(z_1, \ldots, z_n)$ is the generator of some P1-TS system with the input vector $[z_1, \ldots, z_n]^T \in D^n$ and the output vector $[dz_1/dt, \ldots, dz_n/dt]^T$. Using the data from Table 6.1 we will show that (6.1) can be well approximated by the fuzzy rules.

Theorem 6.1. *After using a batch procedure one obtains the P1-TS system, that models equations (6.1) optimally in the sense of minimal sum of squared errors for every input variable z_j, $(j = 1, \ldots, n)$ that is equivalent to the following set of differential equations*

$$\frac{dz_j}{dt} = \mathbf{g}^T\left(z_1, \ldots, z_n\right) \widehat{\boldsymbol{\theta}}_j^* \ , \qquad j = 1, 2, \ldots, n. \tag{6.3}$$

The batch procedure is given by

$$\widehat{\boldsymbol{\theta}}_j^* = \mathbf{G}^{-1} \left[\mathbf{g}_{t_1}, \cdots, \mathbf{g}_{t_K}\right] \mathbf{d}_j \ , \qquad j = 1, 2, \ldots, n, \tag{6.4}$$

where $\mathbf{G}$ *is a matrix assumed to be nonsingular*

$$\mathbf{G} = \sum_{k=1}^{K} \mathbf{g}_{t_k} \mathbf{g}_{t_k}^T \in \mathbb{R}^{2^n \times 2^n}, \tag{6.5}$$

and

- *the vectors* $\mathbf{g}_{t_k}$ *are values of the rule-based system generator in the points* $\mathbf{z}\left(t_k\right)$ *of the trajectory, i.e.*

$$\mathbf{g}_{t_k} = \mathbf{g}\left(z_1\left(t_k\right), \ldots, z_n\left(t_k\right)\right), \qquad k = 1, 2, \ldots, K, \tag{6.6}$$

- *the components of the vector* $\mathbf{d}_j$ *are approximations of the derivatives for the variable* z_j*, which are computed in the subsequent time-intervals* $[t_1, t_2]$*,* $[t_2, t_3]$*, …, and* $[t_K, t_{K+1}]$

$$\mathbf{d}_j = \begin{bmatrix} d_j\left(t_1, t_2\right) \\ d_j\left(t_2, t_3\right) \\ \vdots \\ d_j\left(t_K, t_{K+1}\right) \end{bmatrix} = \begin{bmatrix} \dfrac{z_j\left(t_2\right) - z_j\left(t_1\right)}{t_2 - t_1} \\ \dfrac{z_j\left(t_3\right) - z_j\left(t_2\right)}{t_3 - t_2} \\ \vdots \\ \dfrac{z_j\left(t_{K+1}\right) - z_j\left(t_K\right)}{t_{K+1} - t_K} \end{bmatrix} \in \mathbb{R}^K, \quad j = 1, 2, \ldots, n. \tag{6.7}$$

Proof. Consider the P1-TS system which models the nonlinear dynamical system (6.1) with inputs $z_1 \in [-\alpha_1, \beta_1]$, …, $z_n \in [-\alpha_n, \beta_n]$, and outputs $S_1 = dz_1/dt$, …, $S_n = dz_n/dt$. The fundamental matrix $\boldsymbol{\Omega}$ can be obtained approximately from the data, since

$$\alpha_k = -\min_{t \in \{t_1, \ldots, t_K\}} z_k\left(t\right), \qquad \beta_k = \max_{t \in \{t_1, \ldots, t_K\}} z_k\left(t\right), \qquad k = 1, \ldots, n. \tag{6.8}$$

Thus, in reality $\boldsymbol{\Omega}$ is not exactly known, since the data come from observation (measurements). If the matrix $\boldsymbol{\Theta}$ contains coefficients of the crisp model, i.e. $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n]$, then according to Theorem 2.10, equation $\boldsymbol{\Theta} = \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{Q}$ holds. In the ideal case of fuzzy modeling described in the previous sections (when $\boldsymbol{\Omega}$ and $\mathbf{Q}$ are given precisely), we have

$$\frac{d}{dt}\begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \mathbf{g}^T(z_1(t), \ldots, z_n(t)) \cdot \underbrace{\left(\mathbf{\Omega}^T\right)^{-1}\mathbf{Q}}_{\mathbf{\Theta}} . \tag{6.9}$$

where the matrix $\mathbf{Q}$ contains the consequents of the rules. Equivalently (6.9) can be written as

$$\frac{dz_j(t)}{dt} = \mathbf{w}^T(t)\,\mathbf{q}_j , \qquad j = 1, \ldots, n, \quad \forall\, t \geq 0, \tag{6.10}$$

where $\mathbf{q}_j$ is jth column of $\mathbf{Q}$, and

$$\mathbf{w}^T(t) = \mathbf{g}^T(z_1(t), \ldots, z_n(t))\left(\mathbf{\Omega}^T\right)^{-1}, \qquad \forall\, t \geq 0. \tag{6.11}$$

The derivative of the continuous signal $z_j(t)$ can be approximated as

$$\frac{dz_j(t_k)}{dt} = \frac{z_j(t_{k+1}) - z_j(t_k)}{t_{k+1} - t_k} + \epsilon_j(t_k, t_{k+1}), \tag{6.12}$$

where $\epsilon_j(t_k, t_{k+1})$ is an error. The left-hand side of (6.12) can be modeled exactly by the P1-TS system, whereas the Euler approximation of the derivative can be obtained from observation data. In the case of an "ideal" data set, if there is no measurement noise or disturbances, no quantization errors, etc., ϵ_j vanishes by the vanishing sampling period

$$\lim_{t_{k+1}\to t_k} \epsilon_j(t_k, t_{k+1}) = 0, \qquad j = 1, \ldots, n. \tag{6.13}$$

For real experimental data, even though the sampling period is very small, the above condition is satisfied only in the sense of mean value, since the numbers $z_j(t_k)$ come from measurements. On the other hand from (6.10) and (6.12) for $t = t_k$ we have

$$\epsilon_j(t_k, t_{k+1}) = \mathbf{w}^T(t_k)\,\mathbf{q}_j - d_j(t_k, t_{k+1}), \tag{6.14}$$

where $d_j(t_k, t_{k+1})$ is defined in (6.7).

We should minimize the sum of squared errors with respect to every j-th input of the P1-TS system

$$E_j = \sum_{k=1}^{K} \epsilon_j^2(t_k, t_{k+1}) = \|\boldsymbol{\epsilon}_j\|^2 = \left\|\mathbf{W}\mathbf{q}_j - \mathbf{d}_j\right\|^2, \tag{6.15}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}^T(t_1) \\ \vdots \\ \mathbf{w}^T(t_K) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{t_1}^T \\ \vdots \\ \mathbf{g}_{t_K}^T \end{bmatrix}\left(\mathbf{\Omega}^T\right)^{-1}, \qquad \mathbf{W} \in \mathbb{R}^{K\times 2^n}, \tag{6.16}$$

and K-dimensional vector $\mathbf{d}_j$ is given by (6.7). Thus, we should find such a vector of the rules consequents $\mathbf{q}_j = \mathbf{q}_j^*$ for which the sum of squared errors (6.15) is minimal

$$\mathbf{q}_j^* = \arg \min_{\mathbf{q}_j \in \mathbb{R}^K} E_j \ .$$

One can find a necessary condition for E_j to be minimal by setting the gradient of E_j, with respect to $\mathbf{q}_j$, to zero vector

$$\nabla_{\mathbf{q}_j} \left\| \boldsymbol{\epsilon}_j \right\|^2 = -2\mathbf{W}^T \mathbf{d}_j + 2\mathbf{W}^T \mathbf{W} \mathbf{q}_j = \mathbf{0} \ .$$

In the least-squares sense, the optimal vector of consequents of the fuzzy rules results from the *normal equation* $\mathbf{W}^T \mathbf{W} \mathbf{q}_j = \mathbf{W}^T \mathbf{d}_j$, i.e.

$$\mathbf{q}_j^* = \mathbf{W}^+ \mathbf{d}_j \ , \tag{6.17}$$

where in general $\mathbf{W}^+ = \left(\mathbf{W}^T \mathbf{W}\right)^{-1} \mathbf{W}^T$ is the *Moore-Penrose generalized inverse* (pseudoinverse) [15], which always exists. It is assumed that the matrix $\mathbf{W}^T \mathbf{W}$ is a nonsingular one. Let us continue the transformation of the equation (6.17), by taking into account the form of the matrix $\mathbf{W}$ defined in (6.16):

$$\mathbf{q}_j^* = \left(\left(\begin{bmatrix} \mathbf{g}_{t_1}^T \\ \vdots \\ \mathbf{g}_{t_K}^T \end{bmatrix} \left(\boldsymbol{\Omega}^T\right)^{-1} \right)^T \begin{bmatrix} \mathbf{g}_{t_1}^T \\ \vdots \\ \mathbf{g}_{t_K}^T \end{bmatrix} \left(\boldsymbol{\Omega}^T\right)^{-1} \right)^{-1} \left(\begin{bmatrix} \mathbf{g}_{t_1}^T \\ \vdots \\ \mathbf{g}_{t_K}^T \end{bmatrix} \left(\boldsymbol{\Omega}^T\right)^{-1} \right)^T \mathbf{d}_j \ ,$$

where $\mathbf{g}_{t_k} = \mathbf{g}\left(z_1\left(t_k\right), \ldots, z_n\left(t_k\right)\right)$ and $(K+1)$ is the number of samples. After elementary matrix calculations we obtain

$$\mathbf{q}_j^* = \boldsymbol{\Omega}^T \left(\sum_{k=1}^{K} \mathbf{g}_{t_k} \mathbf{g}_{t_k}^T \right)^{-1} \left[\mathbf{g}_{t_1}, \cdots, \mathbf{g}_{t_K}\right] \mathbf{d}_j \ , \qquad j = 1, \ldots, n,$$

Observe that $\mathbf{g}_{t_k} \mathbf{g}_{t_k}^T = \mathbf{g}_{t_k} \otimes \mathbf{g}_{t_k}^T$ for any k is $(2^n \times 2^n)$ matrix as the outer product of the vector $\mathbf{g}_{t_k}$ with itself. The inverse of $\mathbf{G} = \sum_{k=1}^{K} \mathbf{g}_{t_k} \mathbf{g}_{t_k}^T$ exists, since the matrix $\mathbf{W}^T \mathbf{W}$ is nonsingular.

The j-th output of the rule-based system is $S_j = dz_j/dt$ and the equation $S_j = \mathbf{g}^T\left(z_1, \ldots, z_n\right) \cdot \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{q}_j$ follows from the fuzzy rules. On the other hand, $S_j = \mathbf{g}^T\left(z_1, \ldots, z_n\right) \cdot \widehat{\boldsymbol{\theta}}_j$ from the fact that this system is equivalent to a multilinear function of variables $z_1, \ldots, z_n$. Substituting $\mathbf{q}_j$ by $\mathbf{q}_j^*$ we obtain a P1-TS system which is equivalent to

$$\frac{dz_j}{dt} = \mathbf{g}^T\left(z_1, \ldots, z_n\right) \left(\boldsymbol{\Omega}^T\right)^{-1} \mathbf{q}_j^* = \mathbf{g}^T\left(z_1, \ldots, z_n\right) \widehat{\boldsymbol{\theta}}_j^*.$$

In the case when the P1-TS system would ideally approximate the dynamical system described by (6.1), the vector $\boldsymbol{\theta}_j = \left(\boldsymbol{\Omega}^T\right)^{-1}\mathbf{q}_j$. This means that the best approximation vector $\widehat{\boldsymbol{\theta}}_j$ is given by

$$\widehat{\boldsymbol{\theta}}_j^* = \left(\boldsymbol{\Omega}^T\right)^{-1}\mathbf{q}_j^* = \left(\sum_{k=1}^{K}\mathbf{g}_{t_k}\mathbf{g}_{t_k}^T\right)^{-1}\left[\mathbf{g}_{t_1},\cdots,\mathbf{g}_{t_K}\right]\mathbf{d}_j \ .$$

This ends the proof of Theorem 6.1. □

From the given data set which is usually obtained from observation (measurements) of an unknown multilinear dynamical system, one can find the best model in the form of fuzzy rules. The data are almost always corrupted with noise, quantization and the approximation method of the derivatives calculus. The unique minimal sum of error squares can be easily determined by $\mathbf{q}_j^*$ from (6.17)

$$E_j\left(\mathbf{q}_j^*\right) = \left(\mathbf{d}_j - \mathbf{W}\mathbf{q}_j^*\right)^T\left(\mathbf{d}_j - \mathbf{W}\mathbf{q}_j^*\right) = \mathbf{d}_j^T\left(\mathbf{d}_j - \mathbf{W}\mathbf{q}_j^*\right). \tag{6.18}$$

From (6.18) without noise we obtain

$$E_j\left(\mathbf{q}_j^*\right) = \mathbf{d}_j^T\left(\mathbf{I} - \mathbf{W}\left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\right)\mathbf{d}_j = 0.$$

The procedure given by Theorem 6.1 uses all the data at once and therefore can be called a *batch* (off-line, explicit, one-shot) *method*. With respect to computing time, the critical part of this procedure is the inversion of the symmetric matrix $\mathbf{G}$ in (6.5), since it is of the same dimensions as the fundamental matrix.

6.3 Analytical Solution for Dynamical Systems with Two Variables

In this section we will show an analytical solution to the identification problem for the following dynamical system

$$\left.\begin{array}{ll}\dot{x}(t) = \theta_0 + \theta_1 x(t) + \theta_2 y(t) + \theta_3 x(t)\,y(t), & \\ \dot{y}(t) = \vartheta_0 + \vartheta_1 x(t) + \vartheta_2 y(t) + \vartheta_3 x(t)\,y(t), & t \geq t_1 \geq 0.\end{array}\right\} \tag{6.19}$$

The system trajectory $(x(t_k), y(t_k)) = (x_k, y_k)$ for $k = 1, 2, \ldots K+1$ is assumed to be known as a result of observations. For the sake of simplicity we take a constant sampling period; $t_{k+1} - t_k = T$, $(k = 1, 2, \ldots K+1)$. The result will be given in the form of Theorem.

Theorem 6.2. *The inputs and outputs of the P1-TS system are* $[x(t), y(t)]^T \in D^2$ *and* $[\dot{x}(t), \dot{y}(t)]^T$*, respectively. For any collection of coefficients* θ_i *and*

ϑ_i, ($i = 0,1,2,3$), the above system is optimally modeled (in the sense of minimal squared sum of errors) by the following fuzzy rules

$$\text{If } [x(t), y(t)] \text{ is } \begin{bmatrix} N_1 & N_2 \\ P_1 & N_2 \\ N_1 & P_2 \\ P_1 & P_2 \end{bmatrix}, \text{ then } [\dot{x}(t), \dot{y}(t)] \text{ is } \widehat{\mathbf{Q}}^* \tag{6.20}$$

where the matrix of consequents

$$\widehat{\mathbf{Q}}^* = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 \end{bmatrix} \widehat{\mathbf{\Theta}}^*, \tag{6.21}$$

with the boundaries of the rectangle D^2 given by (6.8) for $z_1(t_k) = x_k$ and $z_2(t_k) = y_k$, ($k = 1,2$). The matrix containing approximate coefficients of the right-hand sides of (6.19)

$$\widehat{\mathbf{\Theta}}^* = \mathbf{G}^{-1}\mathbf{E}\ , \tag{6.22}$$

by

$$\mathbf{G} = \begin{bmatrix} K & \sum_{k=1}^K x_k & \sum_{k=1}^K y_k & \sum_{k=1}^K x_k y_k \\ \sum_{k=1}^K x_k & \sum_{k=1}^K x_k^2 & \sum_{k=1}^K x_k y_k & \sum_{k=1}^K x_k^2 y_k \\ \sum_{k=1}^K y_k & \sum_{k=1}^K x_k y_k & \sum_{k=1}^K y_k^2 & \sum_{k=1}^K x_k y_k^2 \\ \sum_{k=1}^K x_k y_k & \sum_{k=1}^K x_k^2 y_k & \sum_{k=1}^K x_k y_k^2 & \sum_{k=1}^K x_k^2 y_k^2 \end{bmatrix}, \tag{6.23}$$

and

$$\mathbf{E} = \frac{1}{T} \begin{bmatrix} x_{K+1} - x_1 & y_{K+1} - y_1 \\ \sum_{k=1}^K (x_{k+1} - x_k)\, x_k & \sum_{k=1}^K (y_{k+1} - y_k)\, x_k \\ \sum_{k=1}^K (x_{k+1} - x_k)\, y_k & \sum_{k=1}^K (y_{k+1} - y_k)\, y_k \\ \sum_{k=1}^K (x_{k+1} - x_k)\, x_k y_k & \sum_{k=1}^K (y_{k+1} - y_k)\, x_k y_k \end{bmatrix}. \tag{6.24}$$

The number of samples must be $K \geq 2^n = 4$.

Proof. According to Theorem 2.10 for the fundamental matrix $\mathbf{\Omega}$ for $n = 2$ the equation (6.21) holds, since $\widehat{\mathbf{Q}}^* = \mathbf{\Omega}^T \widehat{\mathbf{\Theta}}^*$. According to equations (6.5)-(6.6) from Theorem 6.1 and the generator $\mathbf{g}(x,y) = [1, x, y, xy]^T$ one obtains

$$\mathbf{G} = \sum_{k=1}^{K} \begin{bmatrix} 1 \\ x_k \\ y_k \\ x_k y_k \end{bmatrix} [1,\ x_k,\ y_k,\ x_k y_k],$$

i.e. **G** is the same as in (6.23). From (6.4) and (6.7) we have

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_K \\ y_1 & y_2 & \cdots & y_K \\ x_1 y_1 & x_2 y_2 & \cdots & x_K y_K \end{bmatrix} \begin{bmatrix} (x_2 - x_1)/T & (y_2 - y_1)/T \\ (x_3 - x_2)/T & (y_3 - y_2)/T \\ \vdots & \vdots \\ (x_{K+1} - x_K)/T & (y_{K+1} - y_K)/T \end{bmatrix},$$

and it is the same as in (6.24). Checking that $\det \mathbf{G} = 0$ for $K \le 3$ is left to the reader. This ends the proof of Theorem 6.2. □

Now we consider numerical examples.

Example 6.3. The dynamical system (6.19) is described by

$$\left.\begin{aligned} \dot{x}(t) &= y(t), \\ \dot{y}(t) &= -6x(t) - 2y(t) - 4x(t)\,y(t). \end{aligned}\right\} \tag{6.25}$$

For initial conditions $(x(0), y(0)) = (-1, -2)$ and time interval $[0, 5]$, the data come from the solution of (6.25) shown in Fig. 6.1. The differential equations were integrated by the `ode45` solver from Matlab, which uses the (explicit) fourth order Runge-Kutta-Fehlberg method with 10^{-13} relative error tolerance (`RelTol`) and the same absolute error tolerance (`AbsTol`). Since

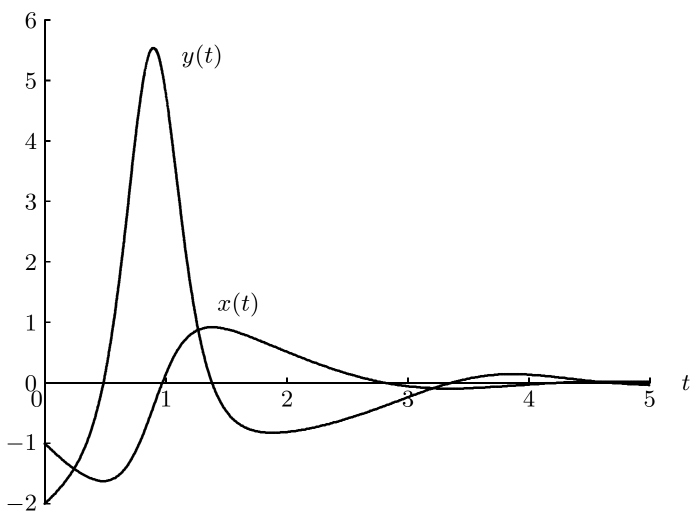


Fig. 6.1 Solution of the differential equations (6.25)

the integration step size $(t_{k+1} - t_k)$ in Matlab solver is not constant, we take into account general formulas from Theorem 6.1. In this example the errors result from simple Euler approximation of the derivatives, numerical solution of the differential equations and rounding off the numerical outcomes. If $[x(t), y(t)]^T$ and $[\dot{x}(t), \dot{y}(t)]^T$ are inputs and outputs of the zero-order P1-TS system, respectively, then equations (6.25) are modeled by the fuzzy rules

$$\text{If } [x, y] \text{ is } \begin{bmatrix} N_1 & N_2 \\ P_1 & N_2 \\ N_1 & P_2 \\ P_1 & P_2 \end{bmatrix}, \text{ then } [\dot{x}(t), \dot{y}(t)] \text{ is } \mathbf{Q}.$$

From the data describing the system trajectory we obtain (see Fig. 6.1):

$$\alpha_1 = -\min_{t\in\{t_1,\ldots,t_K\}} x(t) = 1.6219, \qquad \beta_1 = \max_{t\in\{t_1,\ldots,t_K\}} x(t) = 0.92133,$$

$$\alpha_2 = -\min_{t\in\{t_1,\ldots,t_K\}} y(t) = 2.0000, \qquad \beta_2 = \max_{t\in\{t_1,\ldots,t_K\}} y(t) = 5.5372.$$

For the data obtained numerically by $K = 5928$, according to (6.5) the outcomes are as follows

$$\mathbf{G} = \begin{bmatrix} 5928.0 & -636.15 & 3600.90 & -1281.52 \\ -636.15 & 4097.58 & -1281.52 & 1308.30 \\ 3600.90 & -1281.52 & 24896.30 & -9583.47 \\ -1281.52 & 1308.30 & -9583.47 & 13048.38 \end{bmatrix},$$

and the matrix $\widehat{\mathbf{\Theta}}^*$, that is very close to the true matrix $\mathbf{\Theta}$, since

$$\widehat{\mathbf{\Theta}}^* = \begin{bmatrix} 0.0001 & 0.0019 \\ -0.0015 & -6.0005 \\ 0.9995 & -2.0033 \\ -0.0010 & -3.9993 \end{bmatrix}, \qquad \mathbf{\Theta} = \begin{bmatrix} 0 & 0 \\ 0 & -6 \\ 1 & -2 \\ 0 & -4 \end{bmatrix}.$$

Finally, we obtain the matrix of consequents of the rules $\widehat{\mathbf{Q}}$, that is very close to the true matrix $\mathbf{Q}$:

$$\widehat{\mathbf{Q}}^* = \mathbf{\Omega}^T \widehat{\mathbf{\Theta}}^* = \begin{bmatrix} -1.9997 & 0.7678 \\ -1.9984 & 5.8494 \\ 5.5459 & 34.5583 \\ 5.5280 & -37.0220 \end{bmatrix}, \quad \mathbf{Q} = \mathbf{\Omega}^T \mathbf{\Theta} = \begin{bmatrix} -2.0000 & 0.7562 \\ -2.0000 & 5.8427 \\ 5.5372 & 34.5801 \\ 5.5372 & -37.0087 \end{bmatrix},$$

as well. The relative approximation error

$$\delta_{\mathbf{\Theta}} = \sum_{j=1}^{n} \frac{\left\| \widehat{\boldsymbol{\theta}}_j^* - \boldsymbol{\theta}_j \right\|}{\left\| \boldsymbol{\theta}_j \right\|}. \tag{6.26}$$

is very small; $\delta_{\boldsymbol{\Theta}} = 0.0023952$ since large data set was used and only numerical errors were taken into account.

