World Scientific Series in Robotics and Intelligent Systems – Vol. 27

FUZZY DECISION MAKING IN MODELING AND CONTROL

João M. C. Sousa • Uzay Kaymak

World Scientific

FUZZY DECISION MAKING IN MODELING AND CONTROL

WORLD SCIENTIFIC SERIES IN ROBOTICS AND INTELLIGENT SYSTEMS

Editor-in-Charge: C J Harris (*University of Southampton*)
Advisor: T M Husband (*University of Salford*)

Published:

Vol. 10: Cellular Robotics and Micro Robotic Systems (*T Fukuda and T Ueyama*)

Vol. 11: Recent Trends in Mobile Robots (*Ed. Y F Zheng*)

Vol. 12: Intelligent Assembly Systems (*Eds. M Lee and J J Rowland*)

Vol. 13: Sensor Modelling, Design and Data Processing for Autonomous Navigation (*M D Adams*)

Vol. 14: Intelligent Supervisory Control: A Qualitative Bond Graph Reasoning Approach (*H Wang and D A Linkens*)

Vol. 15: Neural Adaptive Control Technology (*Eds. R Zbikowski and K J Hunt*)

Vol. 17: Applications of Neural Adaptive Control Technology (*Eds. J Kalkkuhl, K J Hunt, R Zbikowski and A Dzielinski*)

Vol. 18: Soft Computing in Systems and Control Technology (*Ed. S Tzafestas*)

Vol. 19: Adaptive Neural Network Control of Robotic Manipulators (*S S Ge, T H Lee and C J Harris*)

Vol. 20: Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel Genetic Algorithms (*M A C Gill and A Y Zomaya*)

Vol. 21: High-Level Feedback Control with Neural Networks (*Eds. F L Lewis and Y H Kim*)

Vol. 22: Odour Detection by Mobile Robots (*R. Andrew Russell*)

Vol. 23: Fuzzy Logic Control: Advances in Applications (*Eds. H B Verbruggen and R Babuska*)

Vol. 24: Interdisciplinary Approaches to Robot Learning (*Eds. J. Demiris and A. Birk*)

Vol. 25: Wavelets in Soft Computing (*M. Thuillard*)

Vol. 26: Active Sensors for Local Planning in Mobile Robotics (*P. P. Smith*)

World Scientific Series in Robotics and Intelligent Systems – Vol. 27

FUZZY DECISION MAKING IN MODELING AND CONTROL

João M. C. Sousa
Technical University of Lisbon, Portugal

Uzay Kaymak
Erasmus University Rotterdam, The Netherlands

World Scientific
New Jersey • London • Singapore • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

FUZZY DECISION MAKING IN MODELING AND CONTROL
World Scientific Series in Robotics and Intelligent Systems – Volume 27

ISBN 981-02-4877-6

This book is printed on acid-free paper.

Printed in Singapore by Mainland Press

To my wife Ana, and our little boy Henrique — JS

To my wife Gülseren, and our passion Tutku — UK

This page is intentionally left blank

Foreword

From its very beginning, fuzzy set theory has generated a multitude of research streams and formed many interdisciplinary scientific communities. The idea to fuzzify crisp boundaries has become an important ingredient in many fields in mathematics, engineering, and computer science. The roots of fuzzy decision making can be viewed in operations research, expert systems, and knowledge processing. Fuzzification here means to blur decision boundaries and to avoid 'black or white' decisions that lead to sub-optimal solutions in many real applications. The mathematical elegance in fuzzy decision making lies in the unified representation of goals and constraints.

Fuzzy modeling has been driven by pattern recognition, function approximation, and by process identification in control engineering. We distinguish the extraction of models from observed process data (identification) and the computation of model outputs from measured inputs (evaluation). Fuzzy clustering plays an important role in model identification. In pattern recognition, clustering is used to fuzzily assign objects to classes. This scheme is adapted in control engineering to assign process states to prototypical operating points or local process characteristics. Fuzzy models identified by clustering can be evaluated using fuzzy rule based models (Mamdani–Assilian, Takagi–Sugeno), so fuzzy modeling can also be viewed as the extraction of fuzzy rules from data. This is also an important application field for neural networks and evolutionary algorithms.

More and more, fuzzy control techniques use the power of modern control methods, solving the problems of multiple-model approaches in the sense of a fuzzy combination of local models and controllers. In this connection, with respect to the total system, the notions of stability of the whole system, its robustness, and performance play an eminent role. It is, furthermore, evident that fuzzy control problems can only satisfactorily be solved in the presence of a well-balanced combination of control issues on the one hand and modeling, optimization, and decision making on the other. This requires the interplay of control strategies like fuzzy model based control, gain scheduling, or fuzzy predictive control and corresponding modeling and optimization methods. Other control techniques such as Fuzzy Aggregated Membership Control go back to the roots of fuzzy decision making and open the way for the control engineer to a direct influence on the control rules and the nonlinear control behavior obtained from expert knowledge.

Fuzzy model based control and fuzzy gain scheduling deal with multiple-model Takagi–Sugeno fuzzy systems that approximate nonlinear control systems both in operating points and in off-equilibrium regions. Here the aspects of modeling and the choice of appropriate input signals come into play. An important point is the derivation of inverse affine Takagi–Sugeno fuzzy models for compensation purposes in model-based control and predictive control, respectively. Inverse models are especially useful for process control applications, e.g., in chemical plants or for fermentation processes. Optimization in fuzzy model predictive control can be done in many ways. Useful methods are the well-known gradient descent method, branch-and-bound methods, and genetic algorithms.

Fuzzy optimization can be viewed in a first instance as a pure math domain. We distinguish to find (crisp) extrema of functions defined by fuzzy models and to find fuzzy extrema using fuzzy goals and constraints. Process optimization with fuzzy models can be done using crisp optimization methods or using crisp or fuzzy controllers. The determination of fuzzy extrema, however, is a fuzzy decision making process again.

All these four fields: fuzzy decision making, fuzzy modeling, fuzzy control, and fuzzy optimization, play an important role in automation and control today. Originally stemming from different scientific areas, these fields are in a continuous process of merging together. Today's state-of-the-art high-level controllers perform decision making and optimization functions based on data or knowledge driven models. These tendencies are well reflected in this book, which covers many of the most important areas of fuzzy technologies in the field of modern automation and control. We expect the techniques studied here to be the basis of a wide range of modern control applications, and we hope that this book contributes to a unification of the promising research areas related to fuzzy decision making.

Thomas A. Runkler
Rainer Palm

Preface

Since Lotfi Zadeh's introductory paper in 1965, the fuzzy set theory and the applications of fuzzy systems have come a long way. The initial hesitation, even the hostile reaction to fuzzy set theory has made way for enthusiasm, or at least tolerance for fuzzy systems. The successes of the practical application of fuzzy set theory in fuzzy systems is an important factor in the change of attitude towards fuzzy set theory. Control engineering has contributed significantly to the number of successful applications of fuzzy systems, including the first industrial application of a fuzzy system. Today, fuzzy modeling and control have taken their place amongst the tools of control engineers for designing control systems. In contrast, fuzzy decision making methods have been applied to a comparatively smaller degree, although the literature on the basics of fuzzy decision making extends back to the beginning of the 1970s. Despite the fact that decision making and control are related fields, the combination of fuzzy control methods and fuzzy decision making methods has hardly been investigated. This book addresses the combination of the two fields. It is shown that looking at control problems from a fuzzy decision making perspective leads to new insights, which can be used in the design of improved control systems.

Various ways in which fuzzy decision making methods can be applied to systems modeling and control are considered in this book for the design of fuzzy controllers. The book consists of three parts. In the first part, which consists of Chapters 2–4, basics of fuzzy decision making as used in the remaining chapters are explained. This part also includes a description of how direct fuzzy controllers can be designed by directly applying the theory of fuzzy decision making to controller design. The second part of the book consists of Chapters 5–9. The main focus of the book, including various forms of fuzzy model-based control (FMBC) and the use of fuzzy decision making in FMBC, is found in this part. In the third part, which consists of Chapters 10–13, we discuss optimization issues in the control schemes presented, and we give an example of a real-world application.

Finally, some guidelines for future research in FMBC are also discussed.

The large spectrum of fields covered by this book, *i.e.*, fuzzy decision making, fuzzy modeling, fuzzy control, and fuzzy optimization, makes it attractive to a large number of possible readers. Previous knowledge of fuzzy set theory, modeling, control and decision making is not mandatory, but it would help to understand the contents presented. As such, the book is intended for researchers, graduate students and advanced undergraduate students. We expect that lecturers, who teach courses related to any of the four main areas covered by the book at an advanced undergraduate or higher level, would also find the contents interesting.

João M. C. Sousa
Uzay Kaymak

Acknowledgments

Part of the material in this book has appeared in various journal and conference publications. These publications are referred to in the relevant paragraphs, and they are listed in the bibliography. The material, including figures and tables, is reproduced with the kind permission of the respective copyright holders. The list of the copyright holders is as follows. IEEE: (Kaymak and van Nauta Lemke 1993), (Kaymak and Babuška 1995), (Sousa, Kaymak, Verhaegen and Verbruggen 1996), (Kaymak et al. 1996), (Kaymak, van Nauta Lemke and den Boer 1998), (Setnes, Babuška, Kaymak and van Nauta Lemke 1998), (Sousa and Setnes 1999) and (Sousa and Kaymak 2001). Elsevier Science: (Sousa et al. 1997), (Onnen et al. 1997), (Setnes, van Nauta Lemke and Kaymak 1998) and (Kaymak and van Nauta Lemke 1998). Kluwer Academic Publishers: (Kaymak et al. 1997) and (Sousa et al. 1999). John Wiley & Sons: (Sousa 2000). Physica-Verlag: (Babuška et al. 1998). IOS Press: (Kaymak, Babuška, van Nauta Lemke and Honderd 1998). Furthermore, we would like to thank our colleagues and students, who have co-authored some of these publications with us.

This page is intentionally left blank

Contents

Chapter 1

Introduction

Control engineering deals with the design, realization and the theory of control systems. Although building 'controlled' systems that behave within design parameters or according to expectations has always been an issue, the real advances in the control theory and control science have appeared in the twentieth century. This chapter describes the recent developments in control engineering. The motivation for using fuzzy control systems is given. It is shown that, due to the new developments in modern process operation and production methods, the control problems are attaining more and more the characteristics of decision problems. In this respect, the motivation for the main theme of this book, *i.e.*, a combination of techniques from fuzzy decision making and fuzzy control, is explained.

Section 1.1 discusses two basic approaches to the design of control systems. The two main design methods for controllers and the importance of measurement in control systems is also discussed. Section 1.2 considers the developments in the advanced control systems and establishes the link to intelligent control systems. The link to fuzzy control and the relation to fuzzy decision making are discussed in Sec. 1.3. Finally, a detailed chapter outline for the book is given in Sec. 1.4.

1.1 Control systems

The design of equipment and systems for accomplishing various tasks has been an important human activity for centuries. One of the most challenging tasks in designing these systems has always been the control of the important quantities for the designed system. Until the twentieth century, the control of desired quantities, such as the speed of steam engines, was the domain of skilled craftsmen, who used experiments, their common sense and their experience to design successful systems. Many of these systems have been vital for the development and sustenance of large communities, like the waterwheels of Hamāh in Syria, built in the fourteenth century. Several of these impressive waterwheels, with carefully designed wheels ranging from 10m to over 20m, used to haul water to different heights for irrigation and drinking, are still in present-day use.

Not all the systems designed were as useful as the waterwheels of Hamāh, however. Some were mere curiosities in their time, such as several mechanical toys that satisfied the fascination of humans for automata. Already in the first century AD, Heron of Alexandria invented a self-moving stand with puppets that could be made to replay whole scenes from a play. The movements of the puppets were caused by special boxes filled with sand, where the sand escaped from a reservoir through a sand hole, inducing motion due to the change in the mass of the boxes. By modifying the size of the sand hole and the amount of sand, the puppets' speed and direction of motion could be controlled. This system essentially uses a *feedforward* control strategy, where the controller parameters are changed up front, depending on the system characteristics and the desired response.

The second and certainly more important control strategy is that of *feedback* control. The notion of feedback control was also familiar to Heron. It is reported that he had built an ingenious system powered by the expansion of heated air that opened the doors of a temple during religious ceremonies by burning a fire on the holy altar. The doors of the temple opening without any human 'intervention' must have been a major attraction at the time, and the priests running the ceremony made sure that the doors opened only when the believers brought in sufficient presents (offerings) by opening or closing a small secret outlet for the expanding air.

Despite the ingenuity of these systems and their designers, a mathematical description and analysis of control systems would not come until the beginning of the twentieth century. The significant change in the design of control systems was brought about by the systematic inclusion of measurement systems for control purposes. Feedback controllers could now be implemented, which processed the information from the measurement equipment and delivered information to the actuators acting on the manipulated variables to influence the system. The possibilities for the control engineering increased even further by the introduction of computer systems. Sophisticated control algorithms could now be implemented, while major developments in the theory of control systems improved the analysis.

The general feedback control scheme for multivariable systems is depicted in Figure 1.1. Various input signals (actions) influence the process $\mathbf{P}$ resulting in output variables $\mathbf{y}$. The input variables are usually divided in control actions or manipulated variables $\mathbf{u}$ and system disturbances $\mathbf{d}$, which can not be influenced before entering the process. The *goals* to be achieved are imposed on the controller (indicated by the double arrow in Fig. 1.1), such that the system under control achieves the desired specifications. Note that the reference $\mathbf{r}$ can also be seen as a goal to be achieved by the control system. The plant under control and the actuators manipulated by $\mathbf{u}$ are included in the process. The sensors are represented by the operator $\mathbf{S}$, having as inputs, the output variables from the process $\mathbf{y}$ and the measurement disturbances $\mathbf{d}_m$, generating the measured outputs $\mathbf{y}_m$. The controller $\mathbf{C}$ generates the control actions $\mathbf{u}$ based on the received information: the measured outputs $\mathbf{y}_m$, the references $\mathbf{r}$ to be followed, the disturbances $\mathbf{d}$ and $\mathbf{d}_m$ if available, and the goals to be obtained.

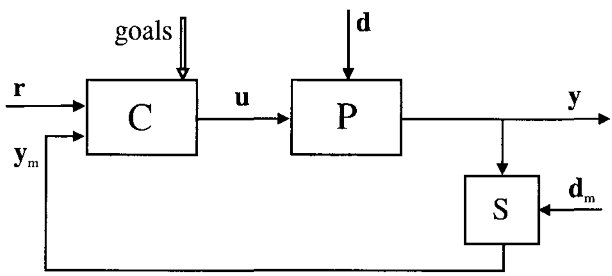


Fig. 1.1 Block diagram of a feedback control scheme.

Two main methods can be distinguished for designing controllers in the general control scheme of Fig. 1.1.

(i) **Signal-based control.** Measurements of various signals of the system are used for computing the control action directly, without evaluating the process knowledge on-line. In other words, the knowledge of the system (*e.g.* a model) is only used during the design stage to determine the proper setting of the controller parameters.

(ii) **Model-based control.** The parameters of the control are optimized based on a given desired response (a reference), a model of the process and performance criteria. Contrary to signal-based control, the model of the system and the controller parameters are directly related in model-based control. In digital control, the model is used at each sampling instant to predict the output of the system.

In this book, both design approaches are considered. As indicated in Sec. 1.3, the relations between fuzzy control and fuzzy decision making are considered in detail, and a synergistic combination of the two fields is studied.

1.2 Advanced control systems

Signal-based control is still by far the most common design approach. This approach has often led to the widespread implementation of simple controllers and the application of simple tuning rules to set the parameters of the designed controllers. In early control systems, the outputs of the controller were derived directly from the difference between the desired and the measured signals obtained from the controlled system. The PID controllers which are extensively used in the industry make use of such a difference signal. Although PID controllers form a substantial part of the control systems used today, their performance is usually not sufficient when high performance requirements are imposed on the controlled system, or when the controlled process shows large variations in the region for which the PID controller is tuned. This is increasingly the case with modern production systems. Verbruggen and Bruijn (1999) distinguish the following characteristics in modern process operation and production methods.