The relative approximation error tends to zero by increasing the cardinality of data sets and decreasing the lengths of intervals $[t_1, t_2]$, $[t_2, t_3]$, ..., and $[t_K, t_{K+1}]$. A small approximation error can be obtained for more complicated dynamical systems. As the next numerical example we will consider again the Chen's attractor discussed in Section 5.2.2.

Example 6.4. Let us consider the Chen's attractor from Example 5.7 described by three multilinear differential equations (5.43) subjected to disturbances:

$$\begin{cases} \dot{x} = a\,(y - x) + \xi_1, \\ \dot{y} = (c - a)\,x - xz + cy + \xi_2, \\ \dot{z} = xy - bz + \xi_3, \end{cases} \tag{6.27}$$

where $a = 35$, $b = 3$, $c = 28$ and

$$\xi_1\,(t) = 0.2\sin(2t\sin\,(4\sin\,(5t))), \tag{6.28}$$

$$\xi_2\,(t) = 0.1\cos(5t\cos\,(3\cos\,(2t))), \tag{6.29}$$

$$\xi_3\,(t) = 2\sin(6t\cos\,(2\sin\,(3t))). \tag{6.30}$$

The above system without disturbances has been already described by the zero-order P1-TS system (5.46) with the inputs $[x\,(t)\,, y\,(t)\,, z\,(t)]^T \in D^3$ and the outputs $[\dot{x}\,(t)\,, \dot{y}\,(t)\,, \dot{z}\,(t)]^T$. The data describing system trajectory come from the solution of (6.27) by nonzero signals $\xi_k\,(t)$ as in (6.28)-(6.30). The first disturbance is shown in Fig. 6.2 To integrate the equations, the `ode45` solver from Matlab was used with relative error tolerance 10^{-10} and the absolute error tolerance 10^{-13}. The integration step size varied from $5.7 \cdot 10^{-9}$ to 10^{-3} with the mean value $4.1 \cdot 10^{-4}$. The plots of the solution by the initial conditions $(x\,(0)\,, y\,(0)\,, z\,(0)) = (0, -10, 1)$ and the time interval $[0, 4]$ are shown in Figs. 6.3, 6.4 and 6.5. The number of samples used for

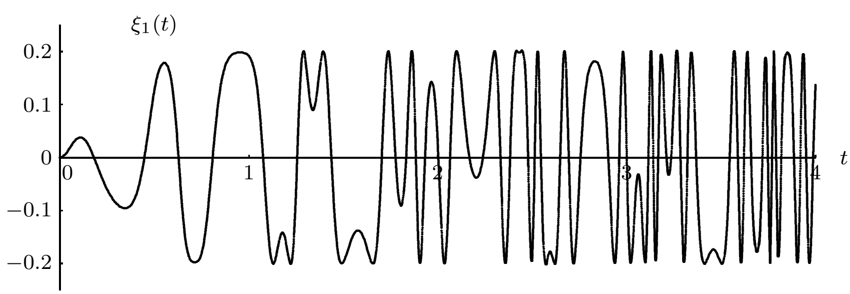


Fig. 6.2 Disturbance ξ_1 described by (6.28)

computations was $K = 9753$. The location of the cuboid D^3 following from the data set is determined by (see Figs. 6.3, 6.4 and 6.5):

$$-\alpha_1 = \min_{t\in\{t_1,\ldots,t_K\}} x(t) = -33.3629, \qquad \beta_1 = \max_{t\in\{t_1,\ldots,t_K\}} x(t) = 24.7308,$$

$$-\alpha_2 = \min_{t\in\{t_1,\ldots,t_K\}} y(t) = -40.8660, \qquad \beta_2 = \max_{t\in\{t_1,\ldots,t_K\}} y(t) = 36.8520,$$

$$-\alpha_3 = \min_{t\in\{t_1,\ldots,t_K\}} z(t) = 0.9987, \qquad \beta_3 = \max_{t\in\{t_1,\ldots,t_K\}} z(t) = 68.6232.$$

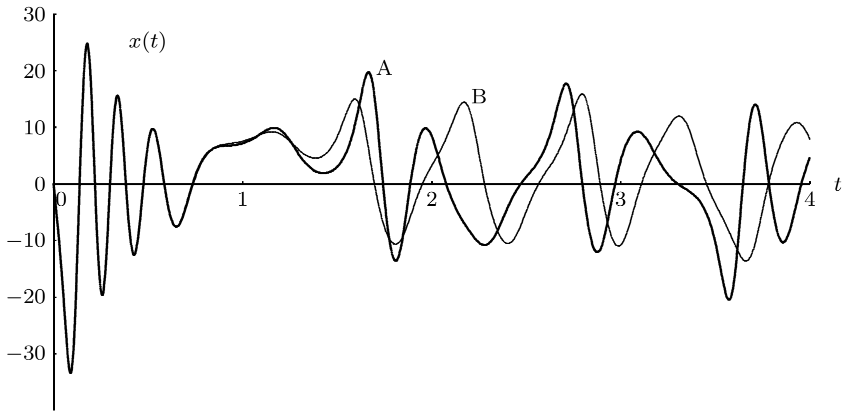


Fig. 6.3 Solution $x(t)$ of differential equations (6.27): A - without disturbances, B - including disturbances $\xi_k(t)$ given by (6.28)-(6.30)

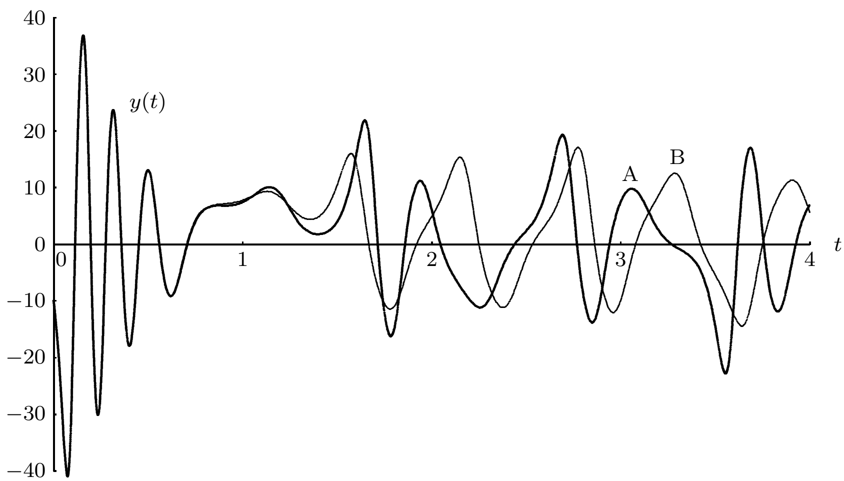


Fig. 6.4 Solution $y(t)$ of differential equations (6.27): A - without disturbances, B - including disturbances $\xi_k(t)$ given by (6.28)-(6.30)

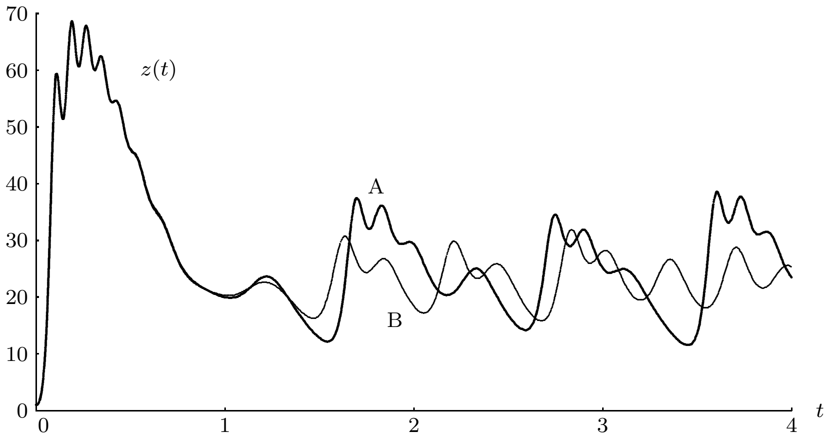


Fig. 6.5 Solution $z(t)$ of differential equations (6.27): A - without disturbances, B - including disturbances $\xi_k(t)$ given by (6.28)-(6.30)

The resulting matrices are as follows

$$\widehat{\mathbf{\Theta}}^* = \begin{bmatrix} 0.1069 & -0.1185 & -0.3446 \\ -34.9591 & -7.0005 & 0.0142 \\ 34.9984 & 28.0296 & -0.0167 \\ -0.0004 & 0.0025 & 1.0061 \\ -0.0025 & 0.0017 & -3.0017 \\ -0.0021 & -0.9996 & -0.0001 \\ -0.0006 & -0.0031 & 0.0006 \\ 0.0000 & 0.0000 & -0.0001 \end{bmatrix}, \quad \mathbf{\Theta} = \begin{bmatrix} 0 & 0 & 0 \\ -35 & -7 & 0 \\ 35 & 28 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -3 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$\widehat{\mathbf{Q}}^* = \begin{bmatrix} -264.25 & -875.18 & 1368.4 \\ -2294.3 & -1345.8 & -1019.0 \\ 2456.7 & 1296.5 & -1241.3 \\ 424.85 & 837.18 & 913.26 \\ -258.02 & 1388.8 & 1154.8 \\ -2296.4 & -3008.8 & -1217.0 \\ 2459.8 & 3544.2 & -1434.2 \\ 419.68 & -842.17 & 705.44 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} -262.61 & -877.39 & 1360.4 \\ -2295.9 & -1342.1 & -1013.6 \\ 2457.5 & 1298.7 & -1232.5 \\ 424.24 & 834.04 & 908.38 \\ -262.61 & 1378.8 & 1157.5 \\ -2295.9 & -3014.5 & -1216.5 \\ 2457.5 & 3554.9 & -1435.4 \\ 424.24 & -838.36 & 705.51 \end{bmatrix}$$

Thus, both the matrix $\widehat{\mathbf{\Theta}}^*$ is close to $\mathbf{\Theta}$, and $\widehat{\mathbf{Q}}^*$ is close to $\mathbf{Q}$. The global relative error (6.26) in the presence of disturbances is $\delta_{\mathbf{\Theta}} = 0.11576$ and it is about two times bigger than the one obtained without disturbances.

6.4 Estimation of P1-TS Model by Recursive Least Squares

For applications, if there is a need to process measurements as soon as they become available, one can use a *recursive* (on-line, implicit, iterative, sequential) *procedures*. The so called *recursive least squares* (RLS) procedure is very attractive in practice and well-known in the literature. Below we will adopt it to our problem.

We assume that the available data for every variable z_j contain K measured data pairs

$$\mathcal{W}_j = \{(\mathbf{w}_1, d_{j,1}), \ldots, (\mathbf{w}_K, d_{j,K})\} \subset \mathbb{R}^{2^n} \times \mathbb{R}, \qquad j = 1, \ldots, n, \tag{6.31}$$

where the vectors $\mathbf{w}_k = \mathbf{w}(t_k)$ are computed according to (6.16) and $d_{j,k} = d_j(t_k, t_{k+1})$ as defined in (6.7). Instead of the sum of squared errors defined in (6.15) we assume a slightly modified (more general) criterion

$$E_j(\lambda) = \sum_{k=1}^{K} \lambda^{K-k} \epsilon_j^2(t_k, t_{k+1}), \qquad j = 1, 2, \ldots, n, \tag{6.32}$$

where λ is called the forgetting factor, $(0 < \lambda \leq 1)$ and the error ϵ_j for the variable z_j is given by (6.14). As one can see, the forgetting factor operates as a weight which diminishes for the more observed data.

By the new criterion (6.32), the pseudocode of the RLS algorithm which minimizes $E_j(\lambda)$ is as follows.

First we should establish the forgetting factor λ; a good rule of thumb is $\lambda \in [0.92, 0.99]$, [87].

For all variables $j = 1, 2, \ldots, n$ perform the following steps:

Initialize the vector $\mathbf{q}_j$ and the matrix $\mathbf{P}$

$$\mathbf{q}_{j,0} = \mathbf{0}, \qquad \mathbf{P}_0 = p_0 \mathbf{I}_{2^n \times 2^n} \ ,$$

where p_0 is a very large number, say $p_0 \in \{10^8, 10^9, \ldots, 10^{15}\}$.

For all samples $k = 1, 2, \ldots, K$ perform the following steps:

For the data $\mathbf{w}_k$, $d_{j,k}$ and the given vector $\mathbf{q}_{j,k-1}$ calculate the error

$$\epsilon_{j,k} = \mathbf{w}_k^T \mathbf{q}_{j,k-1} - d_{j,k} \ . \tag{6.33}$$

Find the Kalman gain vector

$$\mathbf{h}_k = \frac{\mathbf{P}_{k-1} \mathbf{w}_k}{\lambda + \mathbf{w}_k^T \mathbf{P}_{k-1} \mathbf{w}_k} \ . \tag{6.34}$$

Calculate the updated vector

$$\mathbf{q}_{j,k} = \mathbf{q}_{j,k-1} - \epsilon_{j,k}\mathbf{h}_k \ , \tag{6.35}$$

`and the inverse correlation matrix`

$$\mathbf{P}_k = \frac{1}{\lambda}\left(\mathbf{P}_{k-1} - \mathbf{h}_k\mathbf{w}_k^T\mathbf{P}_{k-1}\right), \tag{6.36}$$

end `of calculating` $\mathbf{q}_j$ `for the variable` z_j.
end.

The proof of the algorithm is given in Appendix C.4. In the above procedure both the Kalman gain vector, and the inverse correlation matrices have a local meaning, since they are computed for separate variables z_j.

According to the RLS algorithm, starting from an initial matrix $\mathbf{P}_0$, the matrix $\mathbf{P}_k$ can be calculated in a recursive manner avoiding any matrix inversion. As the forgetting factor λ approaches 1, the memory of the procedure tends to be a perfect one equaling all past measurements with more recent ones. If there are no significant changes in the process parameters (the process is known to be stationary), working with $\lambda = 1$ will result in good estimates [87], [188]. In a nonstationary environment, with changing system dynamics, the influence of past observations will be reduced and λ will be smaller than 1. In this way, the present measurements have a stronger influence on the consequents estimates than the past one.

6.5 Summary

Based on analytical results concerning exact fuzzy modeling of multilinear dynamical systems which provide necessary and sufficient conditions for transformation of fuzzy rules into crisp model, the identification problem from observation data was stated and solved. The theorem on existence of the solution in the form of the P1-TS system was proved. For such system, both the batch procedure and a recursive one, were described in detail. The computations can be performed on-line using RLS method described in Section 6.4, where there is no need to have the whole data set before beginning the estimation process. Examples of identification for two- and three-dimensional nonlinear dynamical systems were given.

The advantages of the proposed approach can be summarized as follows.

- The methodology preserves the interpretability of the fuzzy models, which is a key property of the Pd-TS systems.
- The algorithm applies to P2-TS systems described in Chapter 4, since P2-TS systems are based on the same theory, involving generators and fundamental matrices.
- The continuous dynamical models have been converted into the discrete-time form. Thus, the method automatically applies to discrete-time multilinear systems, as well.

The proposed method can be viewed as a supervised learning algorithm for the adaptive linear neural network (see e.g. [58]), in which the consequents of the rules are interpreted as the weights of neurons. For the learning process of such network, we should take the training data set defined in (6.31). This means that many of the well-known learning procedures developed for the neural networks can be applied to solve the identification problem stated in this section, including both feedforward and recurrent (e.g. Hopfield) networks.

Chapter 7
Binary Classification Using P1-TS Rule Scheme

Most supervised learning algorithms are either regression or classification procedures, depending on whether the desired system output is real-valued or binary-valued. Such algorithms belong to important techniques in machine learning, computational intelligence and data mining [137], [201]. Classification systems (*classifiers* for short) are used for solving the problems which arise in many fields including pattern recognition, vision analysis and other decision making purposes.

Classifiers must often be created from data, because there is not enough expert knowledge to determine their parameters completely. They take as inputs a set of cases (n-dimensional vectors), each belonging to one of a small number of classes. System output must accurately predict the class to which a new case belongs. The smallest reasonable number of classes is two. In such systems, called *binary classifiers*, one class contains "negative" elements and the other - "positive" elements. Binary classification task is a basic one. It can be extended to the problem that involves more than two classes. In an m-class problem this can be done by repeatedly using one of the classes as a positive class, and the rest as the negative classes. In other words an m-class problem can be converted into m two-class problems in which one class is separated from the remaining classes. Such a method is known as the *one-against-all* method. It is worth adding that this simple scheme among many sophisticated methods used for multiclass classification problems is hard to beat [162], [163].

Classification problems have been widely studied in the literature, including theoretical and practical aspects of *machine learning* and *data mining*, such as *empirical risk minimization*, *regularization*, *generalization ability*, *robustness (stability)*, problem solving by large data sets, etc. [1], [34], [57], [76], [126], [137], [148], [154], [156], [157], [167], [189], [201]. Many approaches have been proposed to the automatic generation of the rules from numerical data for classification problems. They involve *heuristic procedures*, *artificial neural networks*, *evolutionary algorithms*, *nearest neighbor method*, *support vector machines*, *clustering methods*, *classification trees*, *Fisher discriminants*, and

J. Kluska: Analytical Methods in Fuzzy Modeling and Control, STUDFUZZ 241, pp. 199–216.
springerlink.com

other approaches which more or less involve *fuzzy logic*, such as *fuzzy support vector machines*, *neuro-fuzzy techniques*, *flexible neuro-fuzzy systems*, *fuzzy nearest neighbor method*, and so forth [2], [3], [24], [32], [44], [61], [62], [63], [65], [66], [67], [69], [88], [113], [116], [119], [127], [138], [141], [144], [145], [146], [159], [165], [166], [171], [172], [173], [204], [218]. The generation of the rules from numerical data for classification problems can also be done by soft-computing methods involving fuzzy logic, such as fuzzy support vector machines, neuro-fuzzy techniques, fuzzy nearest neighbor method and other methods. Usually, the classifiers obtained by the soft-computing techniques are represented by the fuzzy "If-then" prediction rules. Such classifiers are especially suitable, because they do not have some of the drawbacks of crisp rule based classifiers.

The rules discovered by the classifiers can be evaluated according to several criteria, such as the degree of confidence in the prediction, classification accuracy rate on unknown-class instances and interpretability. The last two criteria are of major importance in fuzzy classification systems. The rules should be highly interpretable, since the user of the classifier should be able to understand the rules, especially in such areas as medical or technical diagnostics [73], [145].

Our goal in this chapter is to show that the theory of Pd-TS systems developed in this book can be helpful to obtain very simple classifiers in the form of highly interpretable fuzzy rule-based systems. Namely, we propose a conception of obtaining a set of the rules of the P1-TS system as a binary classification problem solver. We do not attempt to modify the membership functions of the Pd-TS system, as this might degrade the interpretability of the fuzzy rules. Furthermore, we do not aspire to prove the novel classifier to be a good large-scale-problem-solver or the best classifier among a huge number of solutions proposed in the literature. The answer to the question of "how good is P1-TS (or P2-TS) system as a classifier, in comparison to other classifiers" requires a separate comprehensive study and it is not intended in this book. Thus, the result of this chapter should be taken with a grain of salt. We will use the results developed in previous sections, especially those from Chapter 6 related to modeling of the rule-based system from the input-output data, and the results from Section 5.11 referring to contradictory rules.

7.1 Problem Description

Assume that the available original learning data consist of the following input-output pairs (training patterns)

$$\{(\mathbf{z}_1, c_1), (\mathbf{z}_2, c_2), \ldots, (\mathbf{z}_Q, c_Q)\} = Z \times \{-1, 1\} \subset \mathbb{R}^n \times \{-1, 1\}, \qquad (7.1)$$

where

$$\mathbf{z}_i = [z_{i,1}, \ldots, z_{i,n}]^T, \qquad i = 1, \ldots, Q, \quad (Q \geq 2^n). \tag{7.2}$$

In reality we expect the cardinality of the data set to be much greater than 2^n, where n is the dimension of the original space containing the data points. Every vector $\mathbf{z}_i \in Z$ is identified with either the *first class*, denoted by "-1", or with the *second class* denoted by "1".

The smallest nonempty hypercuboid $D^n \supset Z$ is determined by

$$-\alpha_k = \min_{i=1,\ldots,Q} \{z_{i,k}\}, \qquad \beta_k = \max_{i=1,\ldots,Q} \{z_{i,k}\}, \qquad k = 1, \ldots, n. \tag{7.3}$$

Next we define the fuzzy sets N_k and P_k, $(k = 1, 2, \ldots, n)$ as in (2.11)-(2.12). The fuzzy set N_k we interpret as *near "$-\alpha_k$"* and P_k - as *near "β_k"*. For the given data set we want to find the system of fuzzy rules for a P1-TS system, which behaves like a *classifier*. To do this we start with considering the MISO P1-TS system with n inputs z_1, z_2, …, z_n and the output S, described by the following Q fuzzy rules

R_1 : If z_1 is N_1 and z_2 is N_2 and … and z_n is N_n,
then $\mathbf{z}$ belongs to the class c_1,

⋮

R_v : If z_1 is A_{i_1} and z_2 is A_{i_2} and … and z_n is A_{i_n},
then $\mathbf{z}$ belongs to the class c_v,

⋮

R_Q : If z_1 is P_1 and z_2 is P_2 and … and z_n is P_n,
then $\mathbf{z}$ belongs to the class c_Q,

where $A_{i_k} \in \{N_k, P_k\}$, $(k = 1, 2, \ldots, n$ and $i_k \in \{0, 1\}$ as defined in (2.15)) and c_v is the label denoting one of two classes; $c_v \in \{-1, 1\}$. The consequents of the rules are clear, since every vth rule refers to individual vth point from the data set (7.1). The antecedents of the rules have a simple interpretation as well, since we can easily evaluate the degree to which the particular antecedent matches the point $\mathbf{z}_v$ from the data set Z. However, the fuzzy rules R_1, …, R_Q do not seem to be suitable for modeling a classifier, especially when the data set (7.1) contains a large number of elements, in relation to the number of vertices of the hypercuboid D^n (see Corollary 2.7 in Section 2.4). Namely, for large data sets we are not sure that every fuzzy rule reflects correctly the individual membership of every point $\mathbf{z} \in Z$ to the appropriate class, since many points $\mathbf{z}$ can be far from vertices of the hypercuboid D^n, i.e. they have a large entropy [110].

In order to guarantee interpretability of the rules, our main goal is to find "the best" fuzzy rule-based system as a classifier, by preserving the antecedents of the rules which are characteristic for P1-TS systems.

7.2 The Fuzzy Rules with Proximity Degrees

The rules of the classifier should approximate the input-output data pairs. In general the rules do not concern an explicitly given system described by conventional mathematical equations. From the fuzzy rules point of view, every input vector $\mathbf{z}$ belongs to the class $c_v = -1$ or $c_v = 1$ in some degree. To define this formally, let us consider the distance ρ between a point $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{R}^n$ and some (temporarily fixed) point $\mathbf{z}_0 = (z_{0,1}, \ldots, z_{0,n}) \in \mathbb{R}^n$. The main attention we will pay to the Minkowski distance of order p (p-norm distance)

$$\rho(\mathbf{z}, \mathbf{z}_0) = \left(\sum_{i=1}^{n} |z_i - z_{0,i}|^p \right)^{1/p}, \qquad 1 \leq p \leq \infty. \tag{7.4}$$

The p-norm distances by $p = 1$ (1-norm or Manhattan distance) or $p = 2$ (2-norm or Euclidean distance), or infinity norm (Chebyshev distance; $\rho(\mathbf{z}, \mathbf{z}_0) = \max_{i=1,\ldots,n} |z_i - z_{0,i}|$ for $p = \infty$) are commonly used in many fields. The triangle inequality for the distance measure ρ does not hold for $p < 1$ [143].

We can use the other distance measures [40], e.g. Mahalanobis distance

$$\rho(\mathbf{z}, \mathbf{z}_0) = \sqrt{(\mathbf{z} - \mathbf{z}_0)^T \mathbf{K}^{-1} (\mathbf{z} - \mathbf{z}_0)}, \tag{7.5}$$

where $\mathbf{K}$ is the covariance matrix. If $\mathbf{K}$ is diagonal, then the resulting function is the normalized Euclidean distance, which is the same as (7.4) by rescaling the components. i.e. by substituting z_i and $z_{0,i}$ by $k_i z_i$ and $k_i z_{0,i}$, ($k_i > 0$), respectively. For $\mathbf{K} = \mathbf{I}$ the distance (7.5) reduces to (7.4) by $p = 2$.

In order to measure the proximity degree μ of the point $\mathbf{z}$ to $\mathbf{z}_0$ we will use a radial function, i.e. such continuous function that decreases monotonically with the distance ρ. We propose to consider radial functions of the form

$$\mu(\mathbf{z}, \mathbf{z}_0) = a^{-w}, \qquad w = \frac{(\rho(\mathbf{z}, \mathbf{z}_0))^r}{\sigma}, \qquad a > 1, \quad r > 0, \quad \sigma > 0. \tag{7.6}$$

By the power $r = 1$ and the constant $a = \exp(1)$, the above radial function is

$$\mu(\mathbf{z}, \mathbf{z}_0) = \exp(-\rho(\mathbf{z}, \mathbf{z}_0)/\sigma), \qquad \sigma > 0. \tag{7.7}$$

The other radial functions can be taken into account, as well, e.g.

$$\mu(\mathbf{z}, \mathbf{z}_0) = (1 + \rho(\mathbf{z}, \mathbf{z}_0))^{-b}, \qquad b > 0. \tag{7.8}$$

The proximity degree μ as a function of two variables $\mu : D^n \times D^n \to (0, 1]$ can be viewed as a membership function of the fuzzy set interpreted as "the point $\mathbf{z}$ is approximately the same as $\mathbf{z}_0$". Observe that $\mu(\mathbf{z}, \mathbf{z}_0) > 0$ for any $\mathbf{z}$ and $\mathbf{z}_0$.

The TS system should approximate the classification system. Thus, the conclusions of the rules can be only close, but not exactly equal to the desired values from the binary set $\{-1, 1\}$. According to Corollary 2.7 and the proof of Theorem 2.4 we can fuzzify the consequents of the rules $R_1, \ldots, R_Q$ by introducing the "similarity degrees" (certainty degrees or confidence factors) for every rule. Suppose $\mathbf{z}_0 = \mathbf{z}_k$ is a fixed vector from the set Z defined in (7.1). In every rule R_v we replace its consequent by "the output S is c_v with similarity degree $\mu(\mathbf{z}, \mathbf{z}_v)$", since $\mu(\mathbf{z}, \mathbf{z}_v) = 1$ if, and only if $\mathbf{z} = \mathbf{z}_v$. Thus, we formulate the following artificial system of $Q \geq 2^n$ fuzzy rules for the subsequent input vectors $\mathbf{z}_1, \ldots, \mathbf{z}_Q$ from the data set, which are followed by the confidence degrees

$$\begin{aligned}
R'_1 &: \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \\
&\quad \text{then } S \text{ is } c_1 \text{ with similarity degree } \mu(\mathbf{z}, \mathbf{z}_1), \\
&\quad\quad \vdots \\
R'_v &: \text{If } z_1 \text{ is } A_{i_1} \text{ and } z_2 \text{ is } A_{i_2} \text{ and } \ldots \text{ and } z_n \text{ is } A_{i_n}, \\
&\quad \text{then } S \text{ is } c_v \text{ with similarity degree } \mu(\mathbf{z}, \mathbf{z}_v), \\
&\quad\quad \vdots \\
R'_Q &: \text{If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } P_n, \\
&\quad \text{then } S \text{ is } c_Q \text{ with similarity degree } \mu(\mathbf{z}, \mathbf{z}_Q),
\end{aligned}$$

where $(c_1, \ldots, c_Q) \in \{-1, 1\}^Q$. In general, the system $R'_1, \ldots, R'_Q$ is not certain, since it includes contradictory rules in the sense of definition in Sections 2.5 and 5.11. The sources of uncertainty of the rules come from the modeling method and measurements. Furthermore, the system $R'_1, \ldots, R'_Q$ can be viewed as the well-known Mamdani type fuzzy expert system [123], since the consequents of the rules can be considered as fuzzy sets. However, in order to avoid the defuzzification procedure used for Mamdani type rule-bases, we consistently remain at the Takagi-Sugeno type systems with the similarity degrees.

7.3 Binary Classifier Equation

For the considered MISO P1-TS system we can write the following equations

$$\begin{aligned}
S(\mathbf{z}_1) &= \boldsymbol{\theta}^T \mathbf{g}(\mathbf{z}_1) = c_1 \cdot \mu(\mathbf{z}_1, \mathbf{z}_1) + \epsilon_1, \\
&\ \ \vdots \\
S(\mathbf{z}_Q) &= \boldsymbol{\theta}^T \mathbf{g}(\mathbf{z}_Q) = c_Q \cdot \mu(\mathbf{z}_Q, \mathbf{z}_Q) + \epsilon_Q,
\end{aligned}$$

where ϵ_k is an error, $\mathbf{g}(\mathbf{z}_k)$ is the value of the P1-TS system generator in the point $\mathbf{z}_k$, $\boldsymbol{\theta}$ is the vector of coefficients of the same system and $Q \geq 2^n$. In order to involve the similarity degrees of the fuzzy rules we extend the above equations. Since every consequent of the rule depends on $\mathbf{z}$, the vector $\boldsymbol{\theta}$ depends on $\mathbf{z}$, as well. Let $\mathbf{z}$ be any fixed vector $\mathbf{z}$ from D^n. Thus,

$$\boldsymbol{\theta}^T(\mathbf{z})\,\mathbf{g}(\mathbf{z}_1) = c_1 \cdot \mu(\mathbf{z}, \mathbf{z}_1) + \epsilon_1(\mathbf{z}),$$

$$\vdots$$

$$\boldsymbol{\theta}^T(\mathbf{z})\,\mathbf{g}(\mathbf{z}_Q) = c_Q \cdot \mu(\mathbf{z}, \mathbf{z}_Q) + \epsilon_Q(\mathbf{z}).$$

By the vector notation

$$\mathbf{q}(\mathbf{z}) = [c_1\mu(\mathbf{z}, \mathbf{z}_1), \ldots, c_Q\mu(\mathbf{z}, \mathbf{z}_Q)]^T, \quad \boldsymbol{\epsilon}(\mathbf{z}) = [\epsilon_1(\mathbf{z}), \ldots, \epsilon_Q(\mathbf{z})]^T, \tag{7.9}$$

we have the same equations in the matrix form

$$\begin{aligned} \boldsymbol{\epsilon}(\mathbf{z}) &= \mathbf{W}^T\boldsymbol{\theta}(\mathbf{z}) - \mathbf{q}(\mathbf{z}), \\ \mathbf{W}^T &= \left[\mathbf{g}(\mathbf{z}_1) \ldots \mathbf{g}(\mathbf{z}_Q)\right]_{2^n \times Q} . \end{aligned} \tag{7.10}$$

As one can see, the matrix $\mathbf{W}$ resembles the generalized fundamental matrix $\boldsymbol{\Omega}_e$ of P1-TS system, formerly defined by (5.185) in Section 5.11. Now for the given vector $\mathbf{z} \in D^n$ we can find such a vector $\widehat{\boldsymbol{\theta}}(\mathbf{z})$ that minimizes the sum of squared errors $\|\boldsymbol{\epsilon}(\mathbf{z})\|^2$

$$\widehat{\boldsymbol{\theta}}(\mathbf{z}) = \underset{\boldsymbol{\theta} \in \mathbb{R}^{2^n}}{\arg\min} \left\{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right\} = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{q}(\mathbf{z}) . \tag{7.11}$$

In general $\mathbf{W}^+ = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T$ is the pseudoinverse of the matrix $\mathbf{W}$. For the sake of simplicity we assume that the matrix $\mathbf{W}^T\mathbf{W}$ is nonsingular. In the simplest case, when $Q = 2^n$ and the data set is such that $\mathbf{W}$ is nonsingular, the solution is

$$\widehat{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{W}^{-1}\mathbf{q}(\mathbf{z}). \tag{7.12}$$

The output of the P1-TS system approximated by the data is given by

$$S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}(\mathbf{z}). \tag{7.13}$$

In this way we obtained the output of the rule-based system, which is a regressive model for the given data set (7.1). Let

$$\operatorname{sign}(x) = \begin{cases} -1 \Leftrightarrow x \leq 0 \\ \;\;\,1 \Leftrightarrow x > 0 \; . \end{cases} \tag{7.14}$$

For the vector $\widehat{\boldsymbol{\theta}}(\mathbf{z})$ given by (7.11), the generator $\mathbf{g}(\mathbf{z})$ of the P1-TS system and the vector of consequents of the rules $\mathbf{q}(\mathbf{z})$ defined by (7.9), we define the classifier as follows

$$\operatorname{class}(\mathbf{z}) = \operatorname{sign}\left(\mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}(\mathbf{z})\right), \qquad \forall\, \mathbf{z} \in D^n. \tag{7.15}$$

Note that the classifier is not defined for the points outside the hypercuboid D^n. Below we justify the outcome.

Suppose there exist the parameters $p = p_0$ of the norm (7.4), $a = a_0$, $r = r_0$ and $\sigma = \sigma_0$ of the radial function (7.6) (or $b = b_0$ of the radial function (7.8)), such that the equation

$$\mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}(\mathbf{z}) = 0, \tag{7.16}$$

has a solution, where $\widehat{\boldsymbol{\theta}}(\mathbf{z})$ is given by (7.11) and $\mathbf{g}(\mathbf{z})$ is the P1-TS system generator. Then the manifold (7.16) divides all points from the data set Z into two disjoint subsets

$$Z_1 = \{\mathbf{z} \in Z : \text{class}(\mathbf{z}) = -1\}, \tag{7.17}$$

and

$$Z_2 = \{\mathbf{z} \in Z : \text{class}(\mathbf{z}) = 1\}, \tag{7.18}$$

where $\text{class}(\mathbf{z})$ is given by (7.15), such that

$$Z_1 \cup Z_2 = Z. \tag{7.19}$$

According to Theorem 3.15, the output of the P1-TS system is a function $S : D^n \to [-1, 1]$, since $\min\{-1, 1\} = -1$ and $\max\{-1, 1\} = 1$. The output S is a continuous function of $\mathbf{z}$ at any point from the hypercuboid D^n and reaches both extrema -1 and 1. The point 0 belongs to $[-1, 1]$, so that in practice it is possible to find (rather numerically) such parameters $p = p_0$, $a = a_0$, $r = r_0$ and $\sigma = \sigma_0$ (or $b = b_0$), that the equation (7.16) holds, i.e. the manifold (hypersurface) (7.16) is a nonempty subset in D^n. This decision surface partitions D^n into two sets, one for each class. It classifies all the points on one side of the decision boundary as belonging to the class "-1" (when $\mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}(\mathbf{z}) \leq 0$) and all those on the other side as belonging to the class "1", (when $\mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}(\mathbf{z}) > 0$).

Observe that if we substitute the similarity degrees by 1 for every fuzzy rule, then it may happen that the problem has a solution for some data sets. In this case the P1-TS system output is $S(\mathbf{z}) = \mathbf{g}^T(\mathbf{z}) \cdot \widehat{\boldsymbol{\theta}}$, i.e. S is a multilinear function of $z_1, \ldots, z_n$. Thus, we can say that the problem is "multilinearly separable by a regression model". Of course, this does not mean that the data are multilinearly separable at all, since we did not use a general approach to find the multilinear function as a decision boundary, namely we did use the P1-TS model.

Example 7.1. Assume that the available original data set consists of Q input-output pairs on the plane

$$\left\{ \left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, c_1 \right), \ldots, \left(\begin{bmatrix} x_Q \\ y_Q \end{bmatrix}, c_Q \right) \right\} = Z \times \{-1, 1\} \subset \mathbb{R}^2 \times \{-1, 1\}, \tag{7.20}$$

From (7.10) and (7.11) we immediately obtain

$$\widehat{\boldsymbol{\theta}} = \begin{bmatrix} Q & \sum_{k=1}^{Q} x_k & \sum_{k=1}^{Q} y_k & \sum_{k=1}^{Q} x_k y_k \\ \sum_{k=1}^{Q} x_k & \sum_{k=1}^{Q} x_k^2 & \sum_{k=1}^{Q} x_k y_k & \sum_{k=1}^{Q} x_k^2 y_k \\ \sum_{k=1}^{Q} y_k & \sum_{k=1}^{Q} x_k y_k & \sum_{k=1}^{Q} y_k^2 & \sum_{k=1}^{Q} x_k y_k^2 \\ \sum_{k=1}^{Q} x_k y_k & \sum_{k=1}^{Q} x_k^2 y_k & \sum_{k=1}^{Q} x_k y_k^2 & \sum_{k=1}^{Q} x_k^2 y_k^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{k=1}^{Q} q_k \\ \sum_{k=1}^{Q} x_k q_k \\ \sum_{k=1}^{Q} y_k q_k \\ \sum_{k=1}^{Q} x_k y_k q_k \end{bmatrix}, \tag{7.21}$$

where $Q \geq 4$, $q_k = q_k(\mathbf{z})$ are components of the vector $\mathbf{q}(\mathbf{z})$ in (7.9) and of course $\widehat{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}(\mathbf{z})$.

Example 7.2. Let us consider numerical data and use the outcome of Example 7.1. The population consists of $Q = 20$ people; 10 women (labeled by "-1") and 10 men (labeled by "1"). For kth person we assign two attributes which constitute 20 vectors $\mathbf{z}_k = [x_k, y_k]^T \in D^2$, where x_k is the height [m] and y_k is the weight [kg], $(k = 1, \ldots, 20)$:

$$x_1 = 1.75,\ y_1 = 85,\ c_1 = -1, \qquad x_2 = 1.82,\ y_2 = 83,\ c_2 = 1,$$
$$x_3 = 1.61,\ y_3 = 64,\ c_3 = -1, \qquad x_4 = 1.87,\ y_4 = 89,\ c_4 = 1,$$
$$x_5 = 1.70,\ y_5 = 62,\ c_5 = -1, \qquad x_6 = 1.94,\ y_6 = 95,\ c_6 = 1,$$
$$x_7 = 1.60,\ y_7 = 55,\ c_7 = -1, \qquad x_8 = 1.72,\ y_8 = 73,\ c_8 = 1,$$
$$x_9 = 1.69,\ y_9 = 69,\ c_9 = -1, \qquad x_{10} = 1.88,\ y_{10} = 99,\ c_{10} = 1,$$
$$x_{11} = 1.71,\ y_{11} = 63,\ c_{11} = -1, \qquad x_{12} = 1.92,\ y_{12} = 90,\ c_{12} = 1,$$
$$x_{13} = 1.75,\ y_{13} = 60,\ c_{13} = -1, \qquad x_{14} = 2.00,\ y_{14} = 95,\ c_{14} = 1,$$
$$x_{15} = 1.74,\ y_{15} = 70,\ c_{15} = -1, \qquad x_{16} = 1.68,\ y_{16} = 75,\ c_{16} = 1,$$
$$x_{17} = 1.68,\ y_{17} = 55,\ c_{17} = -1, \qquad x_{18} = 1.71,\ y_{18} = 71,\ c_{18} = 1,$$
$$x_{19} = 1.62,\ y_{19} = 51,\ c_{19} = -1, \qquad x_{20} = 1.85,\ y_{20} = 75,\ c_{20} = 1.$$

One can check that the data are not linearly separable, i.e. there are no constants w_0, w_1, w_2 for which the straight line $w_0 + w_1 x + w_2 y = 0$ separates the data (see Fig. 7.1). According to (7.3) the rectangle D^2 is given by

$$D^2 = [1.60, 2.00] \times [51, 99]. \tag{7.22}$$

Assume temporarily that the consequents of the rules do not depend on the location of the points from D^2, i.e. $c_k \in \{-1, 1\}$ for $k = 1, \ldots, Q$. In this case the set of points satisfying equation (7.16) which "attempts" (unsuccessfully) to separate the two classes is nonempty

$$C = \left\{ (x, y) \in D^2 :\ \widehat{\boldsymbol{\theta}}^T \mathbf{g}(x, y) = 0 \right\}, \tag{7.23}$$

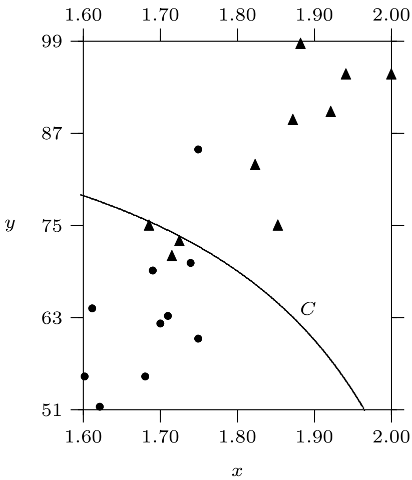


Fig. 7.1 The curve (7.23): $7.9919x + 0.17777y - 0.078455xy - 16.896 = 0$ for the data set from Example 7.2 obtained for consequents of the rules c_k from the binary set $\{-1, 1\}$

where $\widehat{\boldsymbol{\theta}}$ is computed according to (7.21). This set is a curve shown in Fig. 7.1. Observe that C does not separate the given data set into two classes. This is a typical situation, since in practice the data are not "easily separable" in the original n-dimensional space. Thus, the consequents of the rules must depend on the input vector $\mathbf{z}$, since otherwise the antecedents of the rules of the P1-TS system together with the binary consequents of the rules, may be incapable of solving the data separation problem in the original space containing the data set.

Let us choose the parameter $p = 1$ for the Minkowski distance measure (7.4) and $\sigma = 20$ for the radial function (7.7). Fig. 7.2 (a) shows two regions Z_1 and Z_2 defined by (7.17) and (7.18), respectively.

Fig. 7.2 (b) shows the classifier surface. In the same way we can take another radial function, say (7.8). In this case one can numerically check that by choosing the appropriate parameters of the distance measure ρ and similarity degree μ, the results are comparable.

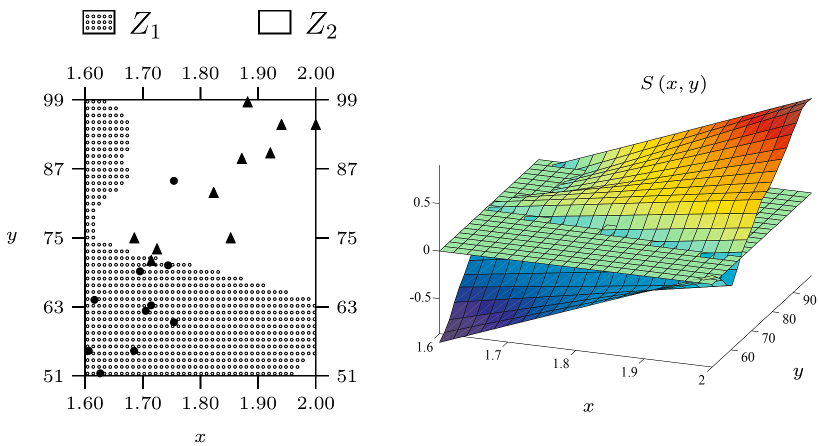
As an introduction to the next section we will consider an "ex-or" type problem of linearly nonseparable data, which cannot be solved by a single perceptron.