- **An increasing demand for flexibility.** The plants operate with varying throughput, product mix and product grade. The customer obtains a lot of freedom in product definition up to the point of almost personalized product specification. Consequently, the process is required to operate at different operating points, and to change from one operating point to another fast and frequently, while taking various constraints into account. Hence, the system exhibits strong nonlinear behavior, and often the control engineer will not have sufficient time for extensive analysis and modeling of the process, contrary to what is customary with production processes operating at a few well-defined operating points.
- **A strong drive for plant-wide control.** New production methods involve multiple control loops, while the total manufacturing process must be optimized for efficiency because of tighter economic constraints (*e.g.*increased competition in many markets). The control and optimization of many loops, many subsystems and their interaction must now be considered simultaneously, so as to optimize the total production process. Mathematical description of many systems with a large degree of interaction amongst themselves is tedious and not always possible. However, experience may be available in the form of expert rules or operator best practice. Such knowledge about the system should be combined with the mathematical descriptions of the system to enhance the performance of the controllers.
- **A growing need for integrated information systems.** The presence of multiple, interacting control loops requires a more sophisticated organization of the controllers than simple feedback control. Typically, controllers are organized

hierarchically, leading to various levels of automation: control, monitoring, optimization, supervision, planning, task management. Such an organization requires the ability to deal with qualitative and quantitative information in a single system with different levels of precision and complexity. Human-machine interaction is also very important, and sophisticated human interfaces are required.

Clearly, these developments mandate the use of sophisticated control systems. Contrary to signal-based control, advanced control techniques use an online model of the controlled process in order to determine the control signal. Initially, linear process models were used, and linear control systems based on these models have been an important area of investigation in control theory. Most real processes are in fact nonlinear, but as long as the system is considered around a constant operating point, it can be linearized and approximated by a linear model.

Because of the approximations and the necessary simplifications, no process model is accurate to an arbitrary degree. Even assuming that a correct class of models can be identified for a process, the parameters of the models are usually identified from the results of experiments on the process itself. These experiments are sometimes time consuming and expensive, and so the model parameters can only be estimated approximately. Hence, there is always a model-process mismatch and it is important to evaluate the sensitivity of the control system to the variations in the model parameters. Recent developments in linear control theory have then been towards the formalization of the design of robust control systems that are less sensitive to the changes in the process parameters and the mismatch between the model and the process, see, *e.g.*, (Palm et al. 1997, Dullerud and Paganini 2000, Ioannou and Sun 1996, Tang et al. 2000). Robust control theory (such as H_∞ control) is motivated by such goals. Another approach that is sometimes combined with robust control and that can deal with deviations from the model parameters is adaptive control, where the controller parameters are modified depending on the changing conditions, see, *e.g.*, (Calise et al. 2001, Su et al. 1999, Veres and Sokolov 1998). These autonomously adapting systems are often protected by extensive safety nets and other safety systems.

Due to the developments in production methods discussed above, the process nonlinearity has become an important issue, which has increased the interest in nonlinear control and modeling. Multiple approaches have been developed in the past decade for nonlinear control and modeling. In addition to the nonlinearity, the importance of the interaction between various control loops has also increased. In such a setting, the communication and the coordination between several controllers, each with its own set of control goals and constraints, becomes dominant. To deal with increasing complexity in such a system, the control systems are organized in hierarchies, which lead to supervisory systems and to task-oriented systems. Higher levels in a hierarchy perform more complicated tasks, while the lower levels are involved in less complicated tasks. Saridis (1977) proposes a three layer structure for classifying hierarchical control systems, as shown in Fig. 1.2. Since the controllers at higher levels of the hierarchy must deal with more varying situations and multiple goals and constraints, they need to be more flexible, making a trade-off amongst the goals of the various controllers and their performance. This corresponds to an increased 'intelligence' in their functioning. Intelligent control recognizes this fact and considers control systems that can operate satisfactorily despite the increasing complexity of a system and its functions. Saridis (1977) observes that the increase in intelligence leads to a decrease in the precision of available and processed information. The goals at higher levels are not known exactly, and they can often be defined in approximate terms only. Moreover, many conflicting goals and constraints must be dealt with to realize the overall objectives of the control system, while the goals may change, depending on the operating conditions. Also, some constraints may be relaxed when important goals cannot be attained otherwise.

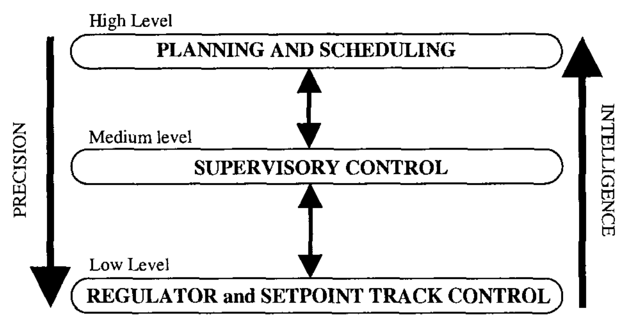


Fig. 1.2 Three levels of organization for control systems, according to Saridis.

Intelligent control methods use techniques inspired by ideas about the functioning of biological systems to design controllers. The methods employed include neural networks, fuzzy sets and evolutionary computation. To deal with increasing complexity in the presence of multiple goals and constraints that may vary in time, as much information as possible about the control system and its relevant variables must be used to design the controllers. However, obtaining precise information can be costly, due to the time it may take to collect the data and due to possible losses in the quality and the quantity of the products. Moreover, precise information is not always relevant, as expressed by the *principle of incompatibility* of Zadeh (1973). According to the principle of incompatibility, as the complexity of a system increases, the ability to make precise and yet relevant statements about the system diminishes until the precision and relevance become mutually exclusive beyond a certain point. To control complex systems, the controllers must be able to deal with approximate information in order to use relevant information. Fuzzy controllers use concepts from fuzzy sets theory to deal with nonlinearity, uncertainty and vagueness at various levels of organization. Although most of the material in this book concerns low and medium levels, fuzzy set theory also provides a mathematical framework to address explicitly the issues regarding supervision, flexible constraints, subjective goals and multiobjective planning that one encounters in the supervisory and planning levels.

1.3 Fuzzy control and decision making

Fuzzy control systems deal with approximate information by using the fuzzy set theory for dealing with imprecise, fuzzy and vague information. Fuzzy controllers apply fuzzy sets and operations on fuzzy sets to model process nonlinearity, to establish a link between linguistic information and mathematics of the controller, to capture heuristic knowledge and rules of thumb, and to model the approximate behavior of systems. Especially in human decision making or in cases where humans are involved in controlling a process, a lot of expertise is available in the form of heuristic rules of thumb, global descriptions of the behavior of the system and the effects of various control alternatives in terms of the desired goals and the imposed constraints. Much of this knowledge is described in linguistic terms, which can be modeled by the use of fuzzy sets. In order to recognize specific situations (such as the occurrence of an error or entering a particular operating region), reasoning based on heuristic information may be required. By the use of heuristic and vague information, the controller attains some ability to use knowledge in a context-dependent way, and to adjust the control actions or control strategy according to that context.

1.3.1 *Fuzzy logic control*

The objective of fuzzy logic control (FLC*), or conventional fuzzy control, is to control complex processes using a knowledge-based control strategy derived from human experience. These controllers are based on the work presented by Zadeh (1973). The first application of FLC on a system was made by Mamdani and Assilian (1975). This type of control is used when a reasonable model of the physical

*We use FLC as an abbreviation for both 'fuzzy logic control' and 'fuzzy logic controller'. It will be clear from the context in what sense we use the abbreviation.

system is not available, not possible to obtain, or if it is unreasonably complex to be used for control purposes (Sugeno 1985). Moreover, the determination of appropriate models is time-consuming, requires a solid theoretical background, and models are always a simplification of the process. However, humans are able to control complex processes (*e.g.*driving a car in busy traffic), which cannot be easily controlled by conventional control systems without a model. Thus, the control design in FLC is based on empirical knowledge regarding the behavior of the process, and does not use a strictly analytic framework. This knowledge, cast into a linguistic or rule-based form, constitutes the basis of a fuzzy logic control system.

FLC design follows the signal-based control approach. It is based on implementing expert knowledge in the form of If–Then control rules, linking the input variables of the controller with the control variables by using linguistic terms. Consider, for example, the control of the temperature in a room. A required temperature can be obtained by using rules of the type: 'If the temperature is too high, decrease the heating power considerably'. The definition of all control rules, acquired from an expert, constitutes the first step in the design of an FLC.

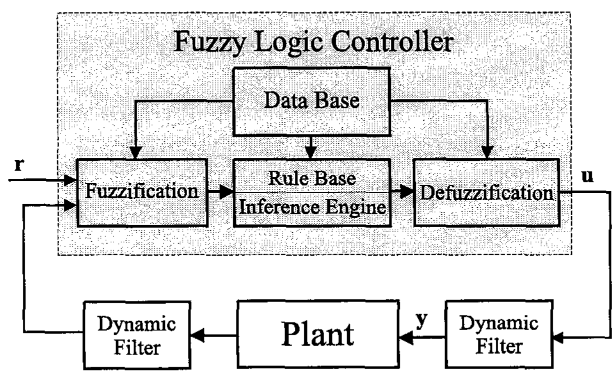


Fig. 1.3 Fuzzy logic controller scheme.

The set of all control rules constitutes the *rule base*. Furthermore, data are included in the *data base* that provides the necessary information for the proper functioning of the *fuzzification* module, the rule base and the *defuzzification* module (see Fig. 1.3).

Fuzzification converts the values of the measured process variables into linguistic values that are represented by fuzzy sets, thereby making them compatible with the fuzzy set representation in the rule base. Sometimes, fuzzification is preceded by a normalization step, which maps the measured values into a suitable range constituting the normalized universes of discourse used in the rule base.

The module *rule base* in Fig. 1.3 contains a relation described in fuzzy terms. The inference engine computes the appropriate control action according to the fuzzified inputs and the rule base. Often, the compositional rule of inference is used (Zadeh 1973).

The defuzzification translates the fuzzy outputs provided by the inference engine into a numerical (crisp) representation. Sometimes this value must be denormalized, *i.e.*, the values of the control output must be mapped onto their physical domains. The most common defuzzification methods are the *center-of-gravity* and the *mean-of-maxima* methods (Driankov et al. 1993). The result of defuzzification can be the control actions $\mathbf{u}$. In that case, they can be applied directly to the controlled process. However, if the fuzzy controller computes the change $\Delta\mathbf{u}$ in the control actions, the controller output must be integrated and a dynamic filter between the defuzzification block and the process is included in the control scheme, as presented in Fig. 1.3.

The fuzzy rules contained in the conventional fuzzy controller do not include any dynamics. The dynamic behavior is provided by an external *dynamic filter*, that computes the variables needed as inputs in the FLC. Examples of these variables are the errors between the references $\mathbf{r}$ and the outputs $\mathbf{y}$, the rate of change or the cumulative sum of these errors, or other dynamic time shift operations such as regressions on the inputs and outputs. The fuzzy control scheme in Fig. 1.3 does not show the disturbances explicitly, but they are usually present in practice. The FLC must be designed such that it can cope with these disturbances, but unfortunately this problem is not always considered. Comparing the design of a conventional fuzzy controller to a classical controller, the steps concerning the modeling and the choice of design specifications are not explicitly present in the FLC. The If–Then rules implicitly contain the performance criteria and the choice and settings of the controller meeting the desired specifications.

Fuzzy logic controllers are usually tuned by a trial-and-error method using simulations or experiments on the system. Unfortunately, experience shows that this design methodology has some significant drawbacks. Expertise to be extracted from operators can not always be expressed in a rule-base form, and it is a time-consuming task. Moreover, in an industrial environment, the in-line trial-and-error controller tuning is often not acceptable for, *e.g.*, safety, economical and environmental reasons. Furthermore, the performance of the FLC mimics the control actions performed by the operator, and therefore does not perform better than the best operator. However, this control is consistent, and independent from the 'mood' of the operator. Therefore, the variance in the output could be reduced compared to manual control.

1.3.2 *Fuzzy model-based control*

Model-based control can also be used for the design of fuzzy controllers. This approach is used in the larger part of this book, where the design of fuzzy controllers closely follows the classical control design of a model-based control system, starting with the modeling of the process to control, followed by the choice of the design specifications and their combination in performance criteria, and finally, designing the controller to be used in the system. This approach is called *fuzzy model-based control*. Depending on how and where the fuzzy set theory is applied, the term fuzzy model-based control (FMBC) can have different meanings. In the following, we define the FMBC problem as a nonlinear control problem for which the main goal is as follows.

> *Given the model of a system under control and the specifications of its desired behavior, design a feedback control law, such that the closed loop system behaves in the desired way, where the model and/or the design specifications and/or the developed controller use elements from the fuzzy set theory.*

This definition is rather broad, and several combinations of the different design components can be made. The explicit utilization of the fuzzy sets theory can be included in three distinct parts of controller synthesis,

(i) by using fuzzy models,
(ii) by defining fuzzy performance criteria as a confluence of fuzzy design specifications, and
(iii) by designing fuzzy control elements.

Therefore, considering that the model, the specifications or the controller can be fuzzy, there are eight possible combinations of conventional and fuzzy design components in FMBC.

(1) The model, the design specifications and the controller are all conventional. In this case, we have conventional model-based control. These types of controllers are outside the scope of this book. Several basic schemes from conventional model-based control, which are used in this book, are sketched in Appendix A and Appendix B.
(2) Only the model is fuzzy. This is the most common application of FMBC. A fuzzy model of the system to be controlled is used in an otherwise conventional model-based control scheme. The fuzzy model can be used to capture experts' knowledge about the system or to deal with nonlinearity in the system.

(3) Only the controller is fuzzy. This type of FMBC is not very common, because when a conventional model of the process is available and the design specifications are classical, there are already good conventional model-based design methods that can be used.
(4) Only the performance criteria are fuzzy. This type of FMBC can be very useful in cases where there are already good process models and control schemes available, but the performance of the closed loop system is best described in linguistic or in approximate terms.
(5) The model and the controller are fuzzy. This is a large class of FMBC that includes, amongst others, inverse fuzzy controllers operating in an internal model control scheme for robustness. The fuzzy controller is somehow derived from the fuzzy model, but the performance of the controller is measured by using conventional crisp performance criteria.
(6) The model and the performance criteria are fuzzy. This approach combines nonlinear control with specification of flexible design criteria. In that way, it becomes possible to obtain nonlinear model-based controllers that can deal with fuzzy and imprecise information.
(7) The performance criteria and the controller are fuzzy. This approach combines the design of fuzzy controllers with the flexible design specifications or constraints.
(8) The model, the performance criteria and the controller are fuzzy. This is essentially a union of the pairwise combinations discussed above. It can be considered as the most general form of FMBC, where all design components incorporate concepts from the fuzzy sets theory.

In this book, we consider mainly fuzzy models in order to deal with process nonlinearity and to capture expert knowledge about the process. We combine fuzzy models with fuzzy controllers and/or with fuzzy performance criteria in most cases.

Note that the FMBC approach is not based on trial-and-error, which makes the development of a fuzzy controller more systematic and more goal oriented. The most common design of FMBC systems considers only fuzzy models, leaving the design specifications and the control element non-fuzzy. Several authors have presented controllers based on fuzzy models. Johansen (1994), for instance, describes a nonlinear controller based on a fuzzy model of MIMO dynamic systems. This controller is a discrete-time nonlinear decoupler, also known as feedback linearizing controller. Zhao et al. (1997) present a controller based on various stabilizing state-feedback controllers, which use linear matrix-inequalities methods. One of the approaches derives a fuzzy controller, where a gain matrix is obtained from fuzzy implications. Fuzzy model-based predictive controllers and controllers based on the inversion of fuzzy models are presented in Babuška (1998). Palm et al. (1997) present a survey on several model-based design methods of fuzzy controllers, where the design of sliding mode FLC and Takagi–Sugeno FLC are presented. In these approaches the controller is always fuzzy, the model might or might not be fuzzy, and the design specifications are crisp. The use of *fuzzy goals* and *fuzzy constraints* in fuzzy decision making was first introduced by Bellman and Zadeh (1970). More recently, Kacprzyk (1997) has applied fuzzy goals and fuzzy constraints as fuzzy design specifications or as fuzzy criteria in control problems.