Example 7.3. The data set consists of four points

$$\left\{ \left(\begin{bmatrix} a_1 \\ b_1 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} a_2 \\ b_1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} a_1 \\ b_2 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} a_2 \\ b_2 \end{bmatrix}, -1 \right) \right\},$$

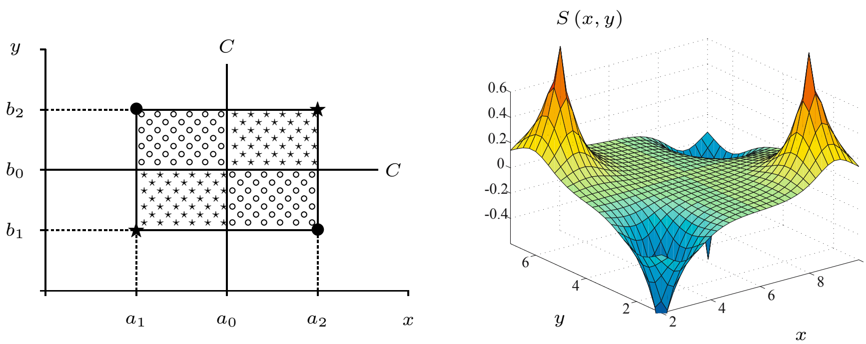
where $a_1 < a_2$ and $b_1 < b_2$ as shown in Fig. 7.3.

According to (7.3) the rectangle $D^2 = [a_1, a_2] \times [b_1, b_2]$, since $-\alpha_1 = a_1$, $\beta_1 = a_2$, $-\alpha_2 = b_1$ and $\beta_2 = b_2$. Let us take the Minkowski distance of order p and the radial function (7.7) by some parameter σ. In this simple case we can immediately write the fuzzy rules that model the binary classifier



(a) (b)

Fig. 7.2 (a) - The data set from Example 7.2 and its partition: '•' - women and '▲' - men. The subsets Z_1 and Z_2 obtained by the parameters $p = 1$ for the Minkowski distance measure (7.4) and $\sigma = 20$ for the radial function (7.7), (b) - decision surface of the classifier by the same parameters.



(a) (b)

Fig. 7.3 (a) - The data set from Example 7.3, (b) - decision surface for $a_1 = 3$, $a_2 = 9$, $b_1 = 2$, $b_2 = 6$, the Minkowski distance parameter $p = 2$ and $\sigma = 0.6$ of the function (7.7).

R_1 : If x is near a_1 and y is near b_1, then $S = c_1\mu_1(x, y)$,

R_2 : If x is near a_2 and y is near b_1, then $S = c_2\mu_2(x, y)$,

R_3 : If x is near a_1 and y is near b_2, then $S = c_3\mu_3(x, y)$,

R_4 : If x is near a_2 and y is near b_2, then $S = c_4\mu_4(x, y)$,

where

$$c_1 = -1, \quad c_2 = 1, \quad c_3 = 1, \quad c_4 = -1 \tag{7.24}$$

and

$$\mu_1(x,y) = \exp\left(-\left(|x-a_1|^p + |y-b_1|^p\right)^{1/p}/\sigma\right),$$

$$\mu_2(x,y) = \exp\left(-\left(|x-a_2|^p + |y-b_1|^p\right)^{1/p}/\sigma\right),$$

$$\mu_3(x,y) = \exp\left(-\left(|x-a_1|^p + |y-b_2|^p\right)^{1/p}/\sigma\right),$$

$$\mu_4(x,y) = \exp\left(-\left(|x-a_2|^p + |y-b_2|^p\right)^{1/p}/\sigma\right).$$

As $Q = n$, we have

$$\mathbf{W} = \boldsymbol{\Omega}^T = \begin{bmatrix} 1 & a_1 & b_1 & a_1 b_1 \\ 1 & a_2 & b_1 & a_2 b_1 \\ 1 & a_1 & b_2 & a_1 b_2 \\ 1 & a_2 & b_2 & a_2 b_2 \end{bmatrix},$$

and according to (7.12)

$$\widehat{\boldsymbol{\theta}} = \mathbf{W}^{-1} \begin{bmatrix} c_1 \mu_1(x,y) \\ c_2 \mu_2(x,y) \\ c_3 \mu_3(x,y) \\ c_4 \mu_4(x,y) \end{bmatrix}.$$

After algebraic calculations for c_k given in (7.24) we obtain

$$S(x,y) = \frac{1}{V} \begin{bmatrix} -\mu_1 a_2 b_2 - \mu_2 a_1 b_2 - \mu_3 a_2 b_1 - \mu_4 a_1 b_1 \\ \mu_1 b_2 + \mu_2 b_2 + \mu_3 b_1 + \mu_4 b_1 \\ \mu_1 a_2 + \mu_2 a_1 + \mu_3 a_2 + \mu_4 a_1 \\ -\mu_1 - \mu_2 - \mu_3 - \mu_4 \end{bmatrix}^T \begin{bmatrix} 1 \\ x \\ y \\ xy \end{bmatrix},$$

where $V = (b_1 - b_2)(a_1 - a_2)$. Below we prove that the set of separating points for which the equation (7.23) holds, is given by

$$C = \left\{(x,y) \in D^2 : \ x = a_0 \text{ or } y = b_0\right\}, \tag{7.25}$$

where

$$a_0 = (a_1 + a_2)/2, \qquad b_0 = (b_1 + b_2)/2.$$

One can check that

$$S(a_0, y) = \frac{y\mu_1 - y\mu_2 + y\mu_3 - y\mu_4 - \mu_1 b_2 + \mu_2 b_2 - \mu_3 b_1 + \mu_4 b_1}{2(b_2 - b_1)},$$

$$S(x, b_0) = \frac{x\mu_1 + x\mu_2 - x\mu_3 - x\mu_4 - \mu_1 a_2 - \mu_2 a_1 + \mu_3 a_2 + \mu_4 a_1}{2(a_2 - a_1)}.$$

Next we obtain

$$\mu_1(x,y) = \mu_2(x,y) \Leftrightarrow x = a_0, \quad \text{and} \quad \mu_3(x,y) = \mu_4(x,y) \Leftrightarrow x = a_0.$$

Thus,

$$S(a_0, y) = 0, \qquad \forall\, y.$$

Analogously for $y = b_0$ we obtain $S(x, b_0) = 0$ for every x. For the Minkowski distance and radial function (7.8) we obtain the same result.

The existence of the separating manifold (two perpendicular straight lines as shown in Fig. 7.3 (a)) has been proved for any p and σ, ($1 \leq p \leq \infty$ and $0 < \sigma < \infty$). Finally, the classifier equation $\text{class}(x, y) = \text{sign}\, S(x, y)$ can be written as

$$\text{class}(x, y) = \begin{cases} -1 \Leftrightarrow x \leq a_0 \ \&\ y \leq b_0 \text{ or } x > a_0 \ \&\ y > b_0 \\ \ \ 1 \Leftrightarrow x > a_0 \ \&\ y \leq b_0 \text{ or } x \leq a_0 \ \&\ y > b_0 \end{cases}.$$

We obtained highly interpretable fuzzy rules of the classifier for the smallest allowable cardinality of the input-output data set.

7.4 P1-TS System with Similarity Degrees as Optimal Binary Classifier

So far we have not considered the problem of how to obtain the best classifier parameters. Our goal in this section is to get the system of fuzzy rules for the binary classifier which preserves high interpretability of the fuzzy rules characteristic for P1-TS system and simultaneously has the best generalization ability in the class of the considered systems. In other words, we want to obtain an "optimal" binary classifier containing 2^n fuzzy rules, where n is the dimension of the input data vectors.

In order to improve generalization ability of the classifier, it would be desirable to *normalize* the data set (7.1) before beginning the whole procedure, especially when so called "outliers" are included in the data set. There are many normalization methods, which has been well described in the literature and therefore this simple technical problem will not be discussed in this book.

For the construction of the classifier we take into account some class (a crisp set) of parametrized functions that measure the similarity degree of vectors. Without loss of generality assume that we take the parameter p of the p-norm (7.4) and parameters a, r and σ of the radial function (7.6). By P_0 we denote the set of parameters for which the manifold (7.16) is a nonempty subset in D^n. By P^* we denote a nonempty subset of P_0 which contains optimal parameters of the classifier by some optimization criterion. Usually we require from the classifier to guarantee the *best generalization ability*. In such case one obtains numerically the set P^* by using the well-known *crossvalidation method*. Without going into details concerning crossvalidation, suppose from (7.11) we obtained the vector

$$\widehat{\boldsymbol{\theta}}_0(\mathbf{z}) = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{q}^*(\mathbf{z}), \tag{7.26}$$

where $\mathbf{q}^*(\mathbf{z})$ is the vector of consequents of the fuzzy rules obtained for the optimal parameters from the set P^*. This means that the classifier (7.15) is optimal in the sense of the best generalization ability.

Now we are able to construct a rule-based system containing only 2^n fuzzy rules instead of Q rules. According to Theorem 2.4, if $\boldsymbol{\theta}$ is the vector of crisp function coefficients (2.26) and $\boldsymbol{\Omega}$ is the fundamental matrix of the P1-TS system with 2^n fuzzy rules, then the vector of the rules consequents is $\mathbf{q} = \boldsymbol{\Omega}^T \boldsymbol{\theta}$. It is known that the vector $\boldsymbol{\theta}$ by the confidence factors $\mu(\mathbf{z}, \mathbf{z}_k) = 1$ for every fuzzy rule defines the P1-TS system as some multilinear function of $\mathbf{z}$. Thus, by $\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_0(\mathbf{z})$ obtained in (7.26) we get the following vector of the rules consequents

$$\widehat{\mathbf{q}}^*(\mathbf{z}) = \boldsymbol{\Omega}^T \left(\mathbf{W}^T \mathbf{W}\right)^{-1} \mathbf{W}^T \mathbf{q}^*(\mathbf{z}) \ , \tag{7.27}$$

which models the best classifier containing 2^n fuzzy rules. Finally, the system of fuzzy rules for the P1-TS system as an optimal classifier is as follows

$$R_1^* : \text{If } z_1 \text{ is } N_1 \text{ and } z_2 \text{ is } N_2 \text{ and } \ldots \text{ and } z_n \text{ is } N_n, \text{ then } S = \widehat{q}_1^*(\mathbf{z}),$$
$$\vdots$$
$$R_v^* : \text{ If } z_1 \text{ is } A_{i_1} \text{ and } z_2 \text{ is } A_{i_2} \text{ and } \ldots \text{ and } z_n \text{ is } A_{i_n}, \text{ then } S = \widehat{q}_v^*(\mathbf{z}),$$
$$\vdots$$
$$R_{2^n}^* : \text{ If } z_1 \text{ is } P_1 \text{ and } z_2 \text{ is } P_2 \text{ and } \ldots \text{ and } z_n \text{ is } P_n, \text{ then } S = \widehat{q}_{2^n}^*(\mathbf{z}),$$

where $A_{i_k} \in \{N_k, P_k\}$, $(k = 1, 2, \ldots, n,\ i_k \in \{0, 1\})$, as defined in (2.15). For the above rules, the values of "$-\alpha_k$" and "β_k" are given by (7.3). The consequents of the rules are not constant. They are weighted sums of the radial functions

$$q_j^*(\mathbf{z}) = \sum_{v=1}^{Q} \xi_{j,v} \mu^*(\mathbf{z}, \mathbf{z}_v), \qquad j = 1, \ldots, 2^n, \quad v = 1, \ldots, Q, \tag{7.28}$$

where $\mu^*(\mathbf{z}, \mathbf{z}_v)$ is the radial function defined for the optimal parameters from the set P^* and $\xi_{j,1}$, $\xi_{j,2}$, $\ldots$, $\xi_{j,Q}$ are real coefficients for jth consequent of the rule. The new rule-based system in which the consequents of the rules are weighted sums of radial functions we will call *P1-TS system with similarity degrees*. It performs the function of the optimal binary classifier.

Example 7.4. Let us consider the data from Example 7.2 again. Suppose the Minkowski distance measure (7.4) and the radial function (7.6) were chosen and the optimal parameters of these functions were (numerically) obtained as

$$p = p^*, \quad a = a^*, \quad r = r^*, \quad \sigma = \sigma^*. \tag{7.29}$$

At this moment the details concerning the method of obtaining the parameters (7.29) are not interesting for us. We assume only that for the above parameters the crossvalidation error as an optimization index is the smallest one. For the data x_k, y_k and c_k from Example 7.2, the consequents of the rules (7.26) are explicitly given by

$$q_k^*(x,y) = c_k \mu^*(x,y,x_k,y_k), \tag{7.30}$$

where

$$\mu^*(x,y,x_k,y_k) = (a^*)^{-w^*}, \qquad w^* = \frac{(\rho^*(x,y,x_k,y_k))^{r^*}}{\sigma^*}, \tag{7.31}$$

and

$$\rho^*(x,y,x_k,y_k) = \left(|x-x_k|^{p^*} + |y-y_k|^{p^*}\right)^{1/p^*}, \tag{7.32}$$

for $k = 1, \ldots, 20$. According to (7.27) and (7.22) the optimal fuzzy rules are as follows

R_1^* : If x is about 1.60 and y is about 51, then $S = \widehat{q}_1^*(x,y)$,
R_2^* : If x is about 2.00 and y is about 51, then $S = \widehat{q}_2^*(x,y)$,
R_3^* : If x is about 1.60 and y is about 99, then $S = \widehat{q}_3^*(x,y)$,
R_4^* : If x is about 2.00 and y is about 99, then $S = \widehat{q}_4^*(x,y)$.

The optimal consequents of the fuzzy rules are given by

$$\begin{bmatrix} \widehat{q}_1^*(x,y) \\ \widehat{q}_2^*(x,y) \\ \widehat{q}_3^*(x,y) \\ \widehat{q}_4^*(x,y) \end{bmatrix} = \mathbf{\Omega}^T \left(\mathbf{W}^T\mathbf{W}\right)^{-1} \begin{bmatrix} \sum_{k=1}^{Q} q_k^*(x,y) \\ \sum_{k=1}^{Q} x_k q_k^*(x,y) \\ \sum_{k=1}^{Q} y_k q_k^*(x,y) \\ \sum_{k=1}^{Q} x_k y_k q_k^*(x,y) \end{bmatrix}, \tag{7.33}$$

where

$$\mathbf{\Omega}^T \left(\mathbf{W}^T\mathbf{W}\right)^{-1} = \begin{bmatrix} 12.843 & -7.2001 & -0.15511 & 0.08598 \\ -45.515 & 28.531 & 0.40853 & -0.26553 \\ -6.7087 & 2.1571 & 0.20709 & -0.09460 \\ 10.865 & -6.2357 & -0.15643 & 0.08823 \end{bmatrix},$$

and the components of the optimal consequents vector are $q_k^* = q_k^*(x,y)$ as defined in (7.30)-(7.32). Since there are $2^n Q = 80$ coefficients $\xi_{j,v}$ in equation (7.28), we will write $\xi_{j,v}$ only for $j \in \{1, 2, 20\}$, for every consequent of the rule $v = 1, 2, 3, 4$:

$$\xi_{1,1} = -0.152, \quad \xi_{1,2} = -0.14717, \quad \ldots \quad \xi_{1,20} = -0.18071,$$

$$\xi_{2,1} = -0.35829, \quad \xi_{2,2} = 0.20845, \quad \ldots \quad \xi_{2,20} = 1.0648,$$

$$\xi_{3,1} = 0.59713, \quad \xi_{3,2} = 0.11542, \quad \ldots \quad \xi_{3,20} = -0.31207,$$

$$\xi_{4,1} = -0.21981, \quad \xi_{4,2} = -0.13964, \quad ... \quad \xi_{4,20} = -0.16138.$$

Thus, the optimal rules are as follows

R_1^* : If x is about 1.60 and y is about 51,
then $S = -0.152a^{-w_1} - 0.14717a^{-w_2} + \ldots - 0.18071a^{-w_{20}}$

R_2^* : If x is about 2.00 and y is about 51,
then $S = -0.35829a^{-w_1} + 0.20845a^{-w_2} + \ldots + 1.0648a^{-w_{20}}$

R_3^* : If x is about 1.60 and y is about 99,
then $S = 0.59713a^{-w_1} + 0.11542a^{-w_2} + \ldots - 0.31207a^{-w_{20}}$

R_4^* : If x is about 2.00 and y is about 99,
then $S = -0.21981a^{-w_1} - 0.13964a^{-w_2} + \ldots - 0.16138a^{-w_{20}}$

where

$$w_k = \left(|x - x_k|^p + |y - y_k|^p\right)^{r/p} / \sigma, \qquad k = 1, \ldots, 20.$$

This completes the description of the P1-TS system with similarity degrees, which performs the function of the best binary classifier, provided that all the parameters $a = a^*$, $p = p^*$, $r = r^*$ and $\sigma = \sigma^*$ are chosen so that they guarantee the smallest crossvalidation error.

7.5 The Regularization Algorithm and Support Vector Machines

A disadvantage of the proposed method is a large number of terms involved in the consequents of the rules. Therefore, before beginning the whole procedure, we can reduce the cardinality of the data set by using the method based on the support vector classification [189], mentioned in the introduction to this chapter. We very briefly characterize this method, since it is well described in the literature.

Most developments concerning classification start from a geometric viewpoint emphasizing *separating hyperplanes* and *margin*. However, there exist other interesting developments that use the idea of *regularization* [44], [157], [162], [167]. From this point of view the solution of the classification problem is as follows.

First, for the given data set $\{(\mathbf{z}_k, c_k) : k = 1, \ldots, Q\} = Z \times \{-1, 1\} \subset \mathbb{R}^n \times \{-1, 1\}$ as in (7.1), we choose a symmetric, positive definite function $K(\mathbf{x}, \mathbf{y})$, continuous on $Z \times Z$, called a *kernel function*. A kernel $K(\mathbf{x}, \mathbf{y})$ is positive definite if $\sum_{i,j=1}^{n} h_i h_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ for any natural n and a choice of $\mathbf{x}_1, \ldots, \mathbf{x}_n \in Z$ and $h_1, \ldots, h_n \in \mathbb{R}^n$. An example of such kernel is the Gaussian function

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma\right), \qquad \sigma > 0, \quad (\mathbf{x}, \mathbf{y}) \in Z \times Z, \qquad (7.34)$$

where $\|\mathbf{x}\|^2 = x_1^2 + \ldots + x_n^2$.

Next we set a weighted sum of the kernel functions

$$f(\mathbf{z}) = \sum_{k=1}^{Q} a_k K(\mathbf{z}, \mathbf{z}_k), \tag{7.35}$$

where the coefficients a_k must be found by solving a quadratic programming problem which will be stated later. The classifier function is

$$\text{class}(z) = \text{sign}(f(\mathbf{z})). \tag{7.36}$$

The above algorithm was compactly derived in [157] from *regularization theory* point of view, as the problem of finding

$$\min_{f \in H} \left\{ \frac{1}{Q} \sum_{k=1}^{Q} V(c_k, f(\mathbf{z}_k)) + \lambda \|f\|_K^2 \right\}, \tag{7.37}$$

where $V(c_k, f(\mathbf{z}_k))$ is a *loss function*

$$V(c_k, f(\mathbf{z}_k)) = \max(0, 1 - c_k f(\mathbf{z}_k)) \tag{7.38}$$

indicating the penalty we pay for guessing $f(\mathbf{z}_k)$ when the true value is c_k. The norm $\|f\|_K$ in (7.37) is the norm in a Reproducing Kernel Hilbert Space H, defined by a kernel function K, and λ is a *regularization parameter* quantifying our willingness to trade off accuracy of classification for a function with a small norm in the space H. The function V in (7.38) is often referred to as the *hinge loss* function. If $c_k f(\mathbf{z}_k)$ is at least 1, we pay no penalty for point k. If $c_k f(\mathbf{z}_k) < 1$, we pay a penalty linear in the amount by which we fail to satisfy the constraint. The quantity $c_k f(\mathbf{z}_k)$ is also known as the *margin*. The classical SVM algorithm as developed by Vapnik at al. [33], [189] uses the hinge loss. Without going into details, one can convert the problem (7.37)-(7.38) into the so called "primal" and "dual" convex quadratic programming (QP) problems, which both have optimal solutions. The SVM dual problem is substantially easier to solve than the primal (see e.g. [162]), namely

$$\max_{\boldsymbol{\tau} \in \mathbb{R}^Q} \left\{ -\frac{1}{2} \boldsymbol{\tau}^T \mathbf{H} \boldsymbol{\tau} + (\tau_1 + \ldots + \tau_Q) \right\}, \qquad \boldsymbol{\tau}^T = [\tau_1, \ldots, \tau_Q], \tag{7.39}$$

subject to the constraints

$$c_1 \tau_1 + \ldots + c_Q \tau_Q = 0, \tag{7.40}$$

$$0 \leq \tau_k \leq C, \qquad k = 1, \ldots, Q, \tag{7.41}$$

where C is some positive constant, and the matrix $\mathbf{H}$ is defined by its elements located in ith row and jth column

$$\mathbf{H} = \{c_i c_j K(\mathbf{z}_i, \mathbf{z}_j)\} \in \mathbb{R}^{Q\times Q}, \quad c_1, \ldots, c_Q \in \{-1, 1\}, \quad \mathbf{z}_1, \ldots, \mathbf{z}_Q \in Z. \tag{7.42}$$

Finally, the coefficients of the classification function (7.35) are $a_k = c_k \tau_k$, $(k = 1, \ldots, Q)$, where every τ_k is an optimal solution of the quadratic problem (7.39)-(7.41). The input data vectors $\mathbf{z}_k$ for which τ_k is different from zero are called *support vectors* (SVs). If we denote

$$SV = \{k : \tau_k > 0,\ \mathbf{z}_k \in Z\}, \tag{7.43}$$

then the solution (7.35) is given by

$$f(\mathbf{z}) = \sum_{k\in SV} c_k \tau_k K(\mathbf{z}, \mathbf{z}_k). \tag{7.44}$$

As one can see, the support vectors are *critical* for the solution of the classification problem. In practice, very often the number of support vectors is much smaller than Q. Thus, there is a big chance to reduce the cardinality of the training data set.

Two things are notable:

1. The Gaussian kernel function (7.34) is the same as the similarity degree $\mu(\mathbf{x}, \mathbf{y})$ in (7.7), with ρ being Euclidean distance by the parameter $r = 2$. Although our approach in Section 7.2 resembles rather regression than classification, it is worth noticing that the regularization method described briefly above for the classification, leads to the same solution as for the regression. For the solution of the regression problem (where c_k are real numbers), only the loss function V in (7.38) is changed for $V(c_k, f(\mathbf{z}_k)) = (c_k - f(\mathbf{z}_k))^2$, (see [157] for the details).
2. The final solution of the classification problem for the Gaussian function (7.34) must be obtained for the smallest generalization error. This error depends on two parameters: σ in the Gaussian function (7.34) and C in the constraints (7.41). Thus, the method of reducing the data set by the solution of quadratic programming problem (7.39) with the constraints (7.40)-(7.41) must be repeated many times to get the best solution for the parameters $\sigma = \sigma^*$ and $C = C^*$.

7.6 Summary

In this chapter we proposed the method of construction of a highly interpretable rule-based system as an optimal binary classifier. The advantages of the presented approach can be summarized as follows:

- The P1-TS fuzzy expert system as an optimal classifier contains highly interpretable fuzzy rules, with the simplest polynomial fuzzy sets for the inputs.

- Even though we used P1-TS systems, the whole procedure of constructing the classifier can be easily extended to P2-TS systems, since they are based on theory of generators and fundamental matrices.

A disadvantage of the method is a large number of terms involved in the consequents of the rules. Therefore for a large cardinality of the data set, we can reduce the number of data by finding the support vectors. However, if the number of support vectors is still large, we can try to substitute the consequents of the rules by some simpler functions of the input vector components. This can be done by using some approximation method which guarantees a sufficiently good generalization ability of the classifier. The result will depend on the data set. If the problem is multilinearly separable by the regressive model, then we obtain a simple solution, since the consequents vector is constant.

Appendix A
Kronecker Product of Matrices

This appendix is a brief description of the Kronecker product of matrices and its properties. For a detailed treatment the reader is referred to [43], [54], [83].

Definition A.1. Let $\mathbf{A} = \{a_{ij}\}$ and $\mathbf{B} = \{b_{kl}\}$ be matrices, $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times m}$. Then the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is the block matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{nm\times nm}. \tag{A.1}$$

The Kronecker product is also known as the direct product or the tensor product.

1. The Kronecker product "$\otimes$" is a bilinear operator. If k is a scalar, and $\mathbf{A}$, $\mathbf{B}$ and C are square matrices, such that $\mathbf{B}$ and C are of the same order, then

$$\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C}, \tag{A.2a}$$

$$(\mathbf{B} + \mathbf{C}) \otimes \mathbf{A} = \mathbf{B} \otimes \mathbf{A} + \mathbf{C} \otimes \mathbf{A}, \tag{A.2b}$$

$$k\,(\mathbf{A} \otimes \mathbf{B}) = (k\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (k\mathbf{B})\,. \tag{A.2c}$$

2. If $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are square matrices such that the products $\mathbf{AC}$ and $\mathbf{BD}$ exist, then $(\mathbf{A} \otimes \mathbf{B})\,(\mathbf{C} \otimes \mathbf{D})$ exists and

$$(\mathbf{A} \otimes \mathbf{B})\,(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}. \tag{A.3}$$

3. If $\mathbf{A}$ and $\mathbf{B}$ are invertible matrices, then

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}. \tag{A.4}$$

4. If $\mathbf{A}$ and $\mathbf{B}$ are square matrices, then for the transpose $(\mathbf{A}^T)$ we have

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T. \tag{A.5}$$

5. Let $\mathbf{A}$ and $\mathbf{B}$ be square matrices of orders n and m, respectively. If $\{\lambda_i \mid i = 1, \ldots, n\}$ are eigenvalues of $\mathbf{A}$ and $\{\mu_j \mid j = 1, \ldots, m\}$ are eigenvalues of $\mathbf{B}$, then $\{\lambda_i \mu_j \mid i = 1, \ldots, n,\ j = 1, \ldots, m\}$ are eigenvalues of $\mathbf{A} \otimes \mathbf{B}$. Also,

$$\det(\mathbf{A} \otimes \mathbf{B}) = (\det \mathbf{A})^m (\det \mathbf{B})^n, \tag{A.6a}$$

$$\operatorname{rank}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{rank} \mathbf{A} \ \operatorname{rank} \mathbf{B}, \tag{A.6b}$$

$$\operatorname{trace}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{trace} \mathbf{A} \ \operatorname{trace} \mathbf{B}. \tag{A.6c}$$

Appendix B
Generators and Fundamental Matrices for P1-TS Systems

This appendix contains the relationship between the ordered set containing vertices of the hypercuboid $D^n = [-\alpha_1, \beta_1] \times \ldots \times [-\alpha_n, \beta_n]$, the generators and fundamental matrices for the P1-TS systems with $n = 1, 2, 3, 4$ inputs $z_1, \ldots, z_4$. They are helpful for the fuzzy rules transformation into the crisp function and vice-versa.

B.1 Formulas for $n = 1$

B.1.1 Vertices of the Interval $D^1 = [-\alpha_1, \beta_1]$

$\gamma_1 = -\alpha_1$, $\gamma_2 = \beta_1$.

B.1.2 Generator

$$\mathbf{g}_1(z_1) = \begin{bmatrix} 1 \\ z_1 \end{bmatrix}. \tag{B.1}$$

B.1.3 Fundamental Matrix and Its Inverse

- General case

$$\boldsymbol{\Omega}_1 = \begin{bmatrix} 1 & 1 \\ -\alpha_1 & \beta_1 \end{bmatrix}, \qquad \boldsymbol{\Omega}_1^{-1} = \frac{1}{V_1} \begin{bmatrix} \beta_1 & -1 \\ \alpha_1 & 1 \end{bmatrix}, \tag{B.2}$$

  for $V_1 = \alpha_1 + \beta_1 > 0$.
- Unity interval $D^1 = [0, 1]$

$$\boldsymbol{\Omega}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{\Omega}_1^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}. \tag{B.3}$$

- Interval symmetrical around zero $D^1 = [-\alpha_1, \alpha_1]$

$$\boldsymbol{\Omega}_1 = \begin{bmatrix} 1 & 1 \\ -\alpha_1 & \alpha_1 \end{bmatrix}, \qquad \boldsymbol{\Omega}_1^{-1} = \frac{1}{2\alpha_1}\begin{bmatrix} \alpha_1 & -1 \\ \alpha_1 & 1 \end{bmatrix}. \tag{B.4}$$

B.2 Formulas for $n = 2$

B.2.1 Vertices of the Rectangle $D^2 = [-\alpha_1, \beta_1] \times [-\alpha_2, \beta_2]$,

$\boldsymbol{\gamma}_1 = (-\alpha_1, -\alpha_2)$, $\boldsymbol{\gamma}_2 = (\beta_1, -\alpha_2)$, $\boldsymbol{\gamma}_3 = (-\alpha_1, \beta_2)$, $\boldsymbol{\gamma}_4 = (\beta_1, \beta_2)$.

B.2.2 Generator

$$\mathbf{g}_2(z_1, z_2) = [1, z_1, z_2, z_1 z_2]^T. \tag{B.5}$$

B.2.3 Fundamental Matrix and Its Inverse

- General case

$$\boldsymbol{\Omega}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \end{bmatrix}, \quad \boldsymbol{\Omega}_2^{-1} = \frac{1}{V_2}\begin{bmatrix} \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \alpha_1\beta_2 & \beta_2 & -\alpha_1 & -1 \\ \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}, \tag{B.6}$$

where $V_2 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2) > 0$.
- Unity square $D^2 = [0, 1]^2$

$$\boldsymbol{\Omega}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \boldsymbol{\Omega}_2^{-1} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.7}$$

- Rectangle symmetrical around zero $D^2 = [-\alpha_1, \alpha_1] \times [-\alpha_2, \alpha_2]$

$$\boldsymbol{\Omega}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \alpha_1 & -\alpha_1 & \alpha_1 \\ -\alpha_2 & -\alpha_2 & \alpha_2 & \alpha_2 \\ \alpha_1\alpha_2 & -\alpha_1\alpha_2 & -\alpha_1\alpha_2 & \alpha_1\alpha_2 \end{bmatrix}, \ \boldsymbol{\Omega}_2^{-1} = \frac{1}{4\alpha_1\alpha_2}\begin{bmatrix} \alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & 1 \\ \alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & -1 \\ \alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & -1 \\ \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}. \tag{B.8}$$

B.3 Formulas for $n = 3$

B.3.1 Vertices of the Cuboid $D^3 = [-\alpha_1, \beta_1] \times [-\alpha_2, \beta_2] \times [-\alpha_3, \beta_3]$

$\gamma_1 = (-\alpha_1, -\alpha_2, -\alpha_3)$, $\gamma_2 = (\beta_1, -\alpha_2, -\alpha_3)$, $\gamma_3 = (-\alpha_1, \beta_2, -\alpha_3)$,
$\gamma_4 = (\beta_1, \beta_2, -\alpha_3)$, $\gamma_5 = (-\alpha_1, -\alpha_2, \beta_3)$, $\gamma_6 = (\beta_1, -\alpha_2, \beta_3)$,
$\gamma_7 = (-\alpha_1, \beta_2, \beta_3)$, $\gamma_8 = (\beta_1, \beta_2, \beta_3)$.

B.3.2 Generator

$$\mathbf{g}_3(z_1, z_2, z_3) = [1, z_1, z_2, z_1 z_2, z_3, z_1 z_3, z_2 z_3, z_1 z_2 z_3]^T . \qquad \text{(B.9)}$$

B.3.3 Fundamental Matrix and Its Inverse

- General case

$$\boldsymbol{\Omega}_3 = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & -\alpha_3 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & -\alpha_3 & -\beta_1\alpha_3 & \alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & -\alpha_3 & \alpha_1\alpha_3 & -\alpha_3\beta_2 & \alpha_1\alpha_3\beta_2 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & -\alpha_3 & -\beta_1\alpha_3 & -\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & \beta_3 & -\alpha_1\beta_3 & -\alpha_2\beta_3 & \alpha_1\alpha_2\beta_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 & \beta_3 & \beta_1\beta_3 & -\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 & \beta_3 & -\alpha_1\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_2\beta_3 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 & \beta_3 & \beta_1\beta_3 & \beta_2\beta_3 & \beta_1\beta_2\beta_3 \end{bmatrix}^T , \qquad \text{(B.10)}$$

$$\boldsymbol{\Omega}_3^{-1} = \frac{1}{V_3} \begin{bmatrix} \beta_1\beta_2\beta_3 & -\beta_2\beta_3 & -\beta_1\beta_3 & \beta_3 & -\beta_1\beta_2 & \beta_2 & \beta_1 & -1 \\ \alpha_1\beta_2\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_3 & -\beta_3 & -\alpha_1\beta_2 & -\beta_2 & \alpha_1 & 1 \\ \alpha_2\beta_1\beta_3 & -\alpha_2\beta_3 & \beta_1\beta_3 & -\beta_3 & -\alpha_2\beta_1 & \alpha_2 & -\beta_1 & 1 \\ \alpha_1\alpha_2\beta_3 & \alpha_2\beta_3 & \alpha_1\beta_3 & \beta_3 & -\alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & -1 \\ \beta_1\alpha_3\beta_2 & -\alpha_3\beta_2 & -\beta_1\alpha_3 & \alpha_3 & \beta_1\beta_2 & -\beta_2 & -\beta_1 & 1 \\ \alpha_1\alpha_3\beta_2 & \alpha_3\beta_2 & -\alpha_1\alpha_3 & -\alpha_3 & \alpha_1\beta_2 & \beta_2 & -\alpha_1 & -1 \\ \alpha_2\beta_1\alpha_3 & -\alpha_2\alpha_3 & \beta_1\alpha_3 & -\alpha_3 & \alpha_2\beta_1 & -\alpha_2 & \beta_1 & -1 \\ \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix} , \qquad \text{(B.11)}$$

where $V_3 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2)(\alpha_3 + \beta_3) > 0$.

- Unity cube $D^3 = [0,1]^3$

$$\boldsymbol{\Omega}_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Omega}_3^{-1} = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.12}$$

- Cuboid symmetrical around zero $D^3 = [-\alpha_1, \alpha_1] \times [-\alpha_2, \alpha_2] \times [-\alpha_3, \alpha_3]$

$$\boldsymbol{\Omega}_3 = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & -\alpha_3 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \alpha_1 & -\alpha_2 & -\alpha_1\alpha_2 & -\alpha_3 & -\alpha_1\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \\ 1 & -\alpha_1 & \alpha_2 & -\alpha_1\alpha_2 & -\alpha_3 & \alpha_1\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \\ 1 & \alpha_1 & \alpha_2 & \alpha_1\alpha_2 & -\alpha_3 & -\alpha_1\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 & \alpha_3 & -\alpha_1\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \\ 1 & \alpha_1 & -\alpha_2 & -\alpha_1\alpha_2 & \alpha_3 & \alpha_1\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & -\alpha_1 & \alpha_2 & -\alpha_1\alpha_2 & \alpha_3 & -\alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & \alpha_1 & \alpha_2 & \alpha_1\alpha_2 & \alpha_3 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \end{bmatrix}^T, \tag{B.13}$$

$$\boldsymbol{\Omega}_3^{-1} = \frac{1}{v_3} \begin{bmatrix} \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & \alpha_1 & -1 \\ \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & 1 \\ \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & 1 \\ \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & -1 \\ \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & 1 \\ \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & -1 \\ \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & -1 \\ \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}, \tag{B.14}$$

where $v_3 = 8\alpha_1\alpha_2\alpha_3$.

B.4 Formulas for $n = 4$

B.4.1 Vertices of the Hypercuboid $D^4 = [-\alpha_1, \beta_1] \times \ldots \times [-\alpha_4, \beta_4]$

$\boldsymbol{\gamma}_1 = (-\alpha_1, -\alpha_2, -\alpha_3, -\alpha_4)$, $\boldsymbol{\gamma}_2 = (\beta_1, -\alpha_2, -\alpha_3, -\alpha_4)$,

$\boldsymbol{\gamma}_3 = (-\alpha_1, \beta_2, -\alpha_3, -\alpha_4)$, $\boldsymbol{\gamma}_4 = (\beta_1, \beta_2, -\alpha_3, -\alpha_4)$,
$\boldsymbol{\gamma}_5 = (-\alpha_1, -\alpha_2, \beta_3, -\alpha_4)$, $\boldsymbol{\gamma}_6 = (\beta_1, -\alpha_2, \beta_3, -\alpha_4)$,
$\boldsymbol{\gamma}_7 = (-\alpha_1, \beta_2, \beta_3, -\alpha_4)$, $\boldsymbol{\gamma}_8 = (\beta_1, \beta_2, \beta_3, -\alpha_4)$,
$\boldsymbol{\gamma}_9 = (-\alpha_1, -\alpha_2, -\alpha_3, \beta_4)$, $\boldsymbol{\gamma}_{10} = (\beta_1, -\alpha_2, -\alpha_3, \beta_4)$,
$\boldsymbol{\gamma}_{11} = (-\alpha_1, \beta_2, -\alpha_3, \beta_4)$, $\boldsymbol{\gamma}_{12} = (\beta_1, \beta_2, -\alpha_3, \beta_4)$,
$\boldsymbol{\gamma}_{13} = (-\alpha_1, -\alpha_2, \beta_3, \beta_4)$, $\boldsymbol{\gamma}_{14} = (\beta_1, -\alpha_2, \beta_3, \beta_4)$,
$\boldsymbol{\gamma}_{15} = (-\alpha_1, \beta_2, \beta_3, \beta_4)$, $\boldsymbol{\gamma}_{16} = (\beta_1, \beta_2, \beta_3, \beta_4)$.

B.4.2 Generator

$$\mathbf{g}_4(z_1, z_2, z_3, z_4) = \begin{matrix} [1, z_1, z_2, z_1 z_2, z_3, z_1 z_3, z_2 z_3, z_1 z_2 z_3, z_4, z_1 z_4, \\ z_2 z_4, z_1 z_2 z_4, z_3 z_4, z_1 z_3 z_4, z_2 z_3 z_4, z_1 z_2 z_3 z_4]. \end{matrix} \tag{B.15}$$

B.4.3 Fundamental Matrix and Its Inverse

- General case

$$\boldsymbol{\Omega}_4 = [\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}], \tag{B.16}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \\ -\alpha_3 & -\alpha_3 & -\alpha_3 & -\alpha_3 \\ \alpha_1\alpha_3 & -\beta_1\alpha_3 & \alpha_1\alpha_3 & -\beta_1\alpha_3 \\ \alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_3\beta_2 & -\alpha_3\beta_2 \\ -\alpha_1\alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 & \alpha_1\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ -\alpha_4 & -\alpha_4 & -\alpha_4 & -\alpha_4 \\ \alpha_1\alpha_4 & -\beta_1\alpha_4 & \alpha_1\alpha_4 & -\beta_1\alpha_4 \\ \alpha_2\alpha_4 & \alpha_2\alpha_4 & -\beta_2\alpha_4 & -\beta_2\alpha_4 \\ -\alpha_1\alpha_2\alpha_4 & \alpha_2\beta_1\alpha_4 & \alpha_1\beta_2\alpha_4 & -\beta_1\beta_2\alpha_4 \\ \alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 \\ -\alpha_1\alpha_3\alpha_4 & \beta_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \beta_1\alpha_3\alpha_4 \\ -\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_3\beta_2\alpha_4 & \alpha_3\beta_2\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\beta_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\beta_2\alpha_4 & \beta_1\alpha_3\beta_2\alpha_4 \end{bmatrix}, \tag{B.17}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \\ \beta_3 & \beta_3 & \beta_3 & \beta_3 \\ -\alpha_1\beta_3 & \beta_1\beta_3 & -\alpha_1\beta_3 & \beta_1\beta_3 \\ -\alpha_2\beta_3 & -\alpha_2\beta_3 & \beta_2\beta_3 & \beta_2\beta_3 \\ \alpha_1\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 & -\alpha_1\beta_2\beta_3 & \beta_1\beta_2\beta_3 \\ -\alpha_4 & -\alpha_4 & -\alpha_4 & -\alpha_4 \\ \alpha_1\alpha_4 & -\beta_1\alpha_4 & \alpha_1\alpha_4 & -\beta_1\alpha_4 \\ \alpha_2\alpha_4 & \alpha_2\alpha_4 & -\beta_2\alpha_4 & -\beta_2\alpha_4 \\ -\alpha_1\alpha_2\alpha_4 & \alpha_2\beta_1\alpha_4 & \alpha_1\beta_2\alpha_4 & -\beta_1\beta_2\alpha_4 \\ -\alpha_4\beta_3 & -\alpha_4\beta_3 & -\alpha_4\beta_3 & -\alpha_4\beta_3 \\ \alpha_1\alpha_4\beta_3 & -\beta_1\alpha_4\beta_3 & \alpha_1\alpha_4\beta_3 & -\beta_1\alpha_4\beta_3 \\ \alpha_2\alpha_4\beta_3 & \alpha_2\alpha_4\beta_3 & -\beta_2\alpha_4\beta_3 & -\beta_2\alpha_4\beta_3 \\ -\alpha_1\alpha_2\alpha_4\beta_3 & \alpha_2\beta_1\alpha_4\beta_3 & \alpha_1\beta_2\alpha_4\beta_3 & -\beta_1\beta_2\alpha_4\beta_3 \end{bmatrix}, \tag{B.18}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \\ -\alpha_3 & -\alpha_3 & -\alpha_3 & -\alpha_3 \\ \alpha_1\alpha_3 & -\beta_1\alpha_3 & \alpha_1\alpha_3 & -\beta_1\alpha_3 \\ \alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_3\beta_2 & -\alpha_3\beta_2 \\ -\alpha_1\alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 & \alpha_1\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ \beta_4 & \beta_4 & \beta_4 & \beta_4 \\ -\alpha_1\beta_4 & \beta_1\beta_4 & -\alpha_1\beta_4 & \beta_1\beta_4 \\ -\alpha_2\beta_4 & -\alpha_2\beta_4 & \beta_2\beta_4 & \beta_2\beta_4 \\ \alpha_1\alpha_2\beta_4 & -\alpha_2\beta_1\beta_4 & -\alpha_1\beta_2\beta_4 & \beta_1\beta_2\beta_4 \\ -\alpha_3\beta_4 & -\alpha_3\beta_4 & -\alpha_3\beta_4 & -\alpha_3\beta_4 \\ \alpha_1\alpha_3\beta_4 & -\beta_1\alpha_3\beta_4 & \alpha_1\alpha_3\beta_4 & -\beta_1\alpha_3\beta_4 \\ \alpha_2\alpha_3\beta_4 & \alpha_2\alpha_3\beta_4 & -\alpha_3\beta_2\beta_4 & -\alpha_3\beta_2\beta_4 \\ -\alpha_1\alpha_2\alpha_3\beta_4 & \alpha_2\beta_1\alpha_3\beta_4 & \alpha_1\alpha_3\beta_2\beta_4 & -\beta_1\alpha_3\beta_2\beta_4 \end{bmatrix}, \tag{B.19}$$