As FMBC closely follows the classical model-based control design approach, the performance criteria must be explicitly defined, because they are not implicitly included in the rules, as in FLC. Therefore, human knowledge can be used at a higher level for defining the control goals. Because of the fuzzy approach, the goals can be quite general (fuzzy) at the beginning, *e.g.*human comfort in an air-conditioned room, being decomposed afterwards in several hierarchical levels. For the given example, comfort has to be translated into different sub-goals, related, *e.g.*, to a desired temperature interval and a desired humidity range for a certain season of the year. Several aspects regarding fuzzy model-based control are considered in this book. The selection of the different elements in the control design problem results in a wide range of possible combinations based on the type of (nonlinear) model, conventional or fuzzy performance criteria, and different control structures such as model-based predictive control or internal model control.

1.3.3 *Fuzzy decisions for control*

The control problems at the supervisory and task-oriented level are characterized by subjective goals for the control system, flexible constraints and a mix of continuous and discrete control actions. Due to these aspects, such control problems show many similarities to the decision making problems. Especially in fuzzy control, where fuzzy sets are used to represent vague, fuzzy and imprecise information, fuzzy decision making methods provide mechanisms for designing fuzzy control systems (model-based or signal-based), in which the subjective information is incorporated through decision making schemes. Such a combination of fuzzy decision making and fuzzy control improves the flexibility of the control systems, and it extends the applicability of the fuzzy control systems to a larger class of control problems. This book considers various ways in which fuzzy control methods can be combined with fuzzy decision making methods, so as to obtain more efficient and flexible controllers.

Fuzzy decision making methods can be used to enhance various steps of reasoning and control in the existing methods by solving sub-problems present in those methods. Furthermore, fuzzy decision making is a good paradigm for dealing with human expert knowledge when designing fuzzy model-based control systems. The combination of fuzzy decision making and fuzzy control is shown in this book to lead to novel control schemes that improve the existing controllers in various ways. The following application of fuzzy decision making methods are considered for designing control systems.

- Fuzzy decision making for designing signal-based fuzzy controllers. The controller mappings and the defuzzification steps can be obtained by decision making methods.
- Fuzzy decision making for enhancing fuzzy modeling. The selection of the values of important parameters in fuzzy modeling algorithms by the use of fuzzy decision making.
- Fuzzy design and performance specifications in model-based control, where fuzzy constraints and goals are used.
- Design of model-based controllers combined with fuzzy decision modules. Incorporation of human operator experience in the performance specification.

The following section provides more details about the topics considered in the rest of the book and their division across the chapters.

1.4 Chapter outline

The material in this book is organized into 13 chapters and a couple of appendices. This section gives a detailed outline of the chapters and the main topics that each chapter considers.

In Chapter 2 we consider the basic formulation of fuzzy decision making. It contains a classification of the decision making problems and a description of the basic elements of the fuzzy discrete choice problem considered in the rest of the book. In this chapter, we also establish the terminology on fuzzy decision making according to the model of Bellman and Zadeh, which is used in the rest of the book.

A key notion in fuzzy decision making is fuzzy aggregation and the decision functions used for the aggregation. Various decision functions used in fuzzy decision making are reviewed in Chapter 3. A large number of decision functions is considered. Weighted aggregation, where the decision criteria are not equally important, is also studied, and various decision functions for weighted aggregation are presented.

In Chapter 4, we consider conventional fuzzy controllers based on Mamdani-type systems. Then, the design of signal-based fuzzy controllers using a fuzzy decision making approach is explained. These so-called *Fuzzy Aggregated Membership* (FAME) controllers implement nonlinear control laws. The relations to the linear PID-controllers and the nonlinear fuzzy PID controllers are also established. Furthermore, the function approximation capabilities of the FAME controllers are studied.

A significant part of the book is concerned with fuzzy model-based control and the application of fuzzy decision making in a fuzzy model-based control setting. One of the most important stages in model-based control is the identification of the process to be controlled. This is often also the most time consuming step in the design of the control system. Fuzzy model-based control can use various types of models for describing the process. We consider in this book mainly the use of fuzzy models. In Chapter 5, we describe how fuzzy models can be obtained. Methods based on the formalization of expert knowledge as well as on identification from process measurements are considered. Different types of fuzzy models such as Mamdani models and Takagi–Sugeno models are also explained in this chapter. Identification of Takagi–Sugeno fuzzy models using product-space fuzzy clustering is explained.

The use of fuzzy decision making methods for obtaining fuzzy models of systems is considered in Chapter 6. A powerful method for obtaining fuzzy models from system measurements is fuzzy clustering. This chapter describes how fuzzy decision making is applied in compatible cluster merging for determining the number of clusters in the clustering algorithm. The second part of the chapter considers the decision points in a Mamdani fuzzy model and describes how the defuzzification step can be formulated as a decision making problem. Based on this formulation, a new defuzzification method is described, which is applied in fuzzy security assessment of power systems.

Chapter 7 starts the discussion of fuzzy model-based control by considering model-based control with fuzzy models. A controller based on the inversion of a fuzzy model is described. The inversion of two different types of fuzzy models is considered,

(i) singleton fuzzy models, and
(ii) Takagi–Sugeno fuzzy models that are affine with respect to the control action.

A scheme for on-line adaptation of singleton fuzzy models is also discussed. Furthermore, control based on fuzzy compensation is explained, as well as a predictive controller based on a fuzzy model. The chapter concludes with an example of pressure control, using the fuzzy model-based control schemes presented.

All controllers must satisfy a set of specifications defined by the user. These specifications determine the final tuning of the controller parameters. In the model-based predictive control scheme, the design specifications are used on-line for computing the controller output. In Chapter 8, we present various design specifications and their translation to design criteria. Design specifications for linear and nonlinear systems are discussed. Classical performance specifications including input–output specifications, regulation specifications and actuator effort are presented. Classical performance criteria are described by using norms and seminorms of signals and systems. A generalization to fuzzy performance criteria is made when the criteria are defined by using fuzzy sets.

Model-based predictive control uses the performance criteria on-line to determine a sequence of optimal control signals. When fuzzy performance criteria are used, fuzzy objective functions aggregate the performance information. Model-based predictive control by using fuzzy objective functions is presented in Chapter 9. It combines fuzzy decision making theory with model-based predictive control (MBPC). The application of fuzzy decision making in the predictive control setting has two main design problems,

(i) the choice of the fuzzy criteria, and
(ii) the aggregation operators to combine them.

Both problems are addressed in this chapter. Moreover, fuzzy decision functions are used to mimic an experienced operator's control strategy. This approach illustrates how fuzzy decision making provides a mechanism to combine expert knowledge with a formal mathematical description. The presented approaches are illustrated with several examples, including temperature control and a simulated container crane system.

In Chapter 10, we discuss how on-line optimization can be used in fuzzy model-based control. First, MBPC using fuzzy models and classical objective functions is addressed. For this situation, the optimization problem is in general non-convex. Two methods are presented to cope with this optimization problem,

(i) branch-and-bound, and
(ii) genetic algorithms.

Secondly, optimization in MBPC using fuzzy objective functions is discussed. A branch-and-bound algorithm is introduced to deal with optimization using fuzzy objective functions.

In Chapter 11, we discuss a couple of advanced issues for on-line optimization in FMBC. First, special conditions under which the optimization by using fuzzy objective functions remains convex are presented. Under these conditions, efficient optimization algorithms can be used for fast optimization. Second, the problem of optimization accuracy and steady-state stability of closed loop is considered when discrete search methods from Chapter 10 are used for the optimization. A solution based on scaling of control alternatives by using a fuzzy gain factor is proposed. The value of the fuzzy scaling factor can be determined based on a simple decision making mechanism by using fuzzy criteria.

Application of the control techniques based on fuzzy decision making is presented in Chapter 12. An air-conditioning system is controlled by using inversion control based on Takagi–Sugeno fuzzy models, PID control, MBPC with classical objective functions and MBPC with fuzzy objective functions. The performances of different methods are compared with one another.

Finally, several promising directions for future research are presented in Chapter 13.

The material in this book assumes some degree of familiarity with conventional model-based control design schemes. The reader who is not familiar with the model-based predictive control and the internal model control schemes can refer to the appendices. Appendix A describes model-based predictive control, which is used as the main model-based control methodology in this book. The model–plant mismatches that diminish the performance of model-based predictive control are often dealt with using the internal model control (IMC) technique, which is described in Appendix B.

Chapter 2

Fuzzy Decision Making

Two main approaches to decision making can be distinguished in literature.

(1) *Descriptive decision making* considers a decision as a specific *information processing process*. It studies the cognitive processes that lead to decisions and the way information is processed in these processes. For example, the ways humans deal with conflicts, perceive the viability of the solutions and commit themselves to those solutions are studied (Janis and Mann 1977). Descriptive decision making searches for explanations for the ways individuals or groups of individuals arrive at decisions so that methods can be developed for influencing and guiding the decision process.

(2) *Normative decision making* considers a decision as a rational *act of choice* amongst the viable alternatives. This is a prescriptive view which studies mathematical theories for modeling decision making (Luce and Raiffa 1957). Classical decision making theory, operations research and most technical applications of decision making follow this approach. Normative decision making strives to make the optimal decision, given the available information. Hence, it is closely related to the optimization theory.

Mathematical analysis of control systems and the design of optimal controllers play an important role in control engineering. Therefore, the normative approach to decision making relates favorably to the control engineering practice. For that reason, the normative decision making approach is followed in the rest of this book. Most of the contribution of the fuzzy set theory to decision making has also been in the context of the normative approach. The theoretical formulation of a decision problem in the normative approach considers a decision as a momentary act (of choice). The applications of decision making, however, must also consider different aspects, such as data gathering, data analysis, or the reduction of complexity. The term 'decision process' is used in the following, when additional aspects for the specification of a decision problem in an application are considered in addition to the act of choice.

This chapter introduces the basics of fuzzy decision making in so far as they are of interest for this book. Emphasis lies on multicriteria, individual decisions. Single-step decisions are analyzed in detail, but multistage decisions are also considered when the application requires it. The outline of the chapter is as follows: in Sec. 2.1, we briefly discuss the classification of decision making methods; Sec. 2.2 contains a general framework for decision making, which is valid both for conventional decision making and fuzzy decision making. A general formulation of fuzzy decisions is given in Sec. 2.3; multiattribute fuzzy decision making is considered in detail in Sec. 2.4 since the methods in this book focus on multiattribute decision making, the elements of fuzzy multiattribute decisions are explained in greater detail; the conclusions and the discussion regarding the chapter are presented in Sec. 2.5. Note that even though this book concentrates on multiattribute decision making, most of the conclusions also apply to multiobjective decision making. For that reason, the terms *multicriteria* (covering both multiattribute and multiobjective decision making) and *multiattribute* decision making are used interchangeably.

2.1 Classification of decision making methods

Decision making problems involve many aspects, and depending on the properties that one wants to highlight, many classifications for the decision making problems are given. The main types of classification for the decision problems are the following.

- **Multistage vs. single-step decisions.** The selection of the best alternative from the available alternatives can occur in one stage by considering all the criteria simultaneously. These constitute the single-step decisions. Alternatively, the decisions may be taken in several steps, possibly iteratively. These are multistage decision problems. Multistage decisions can simplify the decision making process by dividing a large problem into smaller single-step decision problems which can be managed and analyzed more easily. Many control problems fall into this category.
- **Multiperson vs. individual decisions.** In many cases, the decision is taken by one decision maker who considers only his/her own goals. A decision may also require that the goals of a group of decision makers are considered. In that case, a multiperson decision is taken where, in addition to the specification of decision goals and constraints, one must consider the interaction amongst the decision makers and their influence on the decision. This is in general a more

complex decision problem, since the decision makers can form different sub-groups that follow different strategies. Some decision makers may cooperate, while others oppose one another. Some may be more dominating, while others are submissive. In general, the group dynamics must be taken into account in multiperson decision making. These type of decisions are also considered in the game theory (Leinfellner and Köhler 1998, Rosenmüller 2000).

- **Multicriteria vs. relatively simple optimization decisions.** Many activities, such as control, optimization and management, can be formulated as (a set of) decision problems. When optimizing, the decision is often based on a single criterion or on a set of criteria that are combined with (relatively simple) mathematical relations such as a summation. Well-known optimization methods can then be used for finding the optimal decision. In multicriteria decisions, several criteria are also considered, which are often mutually conflicting and a trade-off amongst them is needed. Multiple criteria must then be combined in a suitable manner. This combination should be sufficiently complex to reflect the goals of the decision. Alternatively, a hierarchy of criteria can also be established, and the best decision can be found by gradually limiting the solution set at each level of the hierarchy by taking into account a new criterion at each stage.
- **Operational vs. exploratory decisions.** Operational decisions are characterized by well-specified goals and the knowledge of the available alternatives. Moreover, the consequences and the rewards of alternatives are known. Since the goals of the decision making are known and the rewards can be expressed in mathematical terms, it becomes possible to formulate the decision making problem as an optimization problem that can be solved numerically. In exploratory decisions, in contrast, the goals of the decision are not known precisely at the moment that the decision procedure starts. It is also possible that the available alternatives are not known exactly, and that they only become available during the decision making process. The decision making then typically consists of a number of iterations which decrease the uncertainty at each iteration and which lead to the specification of decision goals and the alternatives.
- **Probabilistic vs. deterministic decisions.** When the environment in which the decision making takes place is known and when the consequences of the alternatives can be specified deterministically, one talks about deterministic decisions. It is also possible that some of the quantities that are relevant for decision making are of a probabilistic nature. In that case, one needs to take the probabilistic uncertainty into account and consider the expected gains from the alternative actions instead of the precise gains.

In control applications, the decisions are usually of an operational type since the goals are known beforehand. Often, a single controller determines the actions that need to be taken, and hence individual decisions are of interest. There are, however, multiple criteria to be considered. In feedback control, the controller output is determined from the values of the process signals that are fed back to the controller. These are often one-step decisions. When prediction is considered, or when the controller must be optimized over the system's future behavior, the control sequence must be determined for several stages after one another or for a sequence of sample instants. Hence, the type of decisions that are of interest in control engineering are in general *multistage, multicriteria individual operational* decisions. When the controlled process is deterministic, the decisions taken are also deterministic. Vagueness and non-probabilistic uncertainty can be accounted for by using decision making methods such as fuzzy decision making. Stochastic decision making can be used when the controlled process is stochastic. In the future, multiperson decisions are likely to become more important, as decentralized control with multiple cooperating controllers (for example, in multiagent control systems) are considered more and more.

2.2 General formulation of decision making

The normative approach to decision making considers a decision as the selection of the best alternative from available alternatives, given the information regarding the decision problem and the goals of the decision maker. The decision is often formulated as a quintuple $(\mathcal{A}, \Theta, \Xi, \kappa, D)$ (Grabisch et al. 1995). The symbols herein are defined as follows.

- $\mathcal{A}$ is the set of alternatives or possible actions. The decision maker is expected to make a selection from this set by using the available information.
- Θ is the set of the states* of the environment in which the decision making is taking place. These states are usually not known, although a probability, possibility or plausibility distribution may sometimes be available. The states of the environment cannot be controlled by the decision maker, but they must be dealt with in the decision making process.
- Ξ is the set of consequences. The consequences result from the choice of a particular alternative. An alternative can lead to several consequences, all of which may need to be taken into account in multicriteria decision making. In that case Ξ is multidimensional.

*The word 'states' is used here in a different meaning than 'state' in control engineering. The states of the environment describe various conditions of the environment in which decisions are taken. However, this does not imply that the states give all the information regarding the environment.

- κ is a mapping $\mathcal{A} \times \Theta \longrightarrow \Xi$, which specifies a resulting consequence for each element of the set of environment states Θ and each element of the set of alternatives $\mathcal{A}$. Ξ is the set of consequences of the decision alternatives. It is assumed that Ξ is known completely and that a consequence can be calculated for each alternative–state pair. The space $\mathcal{A} \times \Theta$ defines the solutions for the decision problem and it is sometimes also called *the solution space*.
- D is the decision function which is defined as $D : \Xi \longrightarrow \mathbb{R}$. It reflects the preference structure of the decision maker. The decision function D incorporates the goals of the decision maker. It induces a preference ordering on the set of consequences Ξ such that

$$\xi_i \succ \xi_j \text{ if and only if } D(\xi_i) > D(\xi_j), \tag{2.1}$$

where $\xi_i, \xi_j \in \Xi$ and $\succ$ is the preference relation, *i.e.*, consequence ξ_i is preferred to consequence ξ_j.

Note that in the deterministic case, κ maps each alternative to a consequence, and so decision function implicitly induces a preference ordering on $\mathcal{A}$ such that for alternatives $a_i, a_j \in \mathcal{A}$

$$a_i \succ a_j \text{ if and only if } D(\xi_i) > D(\xi_j).$$

Consider the following simplified decision problem, which illustrates the above formulation of the decision problem.

Example 2.1 A person is driving a car on a cold winter day down a road. Suddenly, a dog jumps in front of the car. Suppose that the driver can decide between two actions,

(i) he can break hard, applying full power to the brakes, or
(ii) he can brake soft, knowing that the car cannot come to a stop before a collision with the animal.

Because it is a cold winter day, it is possible that the road is slippery, but the driver has no knowledge of the state of the road. What should the driver do?

Figure 2.1 depicts the formulation of the decision problem. The set of the states of the environment are defined by

$$\Theta = \{\text{slippery road, not slippery road}\},$$

and the set of actions is defined by

$$\mathcal{A} = \{\text{brake soft, brake hard}\}.$$

The solution space $\mathcal{A} \times \Theta$ has four elements. The mapping κ associates for each alternative–state pair a consequence:

(1) if the road is slippery and the driver brakes soft, he will hit the dog slightly, causing minor damage to both the dog and his car;
(2) if the road is slippery and the driver brakes hard, the car will slip and hit a nearby tree, causing major damage to the car;
(3) if the road is not slippery and the driver brakes soft, he will hit the dog slightly, causing minor damage to both the dog and his car;
(4) if the road is not slippery and the driver brakes hard, he will not hit anything, and stop in time.

Finally, the decision function D maps the consequences to real numbers $\{D_1, D_2, D_3, D_4\}$, which imposes a preference ordering on the solution set.

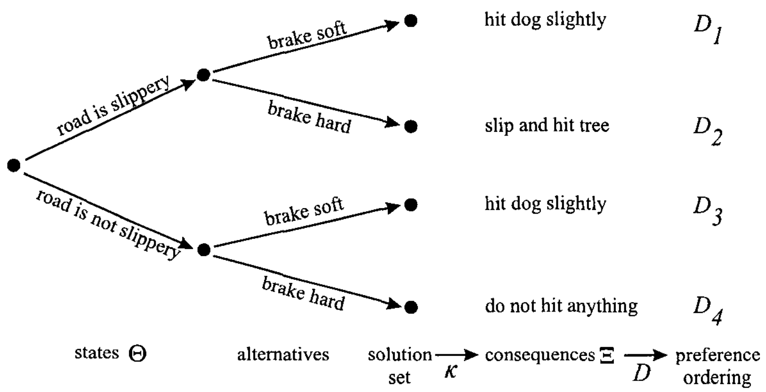


Fig. 2.1 The basic elements of a decision problem.

The consequences can also be expressed in terms of several criteria as shown in Table 2.1. In that case, D is a mapping from a multidimensional Ξ to $\mathbb{R}$. Since the state of the environment is not known by the driver, the decision function D should reflect the decision attitude of the driver and his willingness to take risks. This decision attitude depends on the expectations of the decision maker concerning the uncertainties in the decision making. For example, if the decision maker expects that the likelihood of a slippery road is approximately equal to the likelihood of a non-slippery road, a risk-aware decision maker will brake softly (*i.e.*, minimizes the worst consequence) while a risk-prone decision maker will brake hard (*i.e.*, maximizes the best consequence). However, if the decision maker expects that the road is much more likely to be non-slippery, then a risk-aware attitude may still lead to braking hard.

Table 2.1 Multidimensional consequences in multicriteria decision making.

Consequence	Damage to car	Damage to animal	Damage to tree
hit dog slightly	minor	minor	none
slip and hit tree	major	none	minor
do not hit anything	none	none	none

In addition to the attitude of the decision maker, the importance of various criteria must be taken into account in specifying D. In the example given, for instance, the criterion 'damage to tree' can be given a lower importance than the other two criteria since the consequences of damage to the tree are not very severe. Hence, the third criterion should have less influence on the outcome of the decision making than the other criteria.

2.3 Fuzzy decisions

Since the decision making involves the selection of the best available alternative, it is usually represented mathematically as an optimization problem. Following the formulation of Sec. 2.2, the optimization problem can be specified as follows. Consider first, the case where the states Θ of the environment are known to the decision maker. In that case, the elements of Θ can be incorporated in the set $\mathcal{A}$, so that κ is a mapping $\kappa : \mathcal{A} \longrightarrow \Xi$. The best decision alternative a^* is then given by

$$a^* = \max_{a \in \mathcal{A}} D(\kappa(a)) \tag{2.2}$$

where $\kappa(a) = [\kappa_1(a), \ldots, \kappa_n(a)]^T$ is a vector function for n different decision criteria. The notation $\kappa_j(a)$, $a \in \mathcal{A}$ is used to indicate that the mapping κ_j acts on the variables regarding the alternative $a \in \mathcal{A}$. When the states of the environment are not known, the results of the decision function for different states must be combined by another function $h : \mathbb{R}^k \longrightarrow \mathbb{R}$, for k different states. The optimization problem is then formulated as

$$a^* = \max_{a \in \mathcal{A}} h(D_1(\kappa(a, \theta_1)), \ldots, D_k(\kappa(a, \theta_k))). \tag{2.3}$$

It is seen from Eq. (2.3) that the presence of uncertain states does not essentially change the formulation of the decision making problem. The difference lies mainly in the definition of the decision functions and the interpretation of the final aggregation function h. For example, if the state uncertainties are given in terms of probabilities, the aggregation function represents a combination of probability values which can impose certain requirements on the specification of the aggregation function. Hence the states of the decision environment enter the decision problem in much the same way as different decision alternatives would. Because of this resemblance and for the sake of simplicity, the set of states Θ is assumed to be known in the following. In other words, the decision problems under certainty are considered, which can be formulated in terms of Eq. (2.2).

The set $\mathcal{A}$ of alternatives can not always be defined explicitly. In many cases, this set is defined implicitly by the specification of a number of constraints that need to be satisfied. Suppose that in this case the alternatives for a decision problem can be represented by vectors $\mathbf{x} \in \mathcal{A} \subset \mathbb{R}^n$. The optimization problem can then be formulated as

$$\begin{array}{l} \text{maximize } D(\mathbf{x}) \\ \text{subject to } g_i(\mathbf{x}) \geq 0, \quad i = 1, 2, \ldots, l, \end{array} \tag{2.4}$$

where $g_i(\mathbf{x})$ are the constraints imposed on the solution. Thus, the decision optimizes the overall decision function D, while the constraints define the set within which the search is performed (*i.e.*, they define $\mathcal{A}$). In this formulation, there is a clear distinction between the goals that are represented in the optimized objective function and the constraints. Because of this distinction, this decision model is also referred to as the *asymmetric model.*

Attention is now paid to *fuzzy decision making,* which is proposed to deal with non-probabilistic uncertainty and vagueness in the environment in which the decision making takes place. Two important elements of decision making are the goals of the decision that are represented by the maximized objective function and the imposed constraints that confine the search space. Fuzzy decision making essentially replaces the crisp goals and the constraints with their fuzzy equivalents. The following definitions are due to Bellman and Zadeh (1970).

Definition 2.1 Let $\mathcal{A}$ be a given set of possible alternatives which contains a solution to a decision making problem under consideration. A fuzzy goal G is a fuzzy set on $\mathcal{A}$ characterized by its membership function

$$\mu_G : \mathcal{A} \longrightarrow [0, 1],$$

which represents the degree to which the alternatives satisfy the specified decision goal.

In general, a fuzzy goal indicates that a target should be obtained, but it also quantifies the degree to which the target is fulfilled.

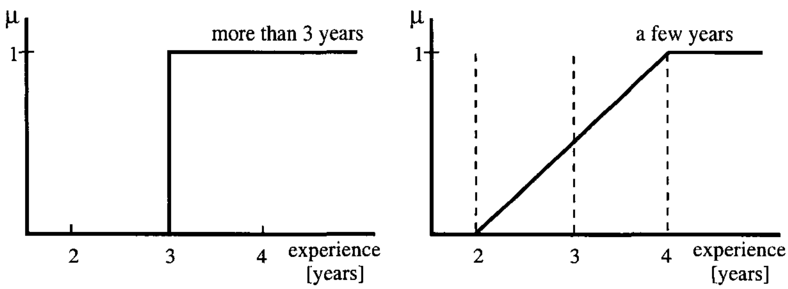
Definition 2.2 Let $\mathcal{A}$ be a given set of possible alternatives which contains a solution to a decision making problem under consideration. A fuzzy constraint C is a fuzzy set on $\mathcal{A}$ characterized by its membership function

$$\mu_C : \mathcal{A} \longrightarrow [0, 1],$$

which constrains the solution to a fuzzy region within the set of possible solutions.

A fuzzy constraint is a generalization of a crisp constraint. In fact, the support of a fuzzy constraint determines the set of alternatives in which the solution to the decision making problem lies. The support defines a crisp set and thus it defines the crisp constraints to the problem. The solution set, however, is not the crisp constraint set as is the case in conventional optimization. Instead, certain alternatives in the solution set satisfy the constraints completely, while others violate them to some degree.

Example 2.2 Suppose that the marketing department of a manufacturing company is looking for a sales engineer. Suppose further that the company policy requires that the recruits have had a few years of experience elsewhere. In conventional decision making, this constraint could be represented in crisp terms by a condition such as 'the recruit must have more than three years of work experience.' Figure 2.2a shows the membership function representation of this crisp set. The solution set now includes applicants with three or more years of experience. The fuzzy set 'a few years of experience' can be represented as shown in Fig. 2.2b. In this case, the solution set includes experiences over two years with different degrees of membership.



(a) Crisp *(b) Fuzzy*

Fig. 2.2 Crisp and fuzzy representation of a few years of work experience.

Realizing that a decision should satisfy the decision goals as well as the decision constraints, Bellman and Zadeh (1970) suggested the following model for the fuzzy decision.

Definition 2.3 Let $\mathcal{A}$ be a given set of possible alternatives which contains a solution to a decision making problem under consideration. Let G be the set of fuzzy goals for the decision, represented by the membership function $\mu_G(a),\ a \in \mathcal{A}$, and let C be the set of fuzzy constraints represented by the membership function $\mu_C(a),\ a \in \mathcal{A}$. Then the *fuzzy decision* F results from the intersection of the fuzzy decision goals and the fuzzy constraints, *i.e.*

$$F = G \cap C. \tag{2.5}$$

The fuzzy decision is characterized by its membership function

$$\mu_F(a) = \mu_G(a) \wedge \mu_C(a)\ a \in \mathcal{A}, \tag{2.6}$$

where $\wedge$ denotes the minimum operation.

Definition 2.4 The optimal decision a^* in fuzzy decision making is the decision with the largest membership value, also called *the maximizing decision*, which is defined by

$$a^* = \arg\max_{a \in \mathcal{A}} \mu_G(a) \wedge \mu_C(a). \tag{2.7}$$

It is important to realize that the distinction between the goals and the constraints disappears in this model. Essentially, both the goals and the constraints are represented by membership functions defined on the set of possible alternatives. The decision function (the conjunction in the model) makes an appropriate combination of the goals and the constraints. Because of the symmetry between the goals and the constraints, this model is sometimes called *a symmetric model.* Since both the goals and the constraints are represented by fuzzy sets, it is quite possible that the goals are achieved only partially while the constraints are violated slightly, as the following simplified example shows.

Example 2.3 A patient with hepatitis-C is going to receive interferon therapy. The medical specialist has to determine the optimum dosage of interferon α that the patient will get. The goal is to give as much of the drug as needed for efficiently counteracting the virus. However, the dosage is constrained by the negative effects of the drug on the body, especially the liver. Figure 2.3 shows the fuzzy sets that represent the fuzzy goal and the fuzzy constraint of the decision.

The set of alternatives $\mathcal{A}$ is the set of possible dosage values, represented here by real numbers. In general, more medicine is better. Thus, the goal is to administer as much medicine as possible. There is, however, a minimum amount below

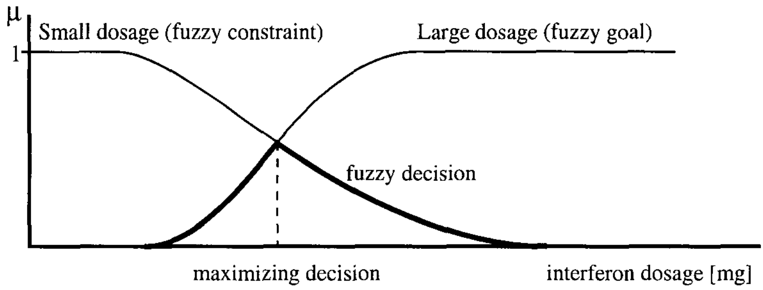


Fig. 2.3 Fuzzy decision for dosage determination in interferon therapy.

which the medicine has no influence and a maximum amount above which the effectiveness of the treatment does not increase any more. The constraint indicates that above a certain dosage the acceptability of the interferon therapy starts to decrease due to its adverse influence on the internal organs, up to a maximum value beyond which death would follow. Assuming that the goal and the constraint can be represented by the fuzzy sets that are depicted in Fig. 2.3, the optimal decision (*i.e.*, the optimal dosage) is given as shown in the figure.

2.4 Fuzzy multiattribute decision making

Multiattribute decision making is concerned with the selection of a decision alternative out of a finite and countable set of alternatives, taking into account multiple criteria that are of importance for the decision. Although the decision problem is essentially the same as in the multiobjective decision problems, the solution methods are more varied as the decision alternatives that need to be considered are known beforehand. Because of this knowledge, multiattribute decisions are used for modeling selection problems. This type of decision problem is denoted here by the term *discrete choice problem.*

Fuzzy multiattribute decision theory applies fuzzy set theory to solve the discrete choice problem which is concerned with the selection of the most suitable alternative out of a given set of possible alternatives, based on a number of decision criteria. The principle elements of the fuzzy set approach to the discrete choice problem can be summarized as follows.

- The set of alternatives.
- The set of goals and/or constraints (criteria) upon which the decisions are based.
- The judgments (ratings or membership values) per alternative per criterion,

which indicate the satisfaction of the criteria by the alternative.
- The weight factors, which indicate the importance that the decision maker attaches to various goals and/or constraints.
- A performance function, which orders the set of alternatives according to the satisfaction of the goals and/or constraints.
- An ordering mechanism for the ranking of alternatives.

Yager (1978) has proposed a fuzzy multiattribute decision making model, where the judgments are crisp numbers that are obtained from the membership of the alternatives to the fuzzy criteria. This model is used in the rest of this book. Other fuzzy multiattribute decision making models consider fuzzy judgments and possibly fuzzy weight factors. An extension of Yager's model with fuzzy evaluations is obtained when one uses type II fuzzy sets for specifying the fuzzy criteria. Type II fuzzy sets and their use in Yager's fuzzy decision model are not considered in this book. Below, the elements of Yager's model are reviewed in more detail.

2.4.1 *Alternatives*

The aim of the decision process is the selection of an alternative out of a set $\mathcal{A}$ of possible alternatives. This set has as its elements the possible outcomes $a_i, i = 1, \ldots, m$ of the decision process that is under consideration, *i.e.*,

$$\mathcal{A} = \{a_1, \ldots, a_m\}.$$

The set does not contain all the alternatives that may exist in the world, but only those that are available to the decision maker within the given context. The preselection of the available alternatives occurs during the structuring of the decision problem. Theoretically, additional alternatives might be thought of, but they will not be available to the decision maker. It is assumed that the alternatives possess properties which allow them to be compared with one another, so that the decision process can lead to a justified selection. These properties are expressed by the decision criteria.

Example 2.4 The fan of a temperature control system has four different velocity settings. A decision system can be used to determine the velocity setting that should be used. The alternative set consists of four velocity settings which are dictated by the system. The settings can be compared with one another based on various criteria, such as the power consumption of the setting and the expected drop in the temperature due to the setting.