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \beta_1 & -\alpha_1 & \beta_1 \\ -\alpha_2 & -\alpha_2 & \beta_2 & \beta_2 \\ \alpha_1\alpha_2 & -\alpha_2\beta_1 & -\alpha_1\beta_2 & \beta_1\beta_2 \\ \beta_3 & \beta_3 & \beta_3 & \beta_3 \\ -\alpha_1\beta_3 & \beta_1\beta_3 & -\alpha_1\beta_3 & \beta_1\beta_3 \\ -\alpha_2\beta_3 & -\alpha_2\beta_3 & \beta_2\beta_3 & \beta_2\beta_3 \\ \alpha_1\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 & -\alpha_1\beta_2\beta_3 & \beta_1\beta_2\beta_3 \\ \beta_4 & \beta_4 & \beta_4 & \beta_4 \\ -\alpha_1\beta_4 & \beta_1\beta_4 & -\alpha_1\beta_4 & \beta_1\beta_4 \\ -\alpha_2\beta_4 & -\alpha_2\beta_4 & \beta_2\beta_4 & \beta_2\beta_4 \\ \alpha_1\alpha_2\beta_4 & -\alpha_2\beta_1\beta_4 & -\alpha_1\beta_2\beta_4 & \beta_1\beta_2\beta_4 \\ \beta_3\beta_4 & \beta_3\beta_4 & \beta_3\beta_4 & \beta_3\beta_4 \\ -\alpha_1\beta_3\beta_4 & \beta_1\beta_3\beta_4 & -\alpha_1\beta_3\beta_4 & \beta_1\beta_3\beta_4 \\ -\alpha_2\beta_3\beta_4 & -\alpha_2\beta_3\beta_4 & \beta_2\beta_3\beta_4 & \beta_2\beta_3\beta_4 \\ \alpha_1\alpha_2\beta_3\beta_4 & -\alpha_2\beta_1\beta_3\beta_4 & -\alpha_1\beta_2\beta_3\beta_4 & \beta_1\beta_2\beta_3\beta_4 \end{bmatrix}. \tag{B.20}$$

$$\mathbf{\Omega}_4^{-1} = \frac{1}{V_4}\left[\mathbf{P}, \mathbf{Q}\right], \tag{B.21}$$

where $V_4 = (\alpha_1 + \beta_1)(\alpha_2 + \beta_2)(\alpha_3 + \beta_3)(\alpha_4 + \beta_4) > 0$ and

$$\mathbf{P} = \begin{bmatrix} \beta_1\beta_2\beta_3\beta_4 & -\beta_2\beta_3\beta_4 & -\beta_1\beta_3\beta_4 & \beta_3\beta_4 & -\beta_1\beta_2\beta_4 & \beta_2\beta_4 & \beta_1\beta_4 & -\beta_4 \\ \alpha_1\beta_2\beta_3\beta_4 & \beta_2\beta_3\beta_4 & -\alpha_1\beta_3\beta_4 & -\beta_3\beta_4 & -\alpha_1\beta_2\beta_4 & -\beta_2\beta_4 & \alpha_1\beta_4 & \beta_4 \\ \alpha_2\beta_1\beta_3\beta_4 & -\alpha_2\beta_3\beta_4 & \beta_1\beta_3\beta_4 & -\beta_3\beta_4 & -\alpha_2\beta_1\beta_4 & \alpha_2\beta_4 & -\beta_1\beta_4 & \beta_4 \\ \alpha_1\alpha_2\beta_3\beta_4 & \alpha_2\beta_3\beta_4 & \alpha_1\beta_3\beta_4 & \beta_3\beta_4 & -\alpha_1\alpha_2\beta_4 & -\alpha_2\beta_4 & -\alpha_1\beta_4 & -\beta_4 \\ \beta_1\alpha_3\beta_2\beta_4 & -\alpha_3\beta_2\beta_4 & -\beta_1\alpha_3\beta_4 & \alpha_3\beta_4 & \beta_1\beta_2\beta_4 & -\beta_2\beta_4 & -\beta_1\beta_4 & \beta_4 \\ \alpha_1\alpha_3\beta_2\beta_4 & \alpha_3\beta_2\beta_4 & -\alpha_1\alpha_3\beta_4 & -\alpha_3\beta_4 & \alpha_1\beta_2\beta_4 & \beta_2\beta_4 & -\alpha_1\beta_4 & -\beta_4 \\ \alpha_2\beta_1\alpha_3\beta_4 & -\alpha_2\alpha_3\beta_4 & \beta_1\alpha_3\beta_4 & -\alpha_3\beta_4 & \alpha_2\beta_1\beta_4 & -\alpha_2\beta_4 & \beta_1\beta_4 & -\beta_4 \\ \alpha_1\alpha_2\alpha_3\beta_4 & \alpha_2\alpha_3\beta_4 & \alpha_1\alpha_3\beta_4 & \alpha_3\beta_4 & \alpha_1\alpha_2\beta_4 & \alpha_2\beta_4 & \alpha_1\beta_4 & \beta_4 \\ \beta_1\beta_2\alpha_4\beta_3 & -\beta_2\alpha_4\beta_3 & -\beta_1\alpha_4\beta_3 & \alpha_4\beta_3 & -\beta_1\beta_2\alpha_4 & \beta_2\alpha_4 & \beta_1\alpha_4 & -\alpha_4 \\ \alpha_1\beta_2\alpha_4\beta_3 & \beta_2\alpha_4\beta_3 & -\alpha_1\alpha_4\beta_3 & -\alpha_4\beta_3 & -\alpha_1\beta_2\alpha_4 & -\beta_2\alpha_4 & \alpha_1\alpha_4 & \alpha_4 \\ \alpha_2\beta_1\alpha_4\beta_3 & -\alpha_2\alpha_4\beta_3 & \beta_1\alpha_4\beta_3 & -\alpha_4\beta_3 & -\alpha_2\beta_1\alpha_4 & \alpha_2\alpha_4 & -\beta_1\alpha_4 & \alpha_4 \\ \alpha_1\alpha_2\alpha_4\beta_3 & \alpha_2\alpha_4\beta_3 & \alpha_1\alpha_4\beta_3 & \alpha_4\beta_3 & -\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 & -\alpha_4 \\ \beta_1\alpha_3\beta_2\alpha_4 & -\alpha_3\beta_2\alpha_4 & -\beta_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \beta_1\beta_2\alpha_4 & -\beta_2\alpha_4 & -\beta_1\alpha_4 & \alpha_4 \\ \alpha_1\alpha_3\beta_2\alpha_4 & \alpha_3\beta_2\alpha_4 & -\alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_1\beta_2\alpha_4 & \beta_2\alpha_4 & -\alpha_1\alpha_4 & -\alpha_4 \\ \alpha_2\beta_1\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \beta_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_2\beta_1\alpha_4 & -\alpha_2\alpha_4 & \beta_1\alpha_4 & -\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 & \alpha_4 \end{bmatrix}, \tag{B.22}$$

$$\mathbf{Q} = \begin{bmatrix}
-\beta_1\beta_2\beta_4 & \beta_2\beta_4 & \beta_1\beta_4 & -\beta_4 & -\beta_1\beta_2\beta_3 & \beta_2\beta_3 & \beta_1\beta_3 & -\beta_3 \\
-\alpha_1\beta_2\beta_4 & -\beta_2\beta_4 & \alpha_1\beta_4 & \beta_4 & -\alpha_1\beta_2\beta_3 & -\beta_2\beta_3 & \alpha_1\beta_3 & \beta_3 \\
-\alpha_2\beta_1\beta_4 & \alpha_2\beta_4 & -\beta_1\beta_4 & \beta_4 & -\alpha_2\beta_1\beta_3 & \alpha_2\beta_3 & -\beta_1\beta_3 & \beta_3 \\
-\alpha_1\alpha_2\beta_4 & -\alpha_2\beta_4 & -\alpha_1\beta_4 & -\beta_4 & -\alpha_1\alpha_2\beta_3 & -\alpha_2\beta_3 & -\alpha_1\beta_3 & -\beta_3 \\
\beta_1\beta_2\beta_4 & -\beta_2\beta_4 & -\beta_1\beta_4 & \beta_4 & -\beta_1\alpha_3\beta_2 & \alpha_3\beta_2 & \beta_1\alpha_3 & -\alpha_3 \\
\alpha_1\beta_2\beta_4 & \beta_2\beta_4 & -\alpha_1\beta_4 & -\beta_4 & -\alpha_1\alpha_3\beta_2 & -\alpha_3\beta_2 & \alpha_1\alpha_3 & \alpha_3 \\
\alpha_2\beta_1\beta_4 & -\alpha_2\beta_4 & \beta_1\beta_4 & -\beta_4 & -\alpha_2\beta_1\alpha_3 & \alpha_2\alpha_3 & -\beta_1\alpha_3 & \alpha_3 \\
\alpha_1\alpha_2\beta_4 & \alpha_2\beta_4 & \alpha_1\beta_4 & \beta_4 & -\alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 \\
-\beta_1\beta_2\alpha_4 & \beta_2\alpha_4 & \beta_1\alpha_4 & -\alpha_4 & \beta_1\beta_2\beta_3 & -\beta_2\beta_3 & -\beta_1\beta_3 & \beta_3 \\
-\alpha_1\beta_2\alpha_4 & -\beta_2\alpha_4 & \alpha_1\alpha_4 & \alpha_4 & \alpha_1\beta_2\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_3 & -\beta_3 \\
-\alpha_2\beta_1\alpha_4 & \alpha_2\alpha_4 & -\beta_1\alpha_4 & \alpha_4 & \alpha_2\beta_1\beta_3 & -\alpha_2\beta_3 & \beta_1\beta_3 & -\beta_3 \\
-\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 & -\alpha_4 & \alpha_1\alpha_2\beta_3 & \alpha_2\beta_3 & \alpha_1\beta_3 & \beta_3 \\
\beta_1\beta_2\alpha_4 & -\beta_2\alpha_4 & -\beta_1\alpha_4 & \alpha_4 & \beta_1\alpha_3\beta_2 & -\alpha_3\beta_2 & -\beta_1\alpha_3 & \alpha_3 \\
\alpha_1\beta_2\alpha_4 & \beta_2\alpha_4 & -\alpha_1\alpha_4 & -\alpha_4 & \alpha_1\alpha_3\beta_2 & \alpha_3\beta_2 & -\alpha_1\alpha_3 & -\alpha_3 \\
\alpha_2\beta_1\alpha_4 & -\alpha_2\alpha_4 & \beta_1\alpha_4 & -\alpha_4 & \alpha_2\beta_1\alpha_3 & -\alpha_2\alpha_3 & \beta_1\alpha_3 & -\alpha_3 \\
\alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 & \alpha_4 & \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3
\end{bmatrix}. \tag{B.23}$$

- Unity hypercube $D^4 = [0,1]^4$

$$\mathbf{\Omega}_4 = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\
0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1 \\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1 \\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\
0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1 \\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1 \\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1 \\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}, \tag{B.24}$$

$$
\mathbf{\Omega}_4^{-1} = \begin{bmatrix}
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & -1 \\
0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} . \tag{B.25}
$$

- Hypercube symmetrical around zero $D^4 = [-\alpha_1, \alpha_1] \times \ldots \times [-\alpha_4, \alpha_4]$

$$
\mathbf{\Omega}_4 = [\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}] , \tag{B.26}
$$

where

$$
\mathbf{A} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
-\alpha_1 & \alpha_1 & -\alpha_1 & \alpha_1 \\
-\alpha_2 & -\alpha_2 & \alpha_2 & \alpha_2 \\
\alpha_1\alpha_2 & -\alpha_1\alpha_2 & -\alpha_1\alpha_2 & \alpha_1\alpha_2 \\
-\alpha_3 & -\alpha_3 & -\alpha_3 & -\alpha_3 \\
\alpha_1\alpha_3 & -\alpha_1\alpha_3 & \alpha_1\alpha_3 & -\alpha_1\alpha_3 \\
\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_2\alpha_3 \\
-\alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\
-\alpha_4 & -\alpha_4 & -\alpha_4 & -\alpha_4 \\
\alpha_1\alpha_4 & -\alpha_1\alpha_4 & \alpha_1\alpha_4 & -\alpha_1\alpha_4 \\
\alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_2\alpha_4 \\
-\alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 \\
\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 \\
-\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 \\
-\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 \\
\alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4
\end{bmatrix} , \tag{B.27}
$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \alpha_1 & -\alpha_1 & \alpha_1 \\ -\alpha_2 & -\alpha_2 & \alpha_2 & \alpha_2 \\ \alpha_1\alpha_2 & -\alpha_1\alpha_2 & -\alpha_1\alpha_2 & \alpha_1\alpha_2 \\ \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 \\ -\alpha_1\alpha_3 & \alpha_1\alpha_3 & -\alpha_1\alpha_3 & \alpha_1\alpha_3 \\ -\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_2\alpha_3 \\ \alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \\ -\alpha_4 & -\alpha_4 & -\alpha_4 & -\alpha_4 \\ \alpha_1\alpha_4 & -\alpha_1\alpha_4 & \alpha_1\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_2\alpha_4 \\ -\alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 \\ -\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_3\alpha_4 \\ \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 \\ \alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 \\ -\alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 \end{bmatrix}, \tag{B.28}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \alpha_1 & -\alpha_1 & \alpha_1 \\ -\alpha_2 & -\alpha_2 & \alpha_2 & \alpha_2 \\ \alpha_1\alpha_2 & -\alpha_1\alpha_2 & -\alpha_1\alpha_2 & \alpha_1\alpha_2 \\ -\alpha_3 & -\alpha_3 & -\alpha_3 & -\alpha_3 \\ \alpha_1\alpha_3 & -\alpha_1\alpha_3 & \alpha_1\alpha_3 & -\alpha_1\alpha_3 \\ \alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_2\alpha_3 \\ -\alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ \alpha_4 & \alpha_4 & \alpha_4 & \alpha_4 \\ -\alpha_1\alpha_4 & \alpha_1\alpha_4 & -\alpha_1\alpha_4 & \alpha_1\alpha_4 \\ -\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_2\alpha_4 \\ \alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 \\ -\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_3\alpha_4 \\ \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 \\ \alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 \\ -\alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 \end{bmatrix}, \tag{B.29}$$

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\alpha_1 & \alpha_1 & -\alpha_1 & \alpha_1 \\ -\alpha_2 & -\alpha_2 & \alpha_2 & \alpha_2 \\ \alpha_1\alpha_2 & -\alpha_1\alpha_2 & -\alpha_1\alpha_2 & \alpha_1\alpha_2 \\ \alpha_3 & \alpha_3 & \alpha_3 & \alpha_3 \\ -\alpha_1\alpha_3 & \alpha_1\alpha_3 & -\alpha_1\alpha_3 & \alpha_1\alpha_3 \\ -\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_2\alpha_3 \\ \alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 & \alpha_1\alpha_2\alpha_3 \\ \alpha_4 & \alpha_4 & \alpha_4 & \alpha_4 \\ -\alpha_1\alpha_4 & \alpha_1\alpha_4 & -\alpha_1\alpha_4 & \alpha_1\alpha_4 \\ -\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_2\alpha_4 \\ \alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_1\alpha_2\alpha_4 \\ \alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_3\alpha_4 \\ -\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 \\ -\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_3\alpha_4 \end{bmatrix}. \tag{B.30}$$

$$\mathbf{\Omega}_4^{-1} = \frac{1}{16\alpha_1\alpha_2\alpha_3\alpha_4}\left[\mathbf{P}, \mathbf{Q}\right], \tag{B.31}$$

where

$$\mathbf{P} = \begin{bmatrix} \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & -\alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & -\alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & -\alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & -\alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & -\alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & -\alpha_2\alpha_4 & \alpha_1\alpha_4 \\ \alpha_1\alpha_2\alpha_3\alpha_4 & \alpha_2\alpha_3\alpha_4 & \alpha_1\alpha_3\alpha_4 & \alpha_3\alpha_4 & \alpha_1\alpha_2\alpha_4 & \alpha_2\alpha_4 & \alpha_1\alpha_4 \end{bmatrix}, \tag{B.32}$$

$$\mathbf{Q} = \begin{bmatrix}
-\alpha_4 & -\alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & 1 \\
\alpha_4 & -\alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & -1 \\
\alpha_4 & -\alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & -1 \\
-\alpha_4 & -\alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1 \\
\alpha_4 & -\alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & \alpha_1 & -1 \\
-\alpha_4 & -\alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & 1 \\
-\alpha_4 & -\alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & 1 \\
\alpha_4 & -\alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & -1 \\
-\alpha_4 & \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & \alpha_1 & -1 \\
\alpha_4 & \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & 1 \\
\alpha_4 & \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & -\alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & 1 \\
-\alpha_4 & \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & -\alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & -1 \\
\alpha_4 & \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & -\alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & -\alpha_1 & 1 \\
-\alpha_4 & \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & \alpha_2 & -\alpha_1 & -1 \\
-\alpha_4 & \alpha_1\alpha_2\alpha_3 & -\alpha_2\alpha_3 & \alpha_1\alpha_3 & -\alpha_3 & \alpha_1\alpha_2 & -\alpha_2 & \alpha_1 & -1 \\
\alpha_4 & \alpha_1\alpha_2\alpha_3 & \alpha_2\alpha_3 & \alpha_1\alpha_3 & \alpha_3 & \alpha_1\alpha_2 & \alpha_2 & \alpha_1 & 1
\end{bmatrix} . \quad (B.33)$$

For $n \geq 5$ it is preferred to generate formulas recurrently using symbolic computations on a computer.

Appendix C
Proofs of Theorems, Remarks and Algorithms

C.1 Proof of Remark 3.2

Proof. First we prove (3.2).

(1) From (2.43), (A.5) and (A.3) we have

$$\begin{aligned}
\mathbf{\Omega}_{k+1}\mathbf{\Omega}_{k+1}^T &= \left(\begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix} \otimes \mathbf{\Omega}_k\right)\left(\begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix} \otimes \mathbf{\Omega}_k\right)^T \\
&= \left(\begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix} \otimes \mathbf{\Omega}_k\right)\left(\begin{bmatrix} 1 & 1 \\ -\alpha_{k+1} & \beta_{k+1} \end{bmatrix}^T \otimes \mathbf{\Omega}_k^T\right) \\
&= \begin{bmatrix} 2 & \beta_{k+1}-\alpha_{k+1} \\ \beta_{k+1}-\alpha_{k+1} & \alpha_{k+1}^2+\beta_{k+1}^2 \end{bmatrix} \otimes \mathbf{\Omega}_k\mathbf{\Omega}_k^T, \qquad \text{(C.1)}
\end{aligned}$$

for $k = 0, 1, 2, \ldots, n-1$. This ends the proof of the first part of Remark 3.2.

(2) Now we prove the orthogonality condition: $\beta_k = \alpha_k$ for $k = 1, \ldots, n$. According to the equation (C.1) we see that $\prod_{k=1}^{n}(\beta_k - \alpha_k)$ is the element in the first row and the last column of the matrix $\mathbf{\Omega}_{k+1}\mathbf{\Omega}_{k+1}^T$, $(k = 0, 1, 2, \ldots, n-1)$. By using recurrence we conclude that the necessary condition under which the rows of the matrix $\mathbf{\Omega} = \mathbf{\Omega}_n$ are orthogonal is

$$\prod_{i=1}^{k}(\beta_i - \alpha_i) = 0$$

for $k = 1, 2, \ldots, n$, where n is the number of system inputs.

Now we prove the sufficient condition. In this case $\beta_k = \alpha_k$ holds for $k = 1, \ldots, n$. According to (C.1)

$$\mathbf{\Omega}_{k+1}\mathbf{\Omega}_{k+1}^T = 2\begin{bmatrix} 1 & 0 \\ 0 & \alpha_{k+1}^2 \end{bmatrix} \otimes \mathbf{\Omega}_k\mathbf{\Omega}_k^T, \qquad k = 0, 1, 2, \ldots, n-1.$$

holds. Using the above recurrency we obtain that $\beta_k = \alpha_k$ is a sufficient condition for orthogonality of $\mathbf{\Omega}$. This ends the proof of the second part of Remark 3.2. □

C.2 Proof of Remark 3.3

Proof. Let us take the following notation

$$\mathbf{\Lambda}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_1 \end{bmatrix} \otimes 1 = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_1 \end{bmatrix},$$

$$\mathbf{\Lambda}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \otimes \mathbf{\Lambda}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 \\ 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & \alpha_1\alpha_2 \end{bmatrix},$$

$$\mathbf{\Lambda}_3 = \begin{bmatrix} 1 & 0 \\ 0 & \alpha_3 \end{bmatrix} \otimes \mathbf{\Lambda}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_1\alpha_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_1\alpha_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_2\alpha_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_1\alpha_2\alpha_3 \end{bmatrix},$$

and so forth. This means that the recurrence (3.4) holds. By induction we obtain

$$\mathbf{\Omega}_1\mathbf{\Omega}_1^T = 2^1\mathbf{\Lambda}_1^2,$$

$$\mathbf{\Omega}_2\mathbf{\Omega}_2^T = 2^2\mathbf{\Lambda}_2^2,$$

$$\vdots$$

$$\mathbf{\Omega}_k\mathbf{\Omega}_k^T = 2^k\mathbf{\Lambda}_k^2 \qquad \text{for} \qquad k = 1, \ldots, n.$$

Thus, we can neglect the subscripts, i.e. $\mathbf{\Omega}_n = \mathbf{\Omega}$ and $\mathbf{\Lambda}_n = \mathbf{\Lambda}$ and simply write

$$\mathbf{\Omega}\mathbf{\Omega}^T = 2^n\mathbf{\Lambda}^2.$$

Taking into account popular features of matrix calculus such, as $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$, $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ and $c\mathbf{A} = \mathbf{A}c$ for $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$ and $c \in \mathbb{R}$, after simple transformations we obtain

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{\Omega}\boldsymbol{\Omega}^T\left(\boldsymbol{\Lambda}^{-1}\right)^T = 2^n\mathbf{I}.$$

Taking into account symmetry of $\boldsymbol{\Lambda}$ we obtain

$$\left(2^{-\frac{n}{2}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Omega}\right)\left(2^{-\frac{n}{2}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Omega}\right)^T = \mathbf{I}.$$

This means that

$$\left(2^{-\frac{n}{2}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Omega}\right)^{-1} = \left(2^{-\frac{n}{2}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Omega}\right)^T,$$

or equivalently

$$\boldsymbol{\Omega}^{-1}\boldsymbol{\Lambda}2^{\frac{n}{2}} = \boldsymbol{\Omega}^T\boldsymbol{\Lambda}^{-1}2^{-\frac{n}{2}},$$

and finally

$$\boldsymbol{\Omega}^{-1} = 2^{-n}\boldsymbol{\Omega}^T\boldsymbol{\Lambda}^{-2}.$$

This ends the proof of Remark 3.3. □

C.3 Proof of Corollary 5.27

Proof. First we prove (5.176). Let us define the generator by

$$\mathbf{g} = \left[1,\ z_1,\ z_2,\ \ldots,\ z_{n-1},\ z_n,\ z_1z_2,\ z_1z_3,\ \ldots,\ \prod_{i=1}^{n} z_i\right]^T,$$

and the corresponding fundamental matrix by $\boldsymbol{\Omega} = \left[\mathbf{g}\left(\boldsymbol{\gamma}_1\right) \ldots \mathbf{g}\left(\boldsymbol{\gamma}_{2^n}\right)\right]$. The linear mapping f in (5.175) is a special case of the function f_0 given by (2.26), where the vector $\boldsymbol{\theta}$ is of the form: $\boldsymbol{\theta} = [0, r_1, r_2, \ldots, r_n, 0, \ldots, 0]^T$. After filling the matrix $\boldsymbol{\Omega}$ for the above generator with the vectors $\boldsymbol{\gamma}_v \in \Gamma^n$, from (2.30) we obtain

$$\mathbf{q} = \begin{bmatrix} 1 & -\alpha_1 & \cdots & -\alpha_n & * & \cdots & * \\ 1 & -\alpha_1 & \cdots & \beta_n & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta_1 & \ldots & \beta_n & * & \cdots & * \end{bmatrix}\boldsymbol{\theta},$$

where $\boldsymbol{\theta} = [0, r_1, r_2, \ldots, r_n, 0, \ldots, 0]^T$ and the symbols "$*$" are nonzero elements depending on α_i and β_j. Thus,

$$\mathbf{q} = \begin{bmatrix} 0 & \boldsymbol{\gamma}_1^T & 0 \cdots & 0 \\ 0 & \boldsymbol{\gamma}_2^T & 0 \cdots & 0 \\ \vdots & \vdots & \vdots \ddots & \vdots \\ 0 & \boldsymbol{\gamma}_{2^n}^T & 0 \cdots & 0 \end{bmatrix}\begin{bmatrix} 0 \\ \mathbf{r} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{L}^T\mathbf{r}.$$

This ends the proof of (5.176).