2.4.2 *Decision criteria*

The decision is taken based upon a number of decision criteria. The best alternative is selected according to the degree to which the decision criteria are satisfied. In fuzzy decision making, the criteria consist of the decision goals and the constraints which are treated equivalently following the Bellman–Zadeh approach to fuzzy decision making. In terms of the general formulation of a decision problem, the criteria represent different aspects of the set of consequences Ξ. Let the decision criteria form the set

$$\mathcal{Z} = \{\zeta_1, \ldots, \zeta_n\}.$$

In order to compare the alternatives, it must be possible to evaluate them for different criteria. This implies that a function $\bar{g}_j$ can be specified for each criterion such that the mapping

$$\bar{g}_j : \mathcal{A} \longrightarrow X_j, \quad j = 1, \ldots, n \tag{2.8}$$

describes the values that the alternatives take on the variable form of the criterion. Herein, X_j is the domain over which the variable form of the criterion j is defined. The selection of the criteria outlines the structure of the decision problem and must be determined carefully by the decision maker by studying various factors that have influence on the outcome of the decision.

Example 2.5 Consider again the temperature control system of Example 2.4 with four fan settings. The criterion 'power consumption', for example, can lead to the following function values $\bar{g}_1(a_i), i = 1, 2, 3, 4$.

$$\bar{g}_1(a_1) = 0\,\mathrm{W} \quad \bar{g}_1(a_2) = 10\,\mathrm{W} \quad \bar{g}_1(a_3) = 25\,\mathrm{W} \quad \bar{g}_1(a_4) = 50\,\mathrm{W}. \tag{2.9}$$

X_1 in this example is the set of (non-negative) power consumption values.

2.4.3 *Membership values*

In order to be able to evaluate the alternatives based on the criteria, it should be known how the preference of the decision maker varies with the variable form of the criterion. For instance, if price is a decision criterion, it should be known what prices are acceptable for the decision maker and what prices are not. This information leads to a judgment for each alternative for each criterion, which indicates the desirability of the alternative in the corresponding criterion. In fuzzy decision making, the judgment information is usually provided in the form of membership values μ_{ij} which indicate the degree of satisfaction of the decision maker for each criterion. The membership values μ_{ij} for each alternative a_i and criterion ζ_j can be obtained by using membership functions that represent the fuzzy criteria. The membership values indicate how much that alternative satisfies the particular criterion. Hence, the term *membership value* is often used to denote the judgments in fuzzy decision making. The judgment information can be summarized in an *evaluation matrix* which looks like

$$\begin{array}{c} \\ a_1 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{ccc} \zeta_1 & \cdots & \zeta_n \end{array} \\ \begin{pmatrix} \mu_{11} & \cdots & \mu_{1n} \\ \vdots & \ddots & \vdots \\ \mu_{m1} & \cdots & \mu_{mn} \end{pmatrix} \end{array}. \tag{2.10}$$

When a membership function cannot be specified explicitly (*e.g.*in dealing with data that is subjective or difficult to measure), the membership values μ_{ij} can be determined directly by the decision maker and filled into the evaluation matrix.

The membership functions, from which the judgments are obtained, define fuzzy sets on the variable form of the criterion. The membership functions are given by the mappings

$$\mu_j : X_j \longrightarrow [0, 1], \qquad j = 1, \ldots, n. \tag{2.11}$$

By evaluating the membership of the alternatives to this fuzzy set, a fuzzy set F_j is defined on the set $\mathcal{A}$ of alternatives. The membership function of F_j is given by

$$\mu_{F_j}(a_i) = \mu_j(\bar{g}_j(a_i)) = \mu_{ij}, \quad i = 1, \ldots, m. \tag{2.12}$$

Each column of the evaluation matrix represents a fuzzy set F_j. The decision problem is the aggregation of the fuzzy sets $F_j, j = 1, \ldots, n$ into the overall fuzzy decision F. The evaluation matrix thus fixes the structure of the decision problem.

Example 2.6 One of the design goals for an electromagnetic component is its mass. Suppose that a design goal is defined as 'the mass of the component should be about 400g'. This goal can be represented by a fuzzy set as shown in Fig. 2.4, which denotes the satisfaction of the criterion as a function of $\bar{g}_j(a) \in X_j$, where X_j denotes the set of all possible component mass values. The fuzzy set 'about 400g' can also be interpreted as a definition of a fuzzy state.

Figure 2.4 also shows four alternatives with various mass. The fuzzy set F_j is then given by

$$F_j = \{(380, 0.6), (400, 1), (425, 0.5), (475, 0)\}. \tag{2.13}$$

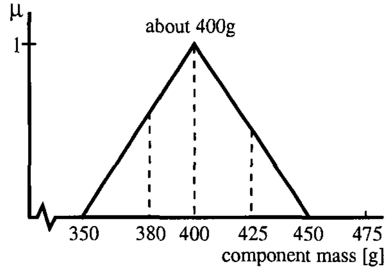


Fig. 2.4 Definition of fuzzy goal "about 400g".

2.4.4 *Weight factors*

Usually the decision criteria are not equally important for a particular decision problem. In this case, the criteria should have different influences on the result of the decision. More important criteria should influence the outcome of the decision more than the less important criteria. The difference in the importance of the criteria can be modeled by the introduction of weight factors into the decision problem. It is then possible to assign different weight factors to different criteria in order to model the difference in the importance of the decision criteria. The weight factors bring additional flexibility to the formulation of the decision problem by disturbing the symmetry between the criteria.

2.4.5 *Aggregation function*

When the evaluation matrix for a particular decision is determined and the corresponding weight factors are known, the structure of the discrete choice problem is fixed. At this point, the information about the alternatives needs to be combined in order to determine the overall suitability of the alternative. This is done by an aggregation function

$$D^{\mathrm{w}}(\mu_{i1}, \dots, \mu_{in}),\ i = 1, \dots, m \tag{2.14}$$

which is a function of the n membership values, n weight factors and possibly other (external) parameters. Note that this corresponds to the aggregation of columns in Eq. (2.10).

In fuzzy decision making, the aggregation function D is often a suitable aggregation operator with which the fuzzy sets $F_j, j = 1, \dots, n$ defined over $\mathcal{A}$ are combined. Since membership values lie in the unit interval, D is a mapping

$$D : \mathbb{I}^n \longrightarrow \mathbb{I}, \tag{2.15}$$

where $\mathbb{I}$ denotes the unit interval. The aggregation function in fuzzy decision making is usually called the *decision function*. The decision function should reflect the aims of the decision maker. Hence, it can be interpreted as a suitable translation of the goals of the decision into a mathematical representation. The decision maker may choose a decision function that suits his purposes. Furthermore, certain decision functions may be more suitable for certain types of decision problems. Hence the selection of a suitable decision function involves some flexibility and different possibilities exist. Consequently, many possible decision functions have been suggested in literature. The aggregation function may consist of a single aggregation operator from the fuzzy set theory, or it can be a combination of these operators. More information about the decision functions is given in Chapter 3. The decision function leads to the fuzzy decision F, which is defined as follows (following Bellman and Zadeh (1970)).

Definition 2.5 The aggregated fuzzy decision in multiattribute fuzzy decision making is a fuzzy set F defined on $\mathcal{A}$ and obtained by the combination of individual fuzzy criteria $F_j, j = 1, \ldots, n$, *i.e.*,

$$F = D^{\mathrm{w}}(F_1, \ldots, F_n). \tag{2.16}$$

2.4.6 *Ranking*

Since the aggregated result is a fuzzy set defined on $\mathcal{A}$, the maximizing decision is found by finding the height of F. The decision function assigns a value to each alternative, which indicates the overall suitability of the particular alternative. When this is a crisp value, the ordering of the combined judgments corresponds to the ranking of the alternatives such that

$$D(a_k) > D(a_l) \quad \Longleftrightarrow \quad a_k \succ a_l$$

or equivalently,

$$\mu_F(a_k) > \mu_F(a_l) \quad \Longleftrightarrow \quad a_k \succ a_l,$$

where $\succ$ represents the preference relation. The relation $D(a_k)$ indicates that the value of the function D is obtained from information regarding the alternative a_k. D can be a multivariable function. The function D is said to impose a preference ordering on the set of alternatives.

It is also possible that the outcome of the decision function is a fuzzy set itself. This is the case when the fuzzy sets F_j are of type II. When F_j are type II fuzzy sets, D is a mapping

$$D : \tilde{\mathcal{P}}([0,1]^n) \longrightarrow \tilde{\mathcal{P}}([0,1]), \tag{2.17}$$

where $\tilde{\mathcal{P}}([0,1])$ denotes the set of all fuzzy subsets of the unit interval. In that case, there is not a total ordering structure of the alternatives that uniquely determines the ranking. The partial ordering of the fuzzy evaluations must then be considered in order to determine the best alternative. This procedure is a decision making process itself and resembles the defuzzification procedure for the fuzzy controllers. The ranking of fuzzy evaluations has been the subject of many research texts. Baas and Kwakernaak (1977), Bortolan and Degani (1985), Buckley (1985), Chen (1985), Dias Jr. (1993), Dubois and Prade (1983), Kim and Park (1990), Liou and Wang (1992) and Tseng and Klein (1989) are several references on the subject.

2.4.7 *Overview of the decision model*

Given the described elements of fuzzy multiattribute decision making, the solution procedure can be summarized as follows.

- Determine the set of alternatives $\mathcal{A} = \{a_1, a_2, \ldots, a_i, \ldots, a_m\}$.
- Determine the set of criteria (goals and constraints) $\mathcal{Z} = \{\zeta_1, \zeta_2, \ldots, \zeta_j, \ldots, \zeta_n\}$, which will be used to evaluate the alternatives.
- Determine the mappings $\bar{g}_j$ and evaluate $\bar{g}_j(a_i), i = 1, \ldots, m, j = 1, \ldots, n$.
- Determine the fuzzy sets $F_j, j = 1, \ldots, n$ by using the mappings μ_j, $j = 1, \ldots, n$ and establish the evaluation matrix with judgments μ_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$.
- Determine mutual importances w_j, $j = 1, \ldots, n$ of the criteria. The decision problem is now represented by the following parameters.

$$\begin{array}{cccc}
\hline\hline
 & \zeta_1 & \cdots & \zeta_n \\
 & w_1 & \cdots & w_n \\
\hline
a_1 & \mu_{11} & \cdots & \mu_{1n} \\
\vdots & \vdots & \ddots & \vdots \\
a_m & \mu_{m1} & \cdots & \mu_{mn} \\
\hline\hline
\end{array} \tag{2.18}$$

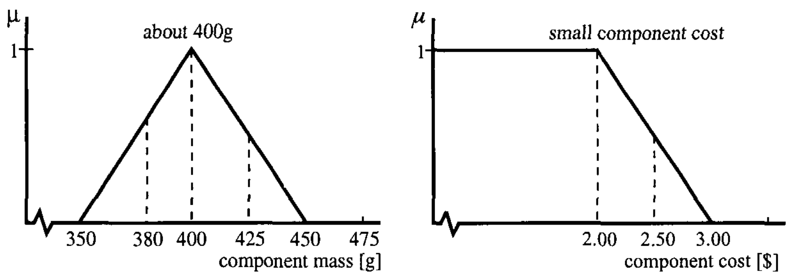
- Determine the final fuzzy decision F by evaluating the mapping $D^{\mathbf{w}}(\mu_{i1}, \ldots, \mu_{in})$, $i = 1, \ldots, m$.
- Select the alternative that maximizes D (*i.e.*, the maximal element of F).

The following example illustrates various stages of fuzzy multiattribute decision making.

Example 2.7 Consider a simplified design problem for an electromagnetic component such as an inductor. An inductor should satisfy certain criteria in order to be suitable for a particular application. Suppose that the component mass should be about 400g, while the cost should be small. First, several possible design alternatives must be determined. These alternatives are to be considered further in the decision process. Assume that four possible design alternatives, a_1, a_2, a_3, and a_4 are available for consideration. The set of alternatives is thus given by $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$. The decision criteria are ζ_1, 'mass about 400g', and ζ_2, 'small component cost'. Hence, $\mathcal{Z} = \{\zeta_1, \zeta_2\}$. The mapping $\bar{g}_1$ now corresponds to the measurement of the mass, and the mapping $\bar{g}_2$ corresponds to the evaluation of the price of the component. For each of the alternatives, the mass and the cost of the component are evaluated. Suppose that the following values are found.

$$\begin{array}{llll} \bar{g}_1(a_1) = 380\text{g} & \bar{g}_1(a_2) = 400\text{g} & \bar{g}_1(a_3) = 475\text{g} & \bar{g}_1(a_4) = 425\text{g} \\ \bar{g}_2(a_1) = \$2.50 & \bar{g}_2(a_2) = \$3.50 & \bar{g}_2(a_3) = \$2.00 & \bar{g}_2(a_4) = \$2.25. \end{array}$$

The fuzzy decision criteria are represented by their membership functions. Figure 2.5 shows the membership functions that might be used for representing 'about 400g' and 'small component cost'. Fuzzy sets F_j can now be determined by using



(a) About 400g

(b) Small cost

Fig. 2.5 Membership functions for two decision criteria.

Eq. (2.12). One obtains the following fuzzy sets from the membership functions in Fig. 2.5, and the evaluations of component mass and component cost.

$$F_1 = \{(380, 0.6), (400, 1), (475, 0), (425, 0.5)\}$$
$$F_2 = \{(2.50, 0.5), (3.50, 0), (2.00, 1), (2.25, 0.75)\}.$$

We assume that both criteria are equally important for the designer. Hence, the weight factors are set to $w_1 = 1$ and $w_2 = 1$. The decision problem is now represented by the following matrix.

	criterion	
	$w_1 = 1$	$w_2 = 1$
alternative	ζ_1	ζ_2
a_1	0.6	0.5
a_2	1	0
a_3	0	1
a_4	0.5	0.75

Empirical studies have shown that geometric mean is a suitable decision function for aggregating information in this design problem (Holt et al. 1997). When we use the geometric mean as the decision function, the final fuzzy decision F becomes

$$F = \{(a_1, 0.55), (a_2, 0), (a_3, 0), (a_4, 0.61)\}.$$

The decision that maximizes F is thus a_4, which is selected as the best decision alternative in this design problem.

2.4.8 *Relationship to other decision methods*

A detailed treatise of the fuzzy multiattribute decision model described in previous paragraphs can be found in Chen and Hwang (1992). The model is related to other methods known from the decision making theory. The membership functions from which the judgments are obtained can be compared to the utility functions in utility theory (Keeney and Raiffa 1976, Hwang and Masud 1979). By measuring the degree of satisfaction of a criterion by an alternative, the membership function fulfills a role similar to the utility that is obtained from the alternative. In utility theory, one tries to maximize the utility to be obtained from the possible alternatives. This corresponds to a maximizing decision. One of the multiattribute utility techniques that is used in literature is SMART (simple multiattribute rating technique) of Edwards (1977). In SMART, the judgments are elicited by a rating technique and the results aggregated using the weighted arithmetic mean operator (von Winterfeldt and Edwards 1986). The weight factors are normalized such that the sum of the weight factors is equal to 1. The fuzzy set approach follows a similar path, but uses operators from the fuzzy set theory for the aggregation step, so that more flexible aggregation behavior can be modeled. Moreover, the weight factors can be normalized in various ways, depending on the requirements of the decision problem. In addition to the rating techniques, the judgments can also be elicited by pairwise comparisons. This method requires more input from the decision maker, but in some cases it may be more reliable.

2.5 Summary and concluding remarks

Various classifications can be given for decision making problems depending on different aspects of the problem. Often, the decision problems are classified based on the number of decision stages, the number of decision makers, the number of decision criteria, the type of the decision problem and the type of uncertainty in the decision. Control applications often deal with multistage or single stage, single decision maker, multicriteria operational decisions. These decisions can be formulated as the selection of the best alternative out of a set of possible alternatives by considering a set of criteria. The ranking of the alternatives is determined by a decision function which maps the consequence of selecting a particular alternative to a real number. In fuzzy decision making, the consequence of an alternative is determined by its membership to the decision goals and constraints, and the aggregation operators for fuzzy sets from the fuzzy set theory are used as the decision functions. In this setting, the fuzzy decision making is formulated according to Bellman and Zadeh's symmetric model, in which the (fuzzy) decision goals and the (fuzzy) decision constraints are treated as equivalent concepts. Fuzzy multiattribute decision making also uses Bellman and Zadeh's approach to fuzzy decisions. Its basic elements are a countable and finite set of alternatives, a countable and finite set of criteria, judgments that indicate how each alternative satisfies each criterion, the weight factors and the decision function. This decision making model considers only type–I fuzzy sets for representing the uncertainty in the decision problem. The extension to type–II fuzzy sets is the subject of current research, as discussed in Mendel (2000) or Türkşen (1999).