On the other hand, according to (2.47) we have $S = f$ if and only if the consequents of the rules are $q_v = f(\gamma_v) = \mathbf{r}^T \gamma_v$ for every $v = 1, \ldots, 2^n$. Thus,

$$q_v = q_{(i_1, i_2, \ldots, i_n)} = \mathbf{r}^T \gamma_v \ ,$$

where $v \leftrightarrow (i_1, \ldots, i_n)$ as in (2.16). Now, if we take into account (2.23), the result in (5.177) is clear. Both (5.177) and (5.176) for the consequents of the rules are the necessary and sufficient conditions which guarantee linearity of the P1-TS system. This ends the proof of Corollary 5.27. □

C.4 Proof of RLS Algorithm from Section 6.4

Proof. Without loss of generality we assume a new simplified notation in which the index j will be neglected, since all computations should be performed for all inputs. For simplicity we will take a notation as shown in Table C.1

Table C.1 Simplified notation for the proof of the algorithm from Section 6.4

Old notation	Number of equation	Simplified notation
$\mathbf{w}(t_k)$	(6.11)	$\mathbf{w}_k$
$d_j(t_k, t_{k+1})$	(6.7)	d_k
$\epsilon_j(t_k, t_{k+1})$	(6.14)	ϵ_k

The gradient of (6.32) with respect to $\mathbf{q}$ must be zero vector

$$\nabla_{\mathbf{q}} E_j(\lambda) = 2 \sum_{k=1}^{K} \lambda^{K-k} \left(\mathbf{w}_k^T \mathbf{q} - d_k \right) \mathbf{w}_k = \mathbf{0}. \tag{C.2}$$

The vector of the consequents $\mathbf{q}$ that satisfies the equation (C.2), we will denote by $\mathbf{q}_K$, since it is computed for the given K data pairs from the available set (6.31). The normal equations are as follows

$$\underbrace{\sum_{k=1}^{K} \lambda^{K-k} d_k \mathbf{w}_k}_{\mathbf{r}_K} = \underbrace{\sum_{k=1}^{K} \lambda^{K-k} \left(\mathbf{w}_k \mathbf{w}_k^T \right)}_{\mathbf{R}_K} \cdot \mathbf{q}_K \ , \tag{C.3}$$

or equivalently

$$\mathbf{q}_K = \mathbf{R}_K^{-1} \mathbf{r}_K \ , \tag{C.4}$$

where

$$\mathbf{r}_K = \lambda \sum_{k=1}^{K-1} \lambda^{K-1-k} d_k \mathbf{w}_k + d_k \mathbf{w}_k = \lambda \mathbf{r}_{K-1} + d_K \mathbf{w}_K \ , \tag{C.5}$$

$$\mathbf{R}_K = \lambda \sum_{k=1}^{K-1} \lambda^{K-1-k} \mathbf{w}_k \mathbf{w}_k^T + \mathbf{w}_k \mathbf{w}_k^T = \lambda \mathbf{R}_{K-1} + \mathbf{w}_K \mathbf{w}_K^T \ . \tag{C.6}$$

Now we consider the Sherman-Morrison-Woodbury matrix identity [161]

$$\left(\mathbf{A} + \mathbf{X}_1 \mathbf{B} \mathbf{X}_2^T\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X}_1 \left(\mathbf{B}^{-1} + \mathbf{X}_2^T \mathbf{A}^{-1} \mathbf{X}_1\right)^{-1} \mathbf{X}_2^T \mathbf{A}^{-1}, \tag{C.7}$$

which holds for matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{X}_1$ and $\mathbf{X}_2$, where the first two are non-singular. From (C.6)-(C.7) by $\mathbf{A} = \lambda \mathbf{R}_{K-1}$, $\mathbf{X}_1 = \mathbf{X}_2 = \mathbf{w}_K$, $\mathbf{B} = 1$ we obtain

$$\mathbf{R}_K^{-1} = \frac{1}{\lambda}\mathbf{R}_{K-1}^{-1} - \underbrace{\frac{1}{\lambda}\mathbf{R}_{K-1}^{-1}\mathbf{w}_K \left(1 + \mathbf{w}_K^T \frac{1}{\lambda}\mathbf{R}_{K-1}^{-1}\mathbf{w}_K\right)^{-1}}_{\mathbf{h}_K} \mathbf{w}_K^T \frac{1}{\lambda}\mathbf{R}_{K-1}^{-1} \ .$$

By assuming $\mathbf{R}_K^{-1} = \mathbf{P}_K$ (the inverse correlation matrix), the last equation can be written as

$$\mathbf{P}_K = \frac{1}{\lambda}\left(\mathbf{P}_{K-1} - \mathbf{h}_K \mathbf{w}_K^T \mathbf{P}_{K-1}\right), \tag{C.8}$$

where

$$\mathbf{h}_K = \frac{\mathbf{P}_{K-1}\mathbf{w}_K}{\lambda + \mathbf{w}_K^T \mathbf{P}_{K-1}\mathbf{w}_K} \ , \tag{C.9}$$

and $\mathbf{h}_K$ is called the Kalman gain vector. From (C.4) and (C.5) we have

$$\mathbf{q}_K = \mathbf{P}_K \mathbf{r}_K = \lambda \mathbf{P}_K \mathbf{r}_{K-1} + d_K \mathbf{P}_K \mathbf{w}_K \ . \tag{C.10}$$

From (C.9) we obtain

$$\mathbf{P}_{K-1}\mathbf{w}_K = \lambda \mathbf{h}_K + \mathbf{h}_K \mathbf{w}_K^T \mathbf{P}_{K-1}\mathbf{w}_K \ .$$

After multiplying (C.8) by $\mathbf{w}_K$ we get

$$\mathbf{P}_K \mathbf{w}_K = \frac{1}{\lambda}\left(\mathbf{P}_{K-1}\mathbf{w}_K - \mathbf{h}_K \mathbf{w}_K^T \mathbf{P}_{K-1}\mathbf{w}_K\right).$$

Thus,

$$\mathbf{P}_K \mathbf{w}_K = \frac{1}{\lambda}\left(\lambda \mathbf{h}_K + \mathbf{h}_K \mathbf{w}_K^T \mathbf{P}_{K-1}\mathbf{w}_K - \mathbf{h}_K \mathbf{w}_K^T \mathbf{P}_{K-1}\mathbf{w}_K\right) = \mathbf{h}_K. \tag{C.11}$$

From (C.8) we have $\lambda\mathbf{P}_K = \mathbf{P}_{K-1} - \mathbf{h}_K\mathbf{w}_K^T\mathbf{P}_{K-1}$. Let us multiply this equation by $\mathbf{r}_{K-1}$ and add $d_K\mathbf{P}_K\mathbf{w}_K = d_K\mathbf{h}_K$. We obtain

$$\lambda\mathbf{P}_K\mathbf{r}_{K-1} + d_K\mathbf{h}_K = \mathbf{P}_{K-1}\mathbf{r}_{K-1} - \mathbf{h}_K\mathbf{w}_K^T\mathbf{P}_{K-1}\mathbf{r}_{K-1} + d_K\mathbf{h}_K. \quad \text{(C.12)}$$

Taking into account (C.10), (C.12) and (C.11), we obtain the consequents vector

$$\mathbf{q}_K = \lambda\mathbf{P}_K\mathbf{r}_{K-1} + d_K\mathbf{P}_K\mathbf{w}_K = \mathbf{P}_{K-1}\mathbf{r}_{K-1} - \mathbf{h}_K\mathbf{w}_K^T\mathbf{P}_{K-1}\mathbf{r}_{K-1} + d_K\mathbf{h}_K. \quad \text{(C.13)}$$

According to (C.4) the equation $\mathbf{P}_{K-1}\mathbf{r}_{K-1} = \mathbf{q}_{K-1}$ holds. Finally, from (C.13) we get

$$\mathbf{q}_K = \mathbf{q}_{K-1} - \mathbf{h}_K\mathbf{w}_K^T\mathbf{q}_{K-1} + d_K\mathbf{h}_K \ . \quad \text{(C.14)}$$

In the RLS algorithm we take k instead of K because of the data inflow. Taking into account the notation from Table C.1 we conclude that:

- the vector (C.9) is the same as the Kalman gain vector in (6.34),
- the equation (C.14) is equivalent to two equations: (6.33) and (6.35).

This completes the proof of RLS algorithm from Section 6.4. □

References

1. Abe, S.: Support Vector Machines for Pattern Classification. Series: Advances in Pattern Recognition. Springer, Berlin (2005)
2. Abe, S., Lan, M.S.: A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. IEEE Transactions on Fuzzy Systems 3(1), 18–28 (1995)
3. Abe, S., Thawonmas, R.: A fuzzy classifier with ellipsoidal regions. IEEE Transactions on Fuzzy Systems 5(3), 358–368 (1997)
4. Abonyi, J.: Fuzzy Model Identification for Control. Birkhäuser, Boston (2003)
5. Alcalá, R., Alcalá-Fdez, J., Casillas, J., Cordón, O., Herrera, F.: Hybrid learning models to get the interpretability-accuracy trade-off in fuzzy modeling. Soft Computing 10, 717–734 (2006)
6. Alekseichuk, A.N.: On complexity of computation of partial derivatives of Boolean functions realized by Zhegalkin polynomials. Cybernetics and Systems Analysis 37(5), 648–653 (2001)
7. Aly, S.: Fuzzy analogical gates for separation sequence synthesis. Chemical Engineering and Processing 36, 209–217 (1997)
8. Anderson, B.D.O., Moore, J.B.: Optimal Control: Linear Quadratic Methods. Prentice-Hall, Upper Saddle River (1990)
9. Astola, J.T., Stanković, R.S.: Fundamentals of Switching Theory and Logic Design. Springer, Heidelberg (2006)
10. Athans, M.: Debate with Prof. Lotfi A. Zadeh. In: 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT 1999), Aachen, Germany (September 1999)
11. Ayari, I., Chatti, A.: Reactive Control Using Behavior Modelling of a Mobile Robot. International Journal of Computers, Communications and Control II(3), 217–228 (2007)
12. Babuška, R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston (1998)
13. Babuška, R., Verbruggen, H.B.: An overview of fuzzy modeling for control. Control Engineering Practice 4, 1593–1606 (1996)
14. Bellman, R.E.: Adaptive Control Processes. Princeton University Press, Princeton (1961)
15. Ben-Israel, A., Greville, T.N.E.: Generalized Inverses: Theory and Applications. Wiley, New York (1977)

16. Beom, H.R., Cho, H.S.: A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning. IEEE Transactions on Systems, Man and Cybernetics 25(3), 464–477 (1995)
17. Boutayeb, M., Aubury, D.: A strong tracking extended Kalman observer for nonlinear discrete-time systems. IEEE Transactions on Automatic Control 44(8), 1550–1556 (1999)
18. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM Studies in Applied Mathematics, vol. 15. SIAM, Philadelphia (1994)
19. Brown, S.D., Vranesic, Z.G.: Fundamentals of Digital Logic with VHDL Design, 2nd edn. McGraw-Hill, New York (2005)
20. Cao, S.G., Rees, N.W., Feng, G.: Analysis and design for a class of complex control systems part II: Fuzzy controller design. Automatica 33(6), 1029–1039 (1997)
21. Casillas, J., Cordón, H.F., Magdalena, L. (eds.): Interpretability Issues in Fuzzy Modeling. Series: Studies in Fuzziness and Soft Computing, vol. 128. Springer, Heidelberg (2003)
22. Castro, J.L., Flores-Hidalgo, L.D., Mantas, C.J., Puche, J.M.: Extraction of fuzzy rules from support vector machines. Fuzzy Sets and Systems 158, 2057–2077 (2007)
23. Chen, C.G., Ueta, T.: Yet another chaotic attractor. International Journal of Bifurcation and Chaos 9, 1465–1466 (1999)
24. Chen, Y., Wang, J.-Z.: Support vector learning for fuzzy-rulebased classification systems. IEEE Transactions on Fuzzy Systems 11, 716–728 (2003)
25. Cheng, C., Chiu, M.-S.: A new data-based methodology for nonlinear process modeling. Chemical Engineering Science 59, 2801–2810 (2004)
26. Choi, B., Tipnis, K.: New Components for Building Fuzzy Logic Circuits. In: Proc. 14th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD2007), Haikou, August 24-27, 2007, pp. 586–590. IEEE Comp. Society, Hainan (2007)
27. Chu, Y.-C., Glover, K.: Bounds of the induced norm and model reduction errors for systems with repeated scalar nonlinearities. IEEE Transactions on Automatic Control 44, 471–483 (1999)
28. Chu, Y.-C., Glover, K.: Stabilization and performance synthesis for systems with repeated scalar nonlinearities. IEEE Transactions on Automatic Control 44, 484–496 (1999)
29. Ciupe, S.M., Ribeiro, R.M., Nelson, P.W., Perelson, A.S.: Modeling the mechanisms of acute hepatitis B virus infection. Journal of Theoretical Biology 247, 23–35 (2007)
30. Cohen, J.S.: Computer Algebra and Symbolic Computation: Mathematical Methods. AK Peters, Ltd., Har/Cdr ed. (2003)
31. Collins, E.G.: Book reviews - Fuzzy control systems design and analysis: a linear matrix inequality approach. Tanaka, K., Wang H. O. (eds.) Automatica 39, 2011–2019 (2003)
32. Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems 141(1), 5–31 (2004)
33. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
34. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)

35. Doyle, F.J., Ogunnaike, A., Pearson, R.K.: Nonlinear model-based control using second-order Volterra models. Automatica 31, 697–714 (1995)
36. Driankov, D., Palm, R. (eds.): Advances in Fuzzy Control. Physica-Verlag, Heidelberg (1998)
37. Dubois, D., Nguyen, H.T., Prade, H., Sugeno, M.: Introduction: The real contribution of fuzzy systems. In: Nguyen, H.T., Sugeno, M. (eds.) Fuzzy Systems: Modeling and Control, pp. 1–17. Kluwer, Norwell (1998)
38. Dubois, D., Prade, H.: Fuzzy logics and the generalized modus ponens revisited. Cybernetics and Systems 15, 293–331 (1984)
39. Duch, W., Setiono, R., Żurada, J.M.: Computational Intelligence Methods for Rule-Based Data Understanding. Proceedings of the IEEE 92(5), 771–805 (2004)
40. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience, Hoboken (2000)
41. Elkan, C.: The Paradoxical Success of Fuzzy Logic. In: IEEE Expert, August 1994, pp. 3–8 (1994); With fifteen responses on pp. 9–46; First version in AAAI 1993 proceedings, pp. 698–703
42. Espinosa, J., Vandewalle, J., Wertz, V.: Fuzzy Logic, Identification and Predictive Control. Springer, London (2005)
43. Eves, H.W.: Elementary Matrix Theory. Dover Publications, Mineola (1980)
44. Evgeniou, T., Pontil, M., Poggio, T.: Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13(1), 1–50 (2000)
45. Eykhoff, P.: System Identification - Parameter and State Estimation. John Wiley and Sons, New York (1974)
46. Falkowski, B.J.: A Note on the Polynomial Form of Boolean Functions and Related Topics. IEEE Transactions on Computers 48(8), 860–864 (1999)
47. Floreano, D., Mondada, F.: Automatic Creation of an Autonomus Agent: Genetic Evolution of a Neural Network Driven Robot. In: 3rd Int. Conf. Simulation of Adaptive Behavior, Brighton (1994)
48. Franke, D.: A linear state space approach to a class of discrete-event systems. Mathematics and Computers in Simulation 39, 499–503 (1995)
49. Gabor, D., Wilby, W.P.L., Woodcock, R.: A universal nonlinear filter, simulator and predictor which optimizes itself by learning process. IEE Proceedings, Part B 108, 422–433 (1961)
50. Gadaras, I., Mikhailov, L., Lekkas, S.: Generation of Fuzzy Classification Rules Directly from Overlaping Input Partitioning. In: IEEE International Conference on Fuzzy Systems, July 24-26, 2007, pp. 1–6 (2007), doi:10.1109/FUZZY.2007.4295424
51. Galichet, S., Boukezzoula, R., Foulloy, L.: Explicit analytical formulation and exact inversion with singleton consequents. Fuzzy Sets and Systems 146, 421–436 (2004)
52. Gaweda, A.E., Żurada, J.M.: Data-Driven Linguistic Modeling Using Relational Fuzzy Rules. IEEE Transactions on Fuzzy Systems 11(1), 121–134 (2003)
53. Gorzałczany, M.B.: Computational Intelligence Systems and Applications: Neuro-Fuzzy and Fuzzy Neural Synergisms. Series: Studies in Fuzziness and Soft Computing. Physica-Verlag, Heidelberg (2002)
54. Graham, A.: Kronecker products and matrix calculus with applications. Halsted Press, John Wiley and Sons, NY (1981)

55. Gniewek, L., Kluska, J.: Family of fuzzy J-K flip-flops based on bounded product, bounded sum and complementation. IEEE Transactions on Systems, Man and Cybernetics, Part B 28(6), 861–868 (1998)
56. Gniewek, L., Kluska, J.: Hardware implementation of fuzzy Petri net as a controller. IEEE Transactions on Systems, Man and Cybernetics, Part B 34(3), 1315–1324 (2004)
57. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, Los Altos (2000)
58. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, New York (1994)
59. Hirota, K. (ed.): Industrial Applications of Fuzzy Technology. Springer, Tokyo (1993)
60. Hirota, K., Ozawa, K.: Algebraic fuzzy flip-flop circuits. Fuzzy Sets and Systems 39, 215–226 (1991)
61. Holte, R.: Very simple classification rules perform well in most common used data sets. Machine Learning 11, 63–91 (1993)
62. Hu, Y.-C.: Determining membership functions and minimum fuzzy support in finding fuzzy association rules for classification problems. Knowledge-Based Systems 19, 57–66 (2006)
63. Hu, Y.-C., Chen, R.-S., Tzeng, G.-H.: Finding fuzzy classification rules using data mining techniques. Pattern Recognition Letters 24, 509–519 (2003)
64. Hubbard, J.H., West, B.H.: Differential Equations: A Dynamical Systems Approach, Part II: Higher-Dimensional Systems. Springer, New York (1995)
65. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy trade-off of fuzzy systems by multiobjective fuzzy genetics-based machine learning. International Journal on Approximate Reasoning 44(1), 4–31 (2007)
66. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets and Systems 52(1), 21–32 (1992)
67. Ishibuchi, H., Yamamoto, T.: Rule weight specification in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 13(4), 428–435 (2005)
68. Ivakhenko, A.G.: Polynomial theory of complex systems. IEEE Transactions on Systems, Man and Cybernetics 1(4), 364–378 (1971)
69. Jahromi, M.Z., Taheri, M.: A proposed method for learning rule weights in fuzzy rule-based classification systems. Fuzzy Sets and Systems 159, 449–459 (2008)
70. Janczak, A.: Identification of Nonlinear Systems Using Neural Networks and Polynomial Models; A Block-Oriented Approach. Springer, Heidelberg (2005)
71. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River (1997)
72. Jayadeva, Khemchandani, R., Chandra, S.: Fuzzy multi-category proximal support vector classification via generalized eigenvalues. Soft Computing 11(7), 679–685 (2007)
73. Jeleń, L., Fevens, T., Krzyżak, A.: Classification of breast cancer malignancy using cytological images of fine needle aspiration biopsies. International Journal of Applied Mathematics and Computer Science 18(1), 75–83 (2008)
74. Jia, X., Nanning, Z.: Reliable fuzzy control with domain guaranteed cost for fuzzy systems with actuator failures. Progress in Natural Science 14(12), 1109–1115 (2004)

75. Jin, L., Nikiforuk, P.N., Gupta, M.M.: Approximation of discrete-time state-space trajectories using dynamic recurrent neural networks. IEEE Transactions on Automatic Control 40, 1266–1270 (1995)
76. Joachims, T.: Making large-scale svm learning practical. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods - Support Vector Learning, pp. 169–185. MIT Press, Cambridge (1999)
77. Jou, L.-J., Liao, C.-M., Chiu, Y.-C.: A Boolean algebra algorithm suitable for use in temperature-humidity control of a grafted seedling acclimatization chamber. Computers and Electronics in Agriculture 48, 1–18 (2005)
78. Juang, C.-F., Chiu, S.-H., Chang, S.-W.: A Self-Organizing TS-Type Fuzzy Network With Support Vector Learning and its Application to Classification to Classification Problems. IEEE Transactions on Fuzzy Systems 15(3), 998–1008 (2007)
79. Kacprzyk, J.: Multistage Decision-Making under Fuzziness: Theory and Applications. Verlag TUV, Köln (1983)
80. Kacprzyk, J.: On measuring the specificity of IF-THEN rules. International Journal on Approximate Reasoning 11(1), 29–53 (1994)
81. Kacprzyk, J.: Multistage Fuzzy Control: A Model-Based Approach to Fuzzy Control and Decision Making. John Wiley and Sons, Chichester (1997)
82. Kaczorek, T.: Vectors and matrices in control and electrical engineering (in Polish: Wektory i macierze w automatyce i elektrotechnice). WNT, Warsaw (1999)
83. Kailath, T., Sayed, A.H., Hassibi, B.: Linear estimation. Prentice-Hall, Englewood Cliffs (2000)
84. Kandel, A., Langholz, G. (eds.): Fuzzy Hardware: Architectures and Applications. Springer, Heidelberg (1997)
85. Karnaugh, M.: The Map Method for Synthesis of Combinational Logic Circuits. Transactions of the AIEE, Part I 72(9), 593–599 (1953)
86. Kaszkurewicz, E., Bhaya, A.: Matrix Diagonal Stability in Systems and Computation. Birkhäuser, Boston (1999)
87. Kecman, V.: Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT Press, Cambridge (2001)
88. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man and Cybernetics 15(4), 580–585 (1985)
89. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice-Hall, Englewood Cliffs (2001)
90. Khiar, D., Lauber, J., Floquet, T., Colin, G., Guerra, T.M., Chamaillard, Y.: Robust Takagi-Sugeno fuzzy control of a spark ignition engine. Control Engineering Practice 15, 1446–1456 (2007)
91. Kim, D.-W., Park, J.-B., Joo, Y.-H.: Effective digital implementation of fuzzy control systems based on approximate discrete-time models. Automatica 43, 1671–1683 (2007)
92. Kluska, J.: Fuzzy Control (in Polish: Sterowanie z logika rozmyta). Zeszyty Naukowe Politechniki Rzeszowskiej, 104(12), Rzeszów (1992)
93. Kluska, J.: A modified fuzzy controller. Systems Science 1(1), 37–50 (1995)
94. Kluska, J.: Stability analysis of SISO, MISO and MIMO continuous fuzzy control systems based on theorems of Kudrewicz. Archives of Control Sciences 5(XLI)(1-2), 5–37 (1996)