Chapter 3

Fuzzy Decision Functions

The decision function reflects the aims of the decision maker for the decision process. The decision function depends on the type of aggregation required, the subjective goals of the decision maker and the boundary conditions imposed on the solution. Fuzzy decision making uses aggregation operators on fuzzy sets for obtaining different types of decision functions. This chapter gives an overview of the available aggregation operators for working with fuzzy sets, and it describes the most common type of aggregation behavior used in fuzzy decision making. These operators are considered in detail in so far as they are relevant to this book. Others are mentioned mainly for comprehensiveness.

The outline of the chapter is as follows: Sec. 3.1 discusses the main types of aggregation used in fuzzy decision making; Sec. 3.2 to Sec. 3.4 discuss different types of fuzzy aggregation operators that can be used in fuzzy decision making; weighted aggregation of decision criteria is considered in Sec. 3.5; finally, the conclusions of the chapter are presented in Sec. 3.6.

3.1 Main types of aggregation

Following the Bellman and Zadeh (1970) approach, the final fuzzy decision in fuzzy multiattribute decision making is arrived at by a confluence of the decision criteria, *i.e.*, the decision goals and constraints. Initially, the decision criteria were combined by the minimum operator, which models a conjunctive aggregation of the criteria. Nowadays, however, it is widely accepted that any suitable aggregation of fuzzy sets may be used in fuzzy decision making. Consequently, many aggregation operators have been proposed in literature (Bellman and Zadeh 1970, Dubois and Prade 1985, Dubois and Prade 1988, Dyckhoff and Pedrycz 1984, Mizumoto 1989b, van Nauta Lemke et al. 1983, Yager 1978, Yager 1988, Zadeh 1973, Zimmermann and Zysno 1980, Zimmermann 1987). These operators are used for modeling different types of decision behavior, and the decision maker may choose a decision function that best reflects the goals of the decision.

The following three types of aggregation are used most commonly in fuzzy decision making.

(1) Conjunctive aggregation of criteria
(2) Disjunctive aggregation of criteria
(3) Compensatory aggregation of criteria

Conjunctive aggregation of criteria implies simultaneous satisfaction of all decision criteria, while the disjunctive aggregation implies full compensation amongst the criteria. The compensatory aggregation is more suitable for dealing with conflicting criteria or with human aggregation behavior.

Example 3.1 Consider a linear, open-loop dynamic system described by the transfer function

$$\frac{Y(s)}{U(s)} = \frac{1}{s(s+0.5)}. \tag{3.1}$$

Suppose that the designer has to choose from three controllers. The step responses of the closed loop system have been plotted in Fig. 3.1.

The selection criteria are 'small overshoot' and 'fast settling time'. Controller 1 leads to the highest overshoot, but it is also the fastest. Controller 2 shows somewhat less overshoot, but it is also slower than Controller 1. Controller 3 leads to a response without overshoot but it is very slow. Assume that this information leads to the following judgments.

	criterion	
controller	small overshoot	fast settling time
1	0.2	0.9
2	0.4	0.6
3	1	0

The decision maker may want the closed-loop step response to have 'small overshoot' *and* 'small settling time'. The minimum operator can be used for this type of aggregation. Then, Controller 2 is selected as the best alternative since it scores best in the worst criterion. Alternatively, the decision maker may want the closed-loop step response to have 'small overshoot' *or* 'small settling time'. In contrast to the former conjunctive decision behavior, the latter is a disjunctive decision behavior which can be modeled by the maximum operator. In that case, Controller 3

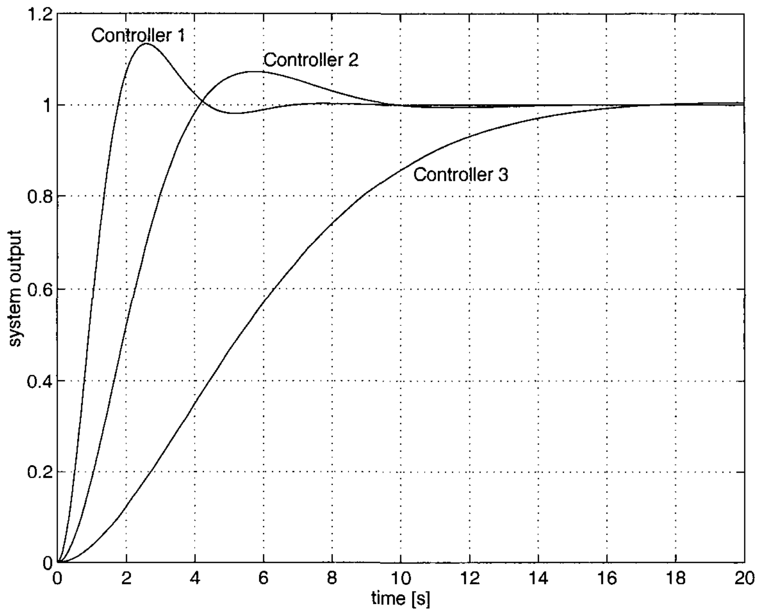


Fig. 3.1 Closed-loop step responses of three controllers for the system of Eq. (3.1).

is selected as the best alternative.

It may also be the case that the decision maker seeks compensatory decision behavior. When there are conflicting criteria, this type of aggregation may inherently be more suitable. In that case, are the boundary conditions on the decision, rather than the personal preferences of the decision maker that determine the choice of the decision function.

Example 3.2 Consider again the system given in Eq. (3.1). The decision maker now decides that a good characteristic for one of the criteria partially compensates for a bad characteristic of the other one. Some overshoot can be tolerated if the system becomes faster by doing so. In this case, the arithmetic mean can be used to aggregate the judgments. Controller 1 is then selected as the best alternative since the slight performance decrease in the overshoot compared to Controller 2 is compensated for by the large increase in the speed of response.

Fuzzy decision theory allows more complicated types of aggregation as well. For example, it is possible to use fuzzy quantifiers such as *most*, *some* and *several* to specify the decision goals.

Example 3.3 In designing an electromagnetic component, several criteria or attributes have to be considered, such as the peak flux density, the air-gap volume, total power loss of the component, its volume and mass, and the cost. Most of these criteria are conflicting and a trade-off amongst them is often necessary. Alternatively, the decision maker may decide to satisfy most of the criteria where the fuzzy quantifier *most* is defined by the decision maker. The ability to deal with fuzzy quantifiers is unique to fuzzy decision making.

Clearly, the selection of a decision function is of vital importance for the outcome of the decision model. For that reason it is important to study classes of aggregation operators that can be used as decision functions in fuzzy decision making, so that the properties of these operations and the type of decision behavior they model can be established. Fuzzy set theory provides the decision maker with a wide range of mathematical operators to model various types of decision behavior. The next sections consider important classes of fuzzy aggregation operators.

3.2 Triangular norms and conorms

3.2.1 *Triangular norms*

A triangular norm (t-norm) is a binary operation $T : [0,1]^2 \longrightarrow [0,1]$ that satisfies the following conditions for all $\mu_1, \mu_2, \mu_3, \mu_4 \in [0,1]$.

(1) boundary conditions:
$T(0,0) = 0, T(\mu_1, 1) = T(1, \mu_1) = \mu_1,$
(2) commutativity:
$T(\mu_1, \mu_2) = T(\mu_2, \mu_1),$
(3) monotone increasing:
$\mu_1 \leq \mu_2, \mu_3 \leq \mu_4 \Rightarrow T(\mu_1, \mu_3) \leq T(\mu_2, \mu_4),$
(4) associativity:
$T(\mu_1, T(\mu_2, \mu_3)) = T(T(\mu_1, \mu_2), \mu_3).$

T-norms are fuzzy set versions of the intersection operation on sets. In this sense, they are used for conjunctive type of aggregation. This is the type of aggregation when the decision maker wants to satisfy all the decision criteria simultaneously. Since simultaneous satisfaction of criteria can only lead to further constraint of the decision set, the addition of more criteria, in general, decreases the value of the overall aggregation. This is indicated by the following property

$$T(\mu_1, \mu_2) \leq \min(\mu_1, \mu_2). \tag{3.2}$$

In other words, the minimum operator is the largest t-norm. Values aggregated by other t-norms cannot be larger than the minimum of the values. A very large number of t-norms are known from the theory. Several important t-norms are: the minimum operator

$$T(\mu_1, \mu_2) = \min(\mu_1, \mu_2), \tag{3.3}$$

the product operator

$$T(\mu_1, \mu_2) = \mu_1 \cdot \mu_2, \tag{3.4}$$

and the bounded difference

$$T(\mu_1, \mu_2) = \max(0, \mu_1 + \mu_2 - 1). \tag{3.5}$$

In addition to the above operators, there are also families of t-norms. These are parametric operators which generate a large number of t-norms when the parameter values are varied. Several classes of t-norms that are also used in this book are the Yager t-norm

$$T(\mu_1, \mu_2) = \max(0, 1 - [(1 - \mu_1)^\gamma + (1 - \mu_2)^\gamma]^{1/\gamma}), \quad \gamma > 0, \tag{3.6}$$

the Hamacher t-norm

$$T(\mu_1, \mu_2) = \frac{\mu_1 \mu_2}{\gamma + (1 - \gamma)(\mu_1 + \mu_2 - \mu_1 \mu_2)}, \quad \gamma \geq 0, \tag{3.7}$$

the Schweizer and Skalar t-norm,

$$T(\mu_1, \mu_2) = 1 - [(1 - \mu_1)^\gamma + (1 - \mu_2)^\gamma - (1 - \mu_1)^\gamma (1 - \mu_2)^\gamma]^{1/\gamma}, \quad \gamma > 0. \tag{3.8}$$

and the Dubois t-norm,

$$T(\mu_1, \mu_2) = \frac{\mu_1 \mu_2}{\mu_1 \vee \mu_2 \vee \gamma}, \quad \gamma \in [0, 1]. \tag{3.9}$$

The Dubois t-norm is interesting as it is a combination of the product operator and the minimum operator. When $\mu_1, \mu_2 \in [0, \gamma]$, the aggregation is a scaled product operator, otherwise it is equal to the minimum operator. For an axiomatic definition of t-norms, the reader is referred to Dubois and Prade (1985), Klir and Yuan (1995) and Mizumoto (1989a). Other references to t-norms include Yager (1980), Giles (1976), Dubois and Prade (1988) and Dombi (1982).

3.2.2 *Triangular co-norms*

A triangular conorm (t-conorm) is a binary operation $S : [0,1]^2 \longrightarrow [0,1]$ that satisfies the following conditions for all $\mu_1, \mu_2, \mu_3, \mu_4 \in [0,1]$.

(1) boundary conditions:
$S(1,1) = 1, S(\mu_1, 0) = S(0, \mu_1) = \mu_1$,
(2) commutativity:
$S(\mu_1, \mu_2) = S(\mu_2, \mu_1)$,
(3) monotone increasing:
$\mu_1 \leq \mu_2, \mu_3 \leq \mu_4 \Rightarrow S(\mu_1, \mu_3) \leq S(\mu_2, \mu_4)$,
(4) associativity:
$S(\mu_1, S(\mu_2, \mu_3)) = S(S(\mu_1, \mu_2), \mu_3)$.

T-conorms are fuzzy set versions of the union operation on sets. In this sense, they are used for disjunctive type of aggregation. This is the type of aggregation when the decision maker wants to satisfy at least one decision criterion. Hence, this is a case of extreme compensation, where good performance in one of the criteria can compensate for poor performance in all the remaining criteria. Since good performance can compensate for poor performance, no matter how the alternative scores in the remaining criteria, the aggregation result can only increase when more criteria are considered. This is indicated by the following property

$$S(\mu_1, \mu_2) \geq \max(\mu_1, \mu_2). \tag{3.10}$$

In other words, the maximum operator is the smallest t-conorm. Values aggregated by other t-conorms cannot be smaller than the maximum of the values.

T-norms and t-conorms are related to each other according to the expression

$$T(\mu_1, \mu_2) = \neg(S(\neg(\mu_1), \neg(\mu_2))), \tag{3.11}$$

where $\neg(\mu_1)$ denotes the complement of μ_1. A triple $\{T, S, \neg\}$ which satisfies Eq. (3.11) constitutes a De Morgan triple where the t-norm T is said to be *the dual* of the t-conorm S with respect to fuzzy complement $\neg$. When Zadeh's complement $\neg(\mu_1) = 1 - \mu_1$ is used, Eq. (3.11) becomes

$$T(\mu_1, \mu_2) = 1 - S(1 - \mu_1, 1 - \mu_2), \quad \text{or} \tag{3.12}$$

$$S(\mu_1, \mu_2) = 1 - T(1 - \mu_1, 1 - \mu_2). \tag{3.13}$$

Because of the duality relation between the t-norms and the t-conorms, a very large number of t-conorms are also known from the theory. Below, the duals of the t-norms of Sec. 3.2.1 are given according to Eq. (3.13). The following are also the most important t-conorms that are used in literature, such as the maximum operator

$$S(\mu_1, \mu_2) = \max(\mu_1, \mu_2), \tag{3.14}$$

the algebraic sum

$$S(\mu_1, \mu_2) = \mu_1 + \mu_2 - \mu_1\mu_2, \tag{3.15}$$

and the bounded sum

$$S(\mu_1, \mu_2) = \min(\mu_1 + \mu_2, 1). \tag{3.16}$$

In addition to the above operators, there are also families of t-conorms. These are parametric operators which generate a large number of t-conorms when the parameter values are varied. Several classes of t-conorms are the Yager t-conorm

$$S(\mu_1, \mu_2) = \min(1, (\mu_1^\gamma + \mu_2^\gamma)^{1/\gamma}), \quad \gamma > 0, \tag{3.17}$$

the Hamacher t-conorm

$$S(\mu_1, \mu_2) = \frac{\mu_1 + \mu_2 + (\gamma - 2)\mu_1\mu_2}{1 + (\gamma - 1)\mu_1\mu_2}, \quad \gamma \geq 0, \tag{3.18}$$

and the Schweizer and Skalar t-conorm

$$S(\mu_1, \mu_2) = (\mu_1^\gamma + \mu_2^\gamma - \mu_1^\gamma\mu_2^\gamma)^{1/\gamma}, \quad \gamma > 0. \tag{3.19}$$

Because of the duality between t-norms and t-conorms, most of the literature on t-norms also considers the t-conorms. Hence, the interested reader is referred to references in Sec. 3.2.1.

3.3 Averaging and compensatory operators

3.3.1 *Averaging operators*

T-norms and t-conorms model simultaneous satisfaction of criteria and complete compensation amongst criteria, respectively. Usually, however, the decision behavior does not fall into one of these extreme cases. Instead, some kind of compensation amongst criteria is desired, so that good characteristics of an alternative compensate partially for the poor characteristics. Averaging operators model this compensatory decision behavior. Given $\mu_1, \mu_2, \mu_3, \mu_4 \in [0, 1]$, averaging operators satisfy the following conditions

(1) idempotent:
$M(\mu_1, \mu_1) = \mu_1$,

(2) commutativity:
$M(\mu_1, \mu_2) = M(\mu_2, \mu_1)$,
(3) monotone increasing:
$\mu_1 \leq \mu_2, \mu_3 \leq \mu_4 \Rightarrow M(\mu_1, \mu_3) \leq M(\mu_2, \mu_4)$

which imply

$$\min(\mu_1, \mu_2) \leq M(\mu_1, \mu_2) \leq \max(\mu_1, \mu_2). \tag{3.20}$$

Unlike the t-norms and the t-conorms, the averaging operators are not associative (except for the medians). Because of the compensation, the output is always between the minimum and the maximum elements.

Unlike the t-norms which model conjunctive aggregation (intersection) and the t-conorms which model disjunctive aggregation (union), the averaging operation has no counterpart in conventional set theory. The concept of partial membership to a set allows the calculation of the mean of two sets. Since most averaging operators are not associative, it is customary to consider the formulae for the n-ary operator instead of the binary operator. The averaging operator with n arguments is a mapping

$$M : \mathbb{R}^n \longrightarrow \mathbb{R}, \tag{3.21}$$

that is idempotent, commutative and non-decreasing in each component. A particular type of averaging operators are *the medians*. Medians are the only associative averaging operators and are defined as follows (Grabisch et al. 1995).

Definition 3.1 Given n different numbers $\mu_1, \ldots, \mu_n$ with $\mu_1 \leq \ldots \leq \mu_n$, their median is defined as

$$med(\mu_1, \ldots, \mu_n) = \begin{cases} \mu_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{1}{2}(\mu_{\frac{n}{2}} + \mu_{\frac{n}{2}+1}) & \text{if } n \text{ is even.} \end{cases} \tag{3.22}$$

Kolmogorov (1968) has studied the class of all decomposable continuous averaging operators given by

$$M_f(\mu_1, \ldots, \mu_n) = f^{-1}\left\{\frac{1}{n}\sum_{j=1}^{n} f(\mu_j)\right\}, \tag{3.23}$$

where f is a continuous, strictly monotonic function. A special case of this class of operators is the generalized averaging operator, sometimes also known as the *Minkovski operator* (Hardy et al. 1973). The generalized averaging operator is obtained from Eq. (3.23) by substituting μ_j^γ, $\mu_j \geq 0$ for $f(\mu_j)$. One obtains

$$M_\gamma(\mu_1, \ldots, \mu_n) = \left\{ \frac{1}{n} \sum_{j=1}^{n} \mu_j{}^\gamma \right\}^{1/\gamma}, \quad \mu_j > 0, \quad \gamma \in \mathbb{R}\backslash\{0\} \tag{3.24}$$

and

$$M_{\gamma=0}(\mu_1, \ldots, \mu_n) \triangleq \prod_{j=1}^{n} \mu_j{}^{1/n} \tag{3.25}$$

with

$$M_{\gamma\leq 0}(\mu_1, \ldots, \mu_n) \triangleq 0, \qquad \text{if } \exists \mu_j = 0, j = 1, \ldots, n. \tag{3.26}$$

The generalized averaging operator generalizes a large number of averaging operators. Well-known special cases of the generalized averaging operator are

$$M_{\gamma\to-\infty}(\mu_1, \ldots, \mu_n) = \bigwedge_{j=1}^{n} \mu_j, \quad \text{the minimum operator} \tag{3.27}$$

$$M_{\gamma=-1}(\mu_1, \ldots, \mu_n) = \frac{n}{\sum_{j=1}^{n} \frac{1}{\mu_j}}, \quad \text{the harmonic mean} \tag{3.28}$$

$$M_{\gamma=0}(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} \mu_j{}^{1/n}, \quad \text{the geometric mean} \tag{3.29}$$

$$M_{\gamma=1}(\mu_1, \ldots, \mu_n) = \frac{1}{n} \sum_{j=1}^{n} \mu_j, \quad \text{the arithmetic mean} \tag{3.30}$$

$$M_{\gamma=2}(\mu_1, \ldots, \mu_n) = \sqrt{\frac{1}{n} \sum_{j=1}^{n} \mu_j{}^2}, \quad \text{the quadratic mean} \tag{3.31}$$

$$M_{\gamma\to\infty}(\mu_1, \ldots, \mu_n) = \bigvee_{j=1}^{n} \mu_j, \quad \text{the maximum operator.} \tag{3.32}$$

The well-known averaging operators satisfy

$$M_{\gamma\to-\infty} \leq M_{\gamma=-1} \leq M_{\gamma=0} \leq M_{\gamma=1} \leq M_{\gamma=2} \leq M_{\gamma\to\infty}. \tag{3.33}$$

In fact, it is known from literature (Hardy et al. 1973) that Eq. (3.24) is monotonic non-decreasing in γ, *i.e.*,

$$\gamma < \gamma' \iff M_\gamma \leq M_{\gamma'}. \tag{3.34}$$

As the value of γ increases from $-\infty$ to ∞, the influence of the larger operand values on the aggregated result increases. When the averaging operator is used as a decision function, good characteristics of an alternative are emphasized for positive large γ values, while poor characteristics are emphasized for negative large values. For that reason, one can interpret the parameter γ as a characteristic index of optimism of the decision maker (Kaymak and van Nauta Lemke 1993).

Similar to the duality between t-norms and the t-conorms, averaging operators also have their dual. The dual of an averaging operator is again an averaging operator. When using Zadeh's complement, the dual generalized averaging operator M'_γ becomes for $\mu_j \in [0,1], j = 1, \ldots, n$

$$M'_\gamma(\mu_1, \ldots, \mu_n) = 1 - \left\{ \frac{1}{n} \sum_{j=1}^{n} (1 - \mu_j)^\gamma \right\}^{1/\gamma}, \quad \mu_j < 1, \quad \gamma \in \mathbb{R} \backslash \{0\} \tag{3.35}$$

and

$$M'_{\gamma=0}(\mu_1, \ldots, \mu_n) \triangleq 1 - \prod_{j=1}^{n} (1 - \mu_j)^{1/n} \tag{3.36}$$

with

$$M'_{\gamma \leq 0}(\mu_1, \ldots, \mu_n) \triangleq 1 \qquad \text{if } \exists \mu_j = 1, j = 1, \ldots, n. \tag{3.37}$$

From Eq. (3.2), Eq. (3.10) and Eq. (3.20) it is seen that the t-norms, averaging operators and the t-conorms cover the whole range of aggregation from the smallest t-norm T_W to the largest t-conorm S_W where T_W and S_W are defined as

$$T_W(\mu_1, \mu_2) = \begin{cases} \mu_1 \text{ if } \mu_2 = 1, \\ \mu_2 \text{ if } \mu_1 = 1, \\ 0 \quad \text{otherwise} \end{cases} \tag{3.38}$$

and

$$S_W(\mu_1, \mu_2) = \begin{cases} \mu_1 \text{ if } \mu_2 = 0, \\ \mu_2 \text{ if } \mu_1 = 0, \\ 1 \quad \text{otherwise,} \end{cases} \tag{3.39}$$

respectively. Figure 3.2 shows the scope of t-norms, averaging operators and t-conorms as fuzzy aggregation operators. The results of aggregating fuzzy sets for different types of operators are shown in Fig. 3.3.

Hardy et al. (1973) have studied generalized averaging operators extensively. Dyckhoff and Pedrycz (1984) have studied the generalized means within the context of fuzzy set theory, while Kaymak and van Nauta Lemke (1993) have studied

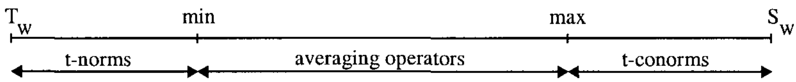


Fig. 3.2 The scope of fuzzy aggregation operators.

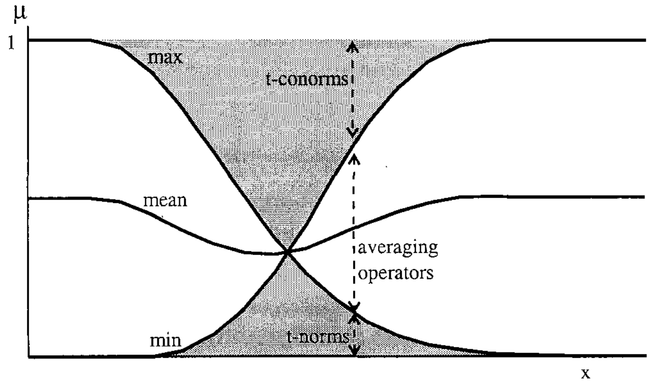


Fig. 3.3 The results of aggregating fuzzy sets with different types of aggregation operators.

this class of operators within the fuzzy decision making context. Other references on the averaging operators include Klir and Yuan (1995), Dubois and Prade (1988), Dubois and Prade (1985), and Yager and Filev (1994).

3.3.2 *Compensatory operators*

Thole et al. (1979) and Zimmermann (1978) have found indications that t-norms and t-conorms are unsuitable for modeling of aggregation by human decision makers. It appears that human beings tend to partially compensate between criteria instead of trying to satisfy them simultaneously or make complete compensations. It has been suggested in the literature that human beings use a mixture of conjunction and disjunction in their decisions. To model this, *compensatory operators* have been proposed, the general form of which has been defined by Mizumoto (1989b) as

$$E(\mu_1, \mu_2) = M\left(O_1(\mu_1, \mu_2), O_2(\mu_1, \mu_2)\right) \tag{3.40}$$

where $M(\mu_1, \mu_2)$ is an averaging operator and $O_1(\mu_1, \mu_2)$ and $O_2(\mu_1, \mu_2)$ are t-norms, t-conorms or averaging operators. Special cases of Eq. (3.40) have been suggested and investigated by Zimmermann and Zysno (1980). These operators have the form

$$E(\mu_1, \mu_2) = (T(\mu_1, \mu_2))^{1-\gamma} (S(\mu_1, \mu_2))^{\gamma} \tag{3.41}$$

and

$$E(\mu_1, \mu_2) = (1-\gamma)T(\mu_1, \mu_2) + \gamma S(\mu_1, \mu_2), \tag{3.42}$$

with $\gamma \in [0, 1]$. Hence, a compensatory operator is defined as a weighted average between a t-norm and a t-conorm. The aggregated result is a compensation between a conjunctive aggregation (intersection) and a disjunctive aggregation (union). By varying the value of the parameter γ, the emphasis can be shifted from the conjunctive to the disjunctive behavior as desired. Because γ controls the degree of 'or-ness' of the resulting operator, and hence indirectly also the degree of compensation between the two types of aggregation behavior, it is interpreted as the 'grade of compensation'. The scope of the compensatory operators extends from the t-norm to the t-conorm used in the definition, and the compensatory operator exhibits a mixture of conjunctive, disjunctive and averaging behavior.

The selection of different γ values, and different intersection and union operators leads to different aggregation operators. Usually the minimum and the maximum operators are used for the intersection and the union, respectively. Hurwicz's optimism–pessimism operator

$$E_H(\mu_1, \mu_2) = (1-\gamma)\min(\mu_1, \mu_2) + \gamma \max(\mu_1, \mu_2) \tag{3.43}$$

from the decision theory is such a compensatory operator (Luce and Raiffa 1957). Zimmermann and Zysno (1980) suggest the use of the algebraic product and the algebraic sum together with the geometric mean, leading to

$$E_Z(\mu_1, \mu_2) = (\mu_1\mu_2)^{1-\gamma}(\mu_1 + \mu_2 - \mu_1\mu_2)^{\gamma}. \tag{3.44}$$

The use of operators other than minimum and maximum allows the modeling of interaction between the operands. Figure 3.4 shows the aggregation of two fuzzy sets when the Zimmermann operator Eq. (3.44) is used.

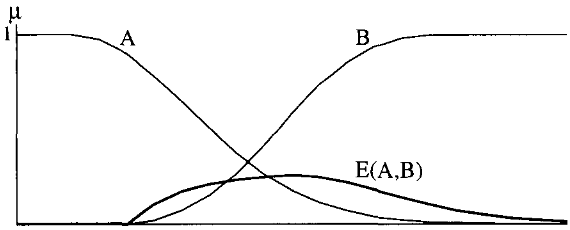


Fig. 3.4 The aggregation of two fuzzy sets with the Zimmermann operator ($\gamma = 0.5$).

3.3.3 *Associative compensatory operators*

Compensatory operators based on the generalized means are not associative. Several authors have investigated mathematical operators that are associative and possess compensatory characteristics. Mesiar (1995) calls these class of associative operators *associative compensatory operators*, while Yager and Rybalov (1995) call them *uninorm aggregation operators*. In the following, the nomenclature of Mesiar (1995) is followed.

An associative compensatory operator is a binary operation

$$\Gamma : [0,1]^2 \backslash \{(0,1),(1,0)\} \longrightarrow [0,1]$$

that satisfies the following conditions for all $\mu_1, \mu_2, \mu_3, \mu_4 \in (0,1)$.

(1) boundary conditions:
$\Gamma(1,1) = 1, \Gamma(0,0) = 0,,$
(2) monotone increasing:
$\mu_1 \leq \mu_2, \mu_3 \leq \mu_4 \Rightarrow \Gamma(\mu_1, \mu_3) \leq \Gamma(\mu_2, \mu_4),$
(3) associativity:
$\Gamma(\mu_1, \Gamma(\mu_2, \mu_3)) = \Gamma(\Gamma(\mu_1, \mu_2), \mu_3),$
(4) Γ is continuous.

The commutativity property follows from these conditions (Mesiar 1995). Associative compensatory operators exhibit reinforcement behavior under certain conditions. They can be used to model reinforcement in decision aggregation. The following example illustrates the need for reinforcement in aggregation.

Example 3.4 In order to achieve a high production rate in a yeast fermentation process, it is important to prevent the production of alcohol by the bio-organisms. The production of alcohol adversely influences the formation of the required products, and stimulates further production of alcohol, which further deteriorates the fermentation process. It is therefore important to monitor the production process by a fault diagnosis system, which can detect errors that result in conditions that lead to the production of alcohol. Consider two simplified rules from a fuzzy rule-based fault diagnosis system for yeast fermentation, which detects errors in relevant system quantities based on measured and/or computed data.

(1) If ammonia consumption is very high and oxygen consumption is very high and carbon dioxide consumption is low, then possible error in carbon dioxide consumption.
(2) If dissolved oxygen is high and oxygen coefficient is very high, then possible error in carbon dioxide consumption.

Assume now that both rules are valid to a relatively high degree (*e.g.*0.8). Since strong evidence for the same consequence comes from two different sources (the rules), it is more likely that an error in carbon dioxide consumption has occurred. Hence, the two sets of evidence should reinforce each other and the aggregated result should be higher than 0.8 which suggests a disjunctive aggregation. Conversely, if both rules fire to a low degree (*e.g.*0.2 and 0.1), then it is likely that the error has not occurred. In that case, the aggregated result should be lower than $\min(0.1, 0.2)$ which suggests a conjunctive aggregation. If one of the rules indicates strong evidence for the error and the other one does not, one can take the average.

An associative compensatory operator Γ has a neutral element μ_e^* for which $\Gamma(\mu_1, \mu_e^*) = \mu_1$. Then, for all $\mu_1, \mu_2 \in [0, 1]$

$$\Gamma(\mu_1, \mu_2) \geq \max(\mu_1, \mu_2), \quad \mu_1, \mu_2 \geq \mu_e^* \tag{3.45}$$

$$\Gamma(\mu_1, \mu_2) \leq \min(\mu_1, \mu_2), \quad \mu_1, \mu_2 \leq \mu_e^* \tag{3.46}$$

$$\Gamma(\mu_1, \mu_2) \in [\mu_1, \mu_2], \quad \text{otherwise.} \tag{3.47}$$

The neutral element μ_e^* divides the unit square $[0, 1] \times [0, 1]$ into four quadrants. Depending on the values of the operands μ_1 and μ_2, the associative compensatory operators exhibit t-conorm, t-norm or averaging operator behavior as shown in Fig. 3.5. Note that when $\mu_e^* = 1$, the associative compensatory operator reduces to a t-norm, and it reduces to a t-conorm when $\mu_e^* = 0$.

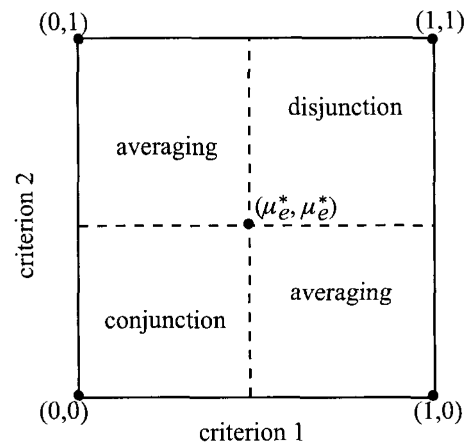


Fig. 3.5 Depending on the values of the judgments, associative compensatory operators exhibit different aggregation behavior.

An example of an associative compensatory operator is the tangent associative

Table 3.1 Several aggregation results using the associative compensatory operator Eq. (3.48).