95. Kluska, J.: Stability analysis of fuzzy control systems containing PID - fuzzy controller. In: Proc. Int. Conf. Fuzzy Logic in Engineering and Natural Sciences FUZZY 1996, Zittau, Germany, September 25-27, 1996, pp. 106–113 (1996)
96. Kluska, J.: Robustness of fuzzy control systems with respect to plant parameters modifications. Archives of Control Sciences 6(XLII)(3-4), 125–138 (1997)
97. Kluska, J.: Absolute stability of continuous fuzzy control systems. In: Aracil, J., Gordillo, F. (eds.) Stability Issues in Fuzzy Control. Series: Studies in Fuzziness and Soft Computing, vol. 44, pp. 15–45. Springer, Heidelberg (2000)
98. Kluska, J.: New results in Analytical Modeling Using Takagi-Sugeno Expert System, International Seminar on Intelligent Systems, Warsaw 2004. In: Dramiński, M., Grzegorzewski, P., Trojanowski, K., Zadrożny, S. (eds.) Issues in Intelligent Systems: Models and Techniques, EXIT, Warsaw, pp. 79–94 (2005)
99. Kluska, J.: Identification of multilinear dynamical systems using Takagi-Sugeno rule scheme. In: Proc. of the 13th East West Fuzzy Colloquium, Wissenschaftliche Berichte, Heft 89/2006, No. 2180-2226, Zittau, September 13-15, 2006, pp. 14–25 (2006)
100. Kluska, J.: Transformation lemma on analytical modeling via takagi-sugeno fuzzy system and its applications. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS, vol. 4029, pp. 230–239. Springer, Heidelberg (2006) (Carbonell, J. G., Siekmann J.(ser. eds.)
101. Kluska, J., Gniewek, L.: New method of fuzzy Petri net synthesis and its application for fuzzy control systems design. In: Proc. of the 7th International Zittau Fuzzy Colloquium, 8-10.09, Zittau, Germany (1999)
102. Kluska, J., Gniewek, L.: A New Method of Fuzzy Petri Net Synthesis and its Application for Fuzzy Control Systems Design. In: Hampel, R., Wagenknecht, M., Chaker, N. (eds.) Fuzzy Control. Series: Advances in Soft Computing, pp. 222–227. Physica-Verlag, Heidelberg (2000)
103. Kluska, J., Gniewek, L.: Fuzzy Petri nets as control systems. Pomiary Automatyka Kontrola 1, Warsaw, 7–10 (2005)
104. Kluska, J., Hajduk, Z.: Hardware implementation of a fuzzy Petri net based on VLSI digital circuits. In: Proc. of the 3rd European Society for Fuzzy Logic and Technology, September 10-12, 2003, pp. 789–793 (2003)
105. Kluska, J., Hajduk, Z.: Digital implementation of fuzzy petri net based on asynchronous fuzzy RS flip-flop. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 314–319. Springer, Heidelberg (2004)
106. Kluska, J., Hajduk, Z., Gniewek, L.: Synthesis of Fuzzy Petri Nets as Hardware Control and Diagnostic Systems. Pomiary Automatyka Kontrola 6bis, 3–5, Warsaw (2006)
107. Kluska, J., Wiktorowicz, K.: Automatic generating of fuzzy control rules for a fuzzy logic controller using Matlab. Systems Science 23(1), 89–99 (1997)
108. Kluska, J., Wiktorowicz, K.: Integral evaluations for a class of multivariable continuous fuzzy control systems. In: Proc. of the 15th IFAC World Congress, Barcelona, July 21-26, 2002, pp. 1–6 (2002)
109. Korbicz, J., Kościelny, J.M., Kowalczuk, Z., Cholewa, W. (eds.): Fault Diagnosis. Models, Artificial Intelligence, Applications. Springer, Heidelberg (2004)
110. Kosko, B.: Neural Networks and Fuzzy Systems. A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Inc., Englewood Cliffs (1992)

111. Krstić, M., Tsiotras, P.: Inverse Optimal Stabilization of a Rigid Spacecraft. IEEE Transactions on Automatic Control 44(5), 1042–1049 (1999)
112. Kumar, M., Stoll, N., Stoll, R.: An energy-gain bounding approach to robust fuzzy identification. Automatica 42, 711–721 (2006)
113. Kuncheva, L.I., Bezdek, J.C.: A fuzzy generalized nearest prototype classifier. Proc. 7th IFSA World Congress 3(1), 217–222 (1997)
114. Landau, I.D., Lozano, R., M'Saad, M.: Adaptive Control. Springer, New York (1998)
115. Langholz, G., Kandel, A., Mott, J.L.: Foundations of digital logic design. World Scientific, Singapore (1998)
116. Leski, J., Czogala, E.: A new artificial neural networks based fuzzy inference system with moving consequents in if-then rules and selected applications. Fuzzy Sets and Systems 108, 289–297
117. Lewis, F.L., Liu, K.: Towards a Paradigm for Fuzzy Logic Control. Automatica 32(2), 167–181 (1996)
118. Lichtenberg, G., Faisal, S., Werner, H.: Ein Ansatz zur dynamischen Modellierung der Genexpression mit Shegalkin-Polynomen. Automatisierungstechnik (12), 589–596 (2005)
119. Lin, C.T., Yeh, C.M., Liang, S.F., Chung, J.F., Kumar, N.: Support-vector based fuzzy neural network for pattern classification. IEEE Transactions on Fuzzy Systems 14(1), 31–41 (2006)
120. Lovassy, R., Kóczy, L.T.: S-Shaped Fuzzy Flip-Flops. In: Proc. of 8th Int. Symp. of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, November 15-17, 2007, pp. 383–390 (2007)
121. Lü, J., Zhang, S.: Controlling Chen's chaotic attractor using backstepping design based on parameters identification. Physics Letters A 286, 148–152 (2001)
122. Lü, J., Cheng, G., Cheng, D.: A new chaotic system and beyond: the generalized Lorentz-like system. International Journal of Bifurcation and Chaos 14(5), 1507–1537 (2004)
123. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of IEE 121(12), 1585–1588 (1974)
124. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man Machine Studies 7, 1–13 (1975)
125. Manamanni, N., Mansouri, B., Hamzaoui, A., Zaytoon, J.: Relaxed conditions in tracking control design for a TS fuzzy model. Journal of Intelligent and Fuzzy Systems 18(2), 185–210 (2007)
126. Mangasarian, O.L., Wild, W.: Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues. IEEE Ttransactions on Pattern Analysis and Machine Intelligence 27(12), 1–6 (2005)
127. Mansoori, E.G., Zolghadri, M.J., Katebi, S.D.: A weighting function for improving fuzzy classification systems performance. Fuzzy Sets and Systems 158, 583–591 (2007)
128. Margaliot, M., Langholz, G.: Hyperbolic approach to fuzzy modeling and control. In: Proc. 8th IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE 1999), Seoul, Korea, pp. 893–897 (1999)
129. Margaliot, M., Langholz, G.: Hyperbolic optimal control and fuzzy control. IEEE Transactions on Systems, Man and Cybernetics, Part A 29, 1–10 (1999)
130. Margaliot, M., Langholz, G.: New Approaches to Fuzzy Modeling and Control–Design and Analysis. World Scientific, Singapore (2000)

131. Margaliot, M., Langholz, G.: Some nonlinear optimal control problems with closed-form solutions. International Journal on Robust Nonlinear Control 11, 1365–1374 (2001)
132. Margaliot, M., Langholz, G.: A new approach to fuzzy modeling and control of discrete-time systems. IEEE Transactions on Fuzzy Systems 11(4), 486–494 (2003)
133. Marino, R., Peresada, S., Valigi, P.: Adaptive input-output linearizing control of induction motors. IEEE Transactions on Automatic Control 38(2), 208–221 (1993)
134. Marquez, H.J.: Nonlinear Control Systems. Analysis and Design. John Wiley and Sons, New Jersey (2003)
135. Mata, N., Alvarez, T., de Prada, C.: State Space Constrained Multivariable Predictive Control of a Milk of Lime Blending Tank. Computers and Chemical Engineering 22(suppl.), S835–S838 (1998)
136. MATLAB. User's Guide. The MathWorks, Inc., Natick, MA 01760 (1994-2006), http://www.mathworks.com
137. Michalski, R.S., Bratko, I., Kubat, M.: Machine Learning and Data Mining: Methods and Applications. Wiley, New York (1998)
138. Mitra, S., Hayashi, Y.: Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 11(3), 748–768 (2000)
139. Mohan, B.M., Sinha, A.: Analytical Structures for Fuzzy PID Controllers? IEEE Transactions on Fuzzy Systems 16(1), 52–60 (2008)
140. Mohanlal, P.P., Kaimal, M.R.: Exact Fuzzy Modeling and Optimal Control of the Inverted Pendulum on Cart. In: Proc. of the 41st IEEE Conf. on Decision and Control, Las Vegas, Nevada, USA, December 2002, pp. 3255–3260 (2002)
141. Mohamadi, H., Habibi, J., Abadeh, M.S., Saadi, H.: Data mining with a simulated annealing based fuzzy classification system. Pattern Recognition 41, 1841–1850 (2008)
142. Mondada, F., Franzi, E., Ienne, P.: Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. In: Yoshikawa, T., Miyazaki, F. (eds.) Proc. of the 3rd Int. Symp. Experimental Robotics, October 1993, pp. 501–513. Springer, Berlin (1993)
143. Musielak, J.: Introduction to Functional Analysis (in Polish: Wstep do analizy funkcjonalnej). PWN, Warsaw (1976)
144. Nakashima, T., Yokota, Y., Ishibuchi, H., Schaefer, G.: A cost-based fuzzy system for pattern classification with class importance. Artificial Life and Robotics 12, 43–46 (2008)
145. Nauck, D., Kruse, R.: Obtaining interpretable fuzzy classification rules from medical data. Artificial Intelligence in Medicine 16, 149–169 (1999)
146. Nozaki, K., Ishibuchi, H., Tanaka, H.: Adaptive fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems 4, 238–250 (1996)
147. Park, B.-J., Pedrycz, W., Oh, S.-K.: Fuzzy polynomial neurons as neurofuzzy processing units. Neural Computing and Applications 15(3-4), 310–327 (2006)
148. Osuna, E.: Support Vector Machines: Training and Applications. Ph.D. thesis, MIT (1998)
149. Pang, C.-T., Lur, Y.-Y.: On the Stability of Takagi-Sugeno Fuzzy Systems With Time-Varying Uncertainties. IEEE Transactions on Fuzzy Systems 16(1), 162–170 (2008)
150. Patel, A.V., Mohan, B.M.: Analytical structures and analysis of the simplest fuzzy PI controllers. Automatica 38, 981–993 (2002)

151. Pedrycz, W.: Fuzzy modelling: fundamentals, construction and evaluation. Fuzzy Sets and Systems 41(1), 1–15 (1991)
152. Pedrycz, W.: Fuzzy Control and Fuzzy Systems. Research Studies Press Ltd (1993)
153. Perkins, W.R., Cruz, J.B.: Engineering of Dynamic Systems. John Wiley, New York (1969)
154. Peterson, L.E., Coleman, M.A.: Machine learning-based receiver operating characteristic (ROC) curves for crisp and fuzzy classification of DNA microarrays in cancer research. International Journal of Approximate Reasoning 47, 17–36 (2008)
155. Piegat, A.: Fuzzy Modeling and Control. Series: Studies in Fuzziness and Soft Computing, vol. 69. Physica-Verlag, Heidelberg (2001)
156. Platt, J.C.: Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MST-TR-98-14, Microsoft Research (1998)
157. Poggio, T., Smale, S.: The Mathematics of Learning: Dealing with Data. Notices of the American Mathematical Society 50(5), 537–544 (2003)
158. Rafiquzzaman, M.: Fundamentals of digital logic and microcomputer design, 5th edn. John Wiley and Sons, Singapore (2005)
159. Rasheed, S., Stashuk, D., Kamel, M.: A software package for interactive motor unit potential classification using fuzzy k-NN classifier. Computer Methods and Programs in Biomedicine 89, 56–71 (2008)
160. Ren, G., Xiu, Z.-H.: Analytical Design of Takagi-Sugeno Fuzzy Control Systems. In: American Control Conf., Portland, OR, USA, June 8-10, 2005, pp. 1733–1738 (2005)
161. Riedel, K.S.: A Sherman Morrison Woodbury Identity for Rank Augmenting Matrices With Application to Centering. SIAM Journal on Mathematical Analysis 12(1), 80–95 (1991)
162. Rifkin, R.: Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning. MIT, Cambridge (2002)
163. Rifkin, R., Klautau, A.: In Defense of One-Vs-All Classification. Journal of Machine Learning Research 5, 101–141 (2004)
164. Roh, S.-B., Pedrycz, W., Oh, S.-K.: Genetic Optimization of Fuzzy Polynomial Neural Networks. IEEE Transactions on Industrial Electronics 54(4), 2219–2237 (2007)
165. Rutkowska, D.: Neuro-Fuzzy Architectures and Hybrid Learning. Series: Studies in Fuzziness and Soft Computing, vol. 85. Springer, Heidelberg (2002)
166. Rutkowski, L.: Flexible Neuro-Fuzzy Systems: Structures, Learning and Performance Evaluation. Kluwer Academic Publishers, Springer (2004)
167. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
168. di Sciascio, F., Carelli, R.: Fuzzy Modelling and Identification of Multilinear Dynamical Systems. Latin American Applied Research 27(3), 175–181 (1997)
169. Seising, R.: From vagueness in medical thought to the foundations of fuzzy reasoning in medical diagnosis. Artificial Intelligence in Medicine 38, 237–256 (2006)
170. Sekiguchi, K., Deng, M., Inoue, A.: Obstacle Avoidance and Two Wheeled Mobile Robot Control Using Potential Function. In: Proc. IEEE Int. Conf. Industrial Technology, December 15-17, 2006, pp. 2314–2319 (2006)

171. Setnes, M., Babuška, R.: Fuzzy relational classifier trained by fuzzy clustering. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 29(1), 619–625 (1999)
172. Sexton, R.S., McMurtrey, S., Cleavenger, D.: Knowledge discovery using a neural network simultaneous optimization algorithm on a real world classification problem. European Journal of Operational Research 168(3), 1009–1018 (2006)
173. Shen, Q., Chouchoulas, A.: A rough-fuzzy approach for generating classification rules. Pattern Recognition 35, 2425–2438 (2002)
174. Shilnikov, A., Nicolis, G., Nicolis, C.: Bifurkation and predictability analysis of a low-order atmospheric circulation model. International Journal of Bifurcation and Chaos 5, 1701–1711 (1995)
175. Slotine, J.J.E., Li, W.: Applied Nonlinear Control. Prentice-Hall, Upper Saddle River (1991)
176. Spooner, J.T., Maggiore, M., Ordóñez, R., Passino, K.M.: Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques. John Wiley and Sons, Inc., New York (2002)
177. Sugeno, M.: An introductory survey of fuzzy control. Information Sciences 36(1/2), 59–83 (1985)
178. Sugeno, M.: On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents. IEEE Transactions on Fuzzy Systems 7(2), 201–224 (1999)
179. Tadeusiewicz, R., Ogiela, M.R.: Medical Image Understanding Technology: Artificial Intelligence and Soft-Computing for Image Understanding. Series: Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2004)
180. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1), 116–132 (1985)
181. Tanaka, K., Ikeda, T., Wang, H.O.: Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, h^{∞} control theory, and linear matrix inequalities. IEEE Transactions on Fuzzy Systems 4(1), 1–13 (1996)
182. Tanaka, K., Sano, M.: A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer. IEEE Transactions on Fuzzy Systems 2(2), 119–134 (1994)
183. Tanaka, K., Sugeno, M.: Stability analysis and design of fuzzy control systems. Fuzzy Sets and Systems 45, 135–156 (1992)
184. Tanaka, K., Wang, H.O.: Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach. Wiley, New York (2001)
185. Tomaszewski, S.P., Celik, I.U., Antoniou, G.E.: WWW-Based Boolean Function Minimization. International Journal of Applied Mathematics and Computer Science 13(4), 577–583 (2003)
186. Tsang, I.W., Kwok, J.T., Cheung, P.-M.: Core vector machines: Fast SVM training on very large data sets. Journal on Machine Learning Research 6, 363–392 (2005)
187. Unbehauen, H.: Some new trends in identification and modeling of nonlinear dynamical systems. Applied Mathematics and Computation 78, 279–297 (1996)
188. Vahidi, A., Stefanopoulou, A., Peng, H.: Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. Vehicle System Dynamics 43(1), 31–55 (2005)

189. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Chichester (1998)
190. Vega, P., Cameron, R.G., de Prada, C.: A model of a milk of lime mixing tank. Research Report R381, School of Control Engineering, Univ. of Bradford, UK (1987)
191. Wang, L.-X.: Adaptive Fuzzy Systems and Control. Prentice-Hall Inc., USA (1994)
192. Wang, L.-X.: A Course in Fuzzy Systems and Control. Prentice-Hall, Upper Saddle River (1997)
193. Wang, W.-J., Chen, Y.-J., Sun, C.-H.: Relaxed Stabilization Criteria for Discrete-Time T-S Fuzzy Control Systems Based on a Switching Fuzzy Model and Piecewise Lyapunov Function. IEEE Transactions on Systems, Man, and Cybernetics, Part B 37(3), 551–559 (2007)
194. Wang, L.-X., Mendel, J.M.: Generating Fuzzy Rules by Leaning from Examples. IEEE Transactions on Systems, Man, and Cybernetics 22(6), 1414–1427 (1992)
195. Wang, N., Meng, X.: Analytical Structure of Three-Dimensional Fuzzy Controller. In: Proc. of the IEEE Int. Conf. on Control and Automation, Guangzhou, China, May 30 - June 1, 2007, pp. 2349–2354 (2007)
196. Wang, H.O., Tanaka, K., Griffin, M.F.: An approach to fuzzy control of nonlinear systems: stability and design issues. IEEE Transactions on Fuzzy Systems 4(1), 14–23 (1996)
197. Wiktorowicz, K., Zajdel, R.: A Fuzzy Navigation of a Mobile Robot. Systems Science 23(4), 87–100 (1997)
198. Wodarz, D.: Viruses as antitumor weapons: defining conditions for tumor remission. Cancer Research 61, 3501–3507 (2001)
199. Wong, C.-C., Wang, H.-Y., Li, S.-A., Cheng, C.-T.: Fuzzy Controller Designed by GA for Two-Wheeled Mobile Robots. International Journal of Fuzzy Systems 9(1), 22–30 (2007)
200. Woodson, H.H., Merchel, J.R.: Electrodynamical Mechanics, Part I: Discrete Systems. John Wiley, New York (1968)
201. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 Algorithms in Data Mining. Knowledge and Information Systems 14(1), 1–37 (2008)
202. Wygralak, M.: Cardinality of Fuzzy Sets. Springer, Heidelberg (2003)
203. Xie, W.F., Rad, A.B.: Fuzzy on-line identification of SISO nonlinear systems. Fuzzy Sets and Systems 107(3), 323–334 (1999)
204. Yen, G.G., Meesad, P.: Constructing a fuzzy-rule-based system using the ILFN network and genetic algorithm. International Journal of Neural Systems 11, 427–443 (2001)
205. Ying, H.: The Takagi-Sugeno Fuzzy Controllers Using the Simplified Linear Control Rules are Nonlinear Variable Gain Controllers. Automatica 34(2), 157–167 (1998)
206. Ying, H.: Theory and application of a novel fuzzy PID controller using a simplified Takagi-Sugeno rule-scheme. Information Sciences 123(3), 281–293 (2000)
207. Ying, H.: Fuzzy control and modeling. Analytical foundations and applications. IEEE Press, New York (2000)

208. Ying, H.: Structure and Stability Analysis of General Mamdani Fuzzy Dynamic Models. International Journal of Intelligent Systems 20, 103–125 (2005)
209. Ying, H.: Deriving Analytical Input-Output Relationship for Fuzzy Controllers Using Arbitrary Input Fuzzy Sets and Zadeh Fuzzy AND Operator. IEEE Transactions on Fuzzy Systems 14(5), 654–662 (2006)
210. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
211. Zadeh, L.A.: Fuzzy algorithm. Information and Control 12, 94–102 (1968)
212. Zadeh, L.A.: Toward a Theory of Fuzzy Systems. In: Kalman, R.E., Dellaris, N. (eds.) Aspects of Network and System Theory, pp. 469–490. Rinehart and Winston, New York (1971)
213. Zadeh, L.A.: A Rationale for Fuzzy Control. Dynamic Systems, Measurement and Control 94, 3–4 (1972)
214. Zadeh, L.A.: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. IEEE Transactions on Systems, Man and Cybernetics 3, 28–44 (1973)
215. Zadeh, L.A.: On the analysis of large scale systems. In: Gottinger, H. (ed.) Systems Approaches and Environment Problems, Vandenhoeck and Ruprecht, Göttingen, pp. 23–37 (1974)
216. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Parts 1, 2 and 3. Information Sciences 8, 199–249, 301–357 (1975); 9, 43–80 (1976)
217. Zarandi, M.H.F., Türksen, I.B., Sobhani, J., Ramezanianpour, A.A.: Fuzzy polynomial neural networks for approximation of the compressive strength of concrete. Applied Soft Computing 8, 488–498 (2008)
218. Zhang, G.P.: Neural networks for classification: a survey. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 30(4), 451–462 (2000)
219. Zhegalkin, I.I.: On the Technique of Calculation the Sentences in Symbolic Logic (in Russian). Matem. Sbornik 34, 9–28 (1927)
220. Zhou, T., Tang, Y.: Chen's attractor exists. International Journal of Bifurcation and Chaos 14(9), 3167–3177 (2004)

Index