μ_1	μ_2	$\Gamma_t(\mu_1, \mu_2)$	Aggregation
0.1	0.3	0.082	conjunctive
0.2	0.8	0.500	averaging
0.3	0.6	0.378	averaging
0.5	0.5	0.500	neutral
0.7	0.5	0.700	neutral
0.7	0.4	0.622	averaging
0.9	0.8	0.930	disjunctive

compensatory operator given by

$$\Gamma_t(\mu_1, \mu_2) = 0.5 - \frac{\arctan(\cot \pi\mu_1 + \cot \pi\mu_2)}{\pi}. \tag{3.48}$$

The neutral element of Eq. (3.48) is 0.5. Table 3.1 illustrates the averaging and the reinforcement behavior of Eq. (3.48) on some numerical values.

3.3.4 *Ordered weighted averaging operators*

A new compensatory aggregation technique based on averaging operators has been proposed by Yager (1988). The *ordered weighted averaging operators* makes use of a linear weighted averaging aggregation with a nonlinear ordering method.

Definition 3.2 An ordered weighted averaging (OWA) operator of n variables is a mapping

$$W : \mathbb{R}^n \longrightarrow \mathbb{R} \tag{3.49}$$

with an associated vector $\mathbf{w}^T = (w_1, w_2, \ldots, w_n)$ where the weight factors satisfy

$$\sum_{j=1}^{n} w_j = 1, \quad w_j \in [0, 1], j = 1, \ldots, n, \tag{3.50}$$

such that

$$W^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \sum_{j=1}^{n} w_j \mu_j \tag{3.51}$$

with the convention $\mu_1 \leq \cdots \leq \mu_n$.

The re-ordering step is a fundamental property of OWA operators. This implies that a particular weight is not associated with a particular operand, but it is associated with an ordered position of the operand.

The flexibility of the OWA operators is due to the possibility of specifying different weight vectors together with the re-ordering of the alternatives. Different weight vectors lead to different aggregation operators. For example, for $\mathbf{w}^T = (0, 0, \ldots, 1)$

$$W^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \bigvee_{j=1}^{n} \mu_j, \tag{3.52}$$

while for $\mathbf{w}^T = (1, 0, \ldots, 0)$

$$W^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \bigwedge_{j=1}^{n} \mu_j, \tag{3.53}$$

and for $\mathbf{w}^T = (1/n, \ldots, 1/n)$

$$W^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \frac{1}{n} \sum_{j=1}^{n} \mu_j. \tag{3.54}$$

Example 3.5 Suppose that a set of judgments for an alternative is given as $\mu_i = (0.3, 0.1, 0.5, 0.4)$. The transposed ordered judgment vector is then given by $\mu = (0.1, 0.3, 0.4, 0.5)$. If the associated weight vector is $\mathbf{w}^T = (0.2, 0.4, 0.1, 0.3)$, then the aggregated membership value using the OWA operator is given by

$$W^{\mathbf{w}}(0.3, 0.1, 0.5, 0.4) = 0.2 \times 0.1 + 0.4 \times 0.3 + 0.1 \times 0.4 + 0.3 \times 0.5 = 0.33.$$

A desired property of the OWA operators is that fuzzy quantifiers can be used to determine the weight vector $\mathbf{w}$. In this way, it becomes possible to transform the linguistic quantifiers into a set of weight factors that can be used in decision aggregation.

Example 3.6 Let $n = 5$. The goal 'several criteria must be satisfied' with the fuzzy quantifier 'several' can be represented by the weight vector $\mathbf{w}^T = (0, 0, 0.1, 0.2, 0.7)$. The zero elements in the weight vector imply that more than two criteria must be satisfied and have a high value in order for the aggregated value to be high. Hence, the weight vector introduces a thresholding effect.

The OWA operators are mentioned for completeness here, and they are not considered in detail in this book. More information on OWA operators can be found in Yager (1988), Grabisch et al. (1995), Klir and Yuan (1995), and Fodor et al. (1995).

3.4 Generalized operators

3.4.1 *Monotonic identity commutative aggregation operators*

A further generalization of the associative compensatory operators has been proposed by Yager (1994). These operators have been called *Monotonic Identity Commutative Aggregation* (MICA) operators. The concept of *a bag* can be used to define the MICA operators.

Definition 3.3 A bag $\langle \mu_1, \ldots, \mu_n \rangle$ drawn from an interval I is a collection of elements $\mu_1, \ldots, \mu_n$, all of which are contained in I.

A bag is similar to a subset in that an ordering of elements in the bag does not matter. It is different, however, from a subset in that a bag may contain multiple copies of the same element.

Definition 3.4 Let $\mathcal{U}$ be the set of all bags drawn from $\mathbb{I}$. A function $f : \mathcal{U} \longrightarrow \mathbb{I}$ is called a *bag mapping* from $\mathcal{U}$ into the unit interval.

Definition 3.5 Assume that A and B are two bags with the same cardinality. If the elements of A and B can be indexed in such a way that $\mu_j \geq \nu_j, j = 1, \ldots, n$, it is denoted by $\mathrm{A} \geq \mathrm{B}$.

Let $\oplus$ denote the concatenation of two bags. Then the MICA operators are defined as follows.

Definition 3.6 A bag mapping $\mathcal{M} : \mathcal{U} \longrightarrow \mathbb{I}$ is called a MICA operator if it satisfies

(1) monotonicity: $\mathrm{A} \geq \mathrm{B} \quad \Longrightarrow \quad \mathcal{M}(\mathrm{A}) \geq \mathcal{M}(\mathrm{B})$,
(2) identity: for every bag A there exists an element $\mu_e^* \in \mathbb{I}$, called the *identity* of A under $\mathcal{M}$, such that $\mathcal{M}(\mathrm{A} \oplus \langle \mu_e^* \rangle) = \mathcal{M}(\mathrm{A})$.

If the identity element is fixed, *i.e.*, it is independent of the bag A, the MICA operators reduce to the associative compensatory operators.

Although the t-norms and the t-conorms are special cases of associative compensatory operators, and associative compensatory operators are special cases of MICA operators, a separate consideration of different types of aggregation operators is preferred, since they represent different decision behaviors. General representations of operators are useful in understanding the relations between different types of decision behavior, but they are not very useful when studying specific aggregation behavior for the influence and interpretation of various parameters. For that reason, the MICA operators are not considered in detail here. For more information concerning the MICA operators, the reader can consult Yager and Filev (1994), Yager (1994), Yager and Rybalov (1995).

3.4.2 *Fuzzy integrals*

As mentioned in Sec. 3.4.1, the generalization of the t-norms and the t-conorms leads to MICA operators. Fuzzy integrals can be interpreted as the generalization of specific types of averaging operators. In particular, medians and OWA operators are shown to be specific cases of fuzzy integrals. The concept of a fuzzy integral is based on *fuzzy measures* (Sugeno 1974).

Definition 3.7 Let $\mathcal{Z} = \{\zeta_1, \ldots, \zeta_n\}$ be the set of decision criteria. Let $\mathcal{P}(\mathcal{Z})$ denote the power set of $\mathcal{Z}$, *i.e.*, the set of all subsets of $\mathcal{Z}$. A *fuzzy measure* on $\mathcal{Z}$ is a function $f_g : \mathcal{P}(\mathcal{Z}) \longrightarrow [0,1]$ that satisfies

(1) $f_g(\emptyset) = 0, g(\mathcal{Z}) = 1$.
(2) $Z_A \subset Z_B \subset \mathcal{Z} \quad \Longrightarrow f_g(Z_A) \leq f_g(Z_B)$.

Using fuzzy measures, Sugeno (1974) proposed the concept of fuzzy integral, which Grabisch (1996) interprets 'as a kind of distorted mean' in the discrete case. Sugeno (1974) suggested a fuzzy integral based on the minimum and maximum operators.

Definition 3.8 Given a fuzzy measure f_g on $\mathcal{Z}$, the *Sugeno integral* of a function $f_h : \mathcal{Z} \longrightarrow [0,1]$ with respect to f_g is defined by

$$S_g(f_h(\zeta_1), \ldots, f_h(\zeta_n)) = \bigvee_{j=1}^{n} \left(f_h(\zeta_{(j)}) \wedge f_g((Z_A)_{(j)})\right). \tag{3.55}$$

In Eq. (3.55), $(\cdot)_{(j)}$ indicates that the indices have been permuted so that $0 \leq f_h(\zeta_{(1)}) \leq \cdots \leq f_h(\zeta_{(n)}) \leq 1$, and $(Z_A)_{(j)} = \{\zeta_{(j)}, \ldots, \zeta_{(n)}\}$.

Later, Murofushi and Sugeno (1991) proposed a fuzzy integral based on linear operators.

Definition 3.9 Given a fuzzy measure f_g on $\mathcal{Z}$, the *Choquet integral* of a function $f_h : \mathcal{Z} \longrightarrow [0,1]$ with respect to f_g is defined by

$$C_g(f_h(\zeta_1), \ldots, f_h(\zeta_n)) = \sum_{j=1}^{n} (f_h(\zeta_{(j)}) - f_h(\zeta_{(j-1)})) f_g((Z_A)_{(j)}). \tag{3.56}$$

In Eq. (3.56), $(\cdot)_{(j)}$ indicates that the indices have been permuted so that $0 \leq f_h(\zeta_{(1)}) \leq \cdots \leq f_h(\zeta_{(n)}) \leq 1$, $f_h(\zeta_{(0)}) = 0$, and $(Z_A)_{(j)} = \{\zeta_{(j)}, \ldots, \zeta_{(n)}\}$.

Kandel and Byatt (1978) have shown that the Sugeno integral is an associative averaging operator. The Choquet integral can also be formulated as an associative averaging operator to correspond to the median. Grabisch et al. (1995) have shown that the Choquet integral also covers the OWA operators, and in particular the weighted arithmetic mean, where the weights w_j are represented by the fuzzy measure $f_g(\zeta_j)$.

The advantage of using fuzzy integrals as fuzzy aggregation operators is that they can be used to model interaction amongst criteria in an explicit fashion. By considering subsets of decision criteria, fuzzy integrals can model different types of interaction, including synergy and redundancy amongst criteria. The interaction between pairwise combination of criteria as well as higher order combinations can be modeled. This property renders fuzzy integrals flexible operators. This flexibility, however, is obtained by the specification of $2^n - 2$ parameters (for all possible combinations except for the empty and the total sets) in a problem with n criteria. Hence, for problems with a high number of criteria, the determination of the fuzzy integral parameters (the values of the fuzzy measure f_g) can be cumbersome, if not impossible. Guidelines exist for determining the fuzzy integral parameters. However, the dimensionality of the problem remains. For small problems, Grabisch (1996) suggest the use of domain knowledge for specifying the problem. For medium size problems, or in cases where data about the decision problem can be collected, the identification can be made using optimization techniques. A formulation of the identification problem that leads to a quadratic programming formulation is given in Grabisch et al. (1995). It is mentioned, however, that the conditioning of the matrices may present a problem. Moreover, $n!/[(n/2)!]^2$ data points are required (Grabisch 1996). Therefore, large problems remain intractable with this method, and heuristic approaches have been suggested by Ishii and Sugeno (1985). The heuristic methods do not guarantee optimality, but they reduce the dimensionality problem.

Partly due to the identification problem of the fuzzy measure values, the applications of fuzzy integrals have remained limited despite their flexibility. The interested reader is referred to Sugeno (1974), Murofushi and Sugeno (1991), Grabisch (1993), Grabisch et al. (1995) and Grabisch (1996).

3.4.3 *Rule-based mappings*

It is known that fuzzy systems are universal function approximators (Wang 1992, Kosko 1994). Hence, a rule based fuzzy system can be used to approximate a decision function that would represent the preference structure of the decision maker, similar to the way fuzzy rule-based systems are used to specify a controller's input–output mapping. The decision maker can articulate his preference information in the form of fuzzy rules which are used for the design of the fuzzy system. It is possible that the decision maker uses different types of aggregation for different performance regions. Different types of aggregation can be specified in different rules, which are then combined by the fuzzy inference mechanism. In addition to the fuzzy rules, the mapping described by a fuzzy system is determined by the definition of the membership functions, the inference mechanism and the defuzzification method (see also Sec. 4.2). If the preference function should satisfy certain constraints (*e.g.*the monotonicity of the decision function in its parameters as is usually the case), the determination of the fuzzy system may prove to be tedious. The determination of the linguistic rules is then more difficult. In literature, the combination of decision criteria using linguistic rules has found little application. A few examples can be found in Mandić and Mamdani (1984) and Efstathiou (1984).

3.4.4 *Hierarchies of operators*

Another way of bringing more flexibility to the definition of the decision function is the use of hierarchies of aggregation operators. Three goals are aimed at by establishing a hierarchy of decision functions.

(1) It becomes possible to model interaction amongst various criteria and to obtain more complicated decision functions.
(2) The formation of the hierarchy helps to divide a complex problem into smaller sub-problems which are more tractable for analysis.
(3) The criteria which are logically related can be grouped together, increasing the transparency of the decision making.

At different levels of the hierarchy, the information regarding the criteria can be combined with different aggregation operators. As far as the interaction amongst criteria is concerned, the interacting criteria can be combined with different aggregation operators so that positive interaction (synergy) or negative interaction (redundancy) can be dealt with. The division of a complex problem into simpler sub-problems increases the ability of the decision maker to determine the correct aggregation for a given set of criteria. The criteria that are logically related can be grouped together, which improves the tractability of the decision problem.

Example 3.7 Consider a controller whose gain factor must be adjusted for a process. The user may consider the following criteria and the available information.

(1) Speed of the response, which implies a large gain factor.
(2) Energy consumption of the system, which implies a not very large gain factor.
(3) The accuracy of the closed loop system, which implies a large gain factor.
(4) The stability of the closed loop system, which implies a moderate gain factor.

The decision problem is formulated as the fulfillment of the goals G_1 (maximization of the speed of response) and G_2 (minimization of energy consumption), while the two constraints C_1 (accuracy) and C_2 (stability) are satisfied. Suppose that the decision maker uses fuzzy decision analysis and wants to specify a decision function for combining the fuzzy goals and fuzzy constraints. The constraints must be satisfied simultaneously. The goals, however, can be combined in a compensatory manner, since an increase in the energy consumption can be compensated to some degree by an increase in the speed of the response, and vice versa. To model the different types of decision behavior, the decision maker decides to establish a hierarchy of decision functions. The constraints are combined by using a t-norm T_1 (*e.g.*minimum) for modeling the simultaneous satisfaction, and the goals are combined by using an averaging operator M (*e.g.*arithmetic mean). The aggregated results are then combined in a higher level with t-norm T_2 (*e.g.*product operator), which can be a different operator from T_1. Figure 3.6 shows graphically the hierarchical tree that is formed.

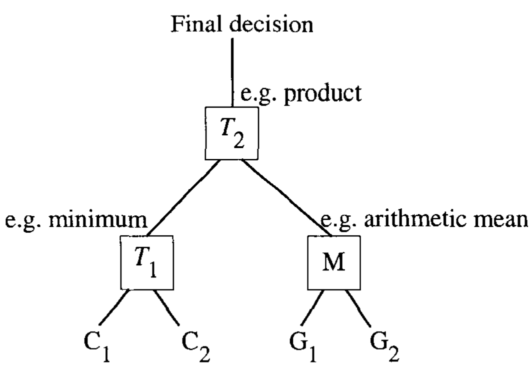


Fig. 3.6 A decision tree showing the combination of decision goals and constraints by different operators.

3.5 Weighted aggregation

Most aggregation operators that are used as decision functions assume a symmetric aggregation of the judgments, in the sense that the ordering of the judgments does not have any influence on the aggregation result, as long as the numerical values of the judgments do not change. In many decision problems, however, this symmetry does not exist. Usually, the decision maker considers some of the decision criteria to be more important than others. These criteria must have more influence on the result of the decision making than other criteria. This means that many of the decision functions that have been considered in this chapter cannot be used directly in the decision making process. They must be modified in order to account for the influence of different importance of the criteria.

The importance of decision criteria is modeled by the introduction of additional parameters, called *weight factors*, to the decision model. A weight factor $w_j \geq 0, j = 1, \ldots, n$ is a number that indicates the (relative) importance that a decision maker assigns to a criterion in relation to other decision criteria. The larger the weight factor, the more important the corresponding criterion is assumed to be for the outcome of the decision problem.

Usually, the weight factors are assumed to be elements of the unit interval, *i.e.*, $w_j \in [0, 1], j = 1, \ldots, n$ and they are normalized such that

$$\sum_{j=1}^{n} w_j = 1. \tag{3.57}$$

When one criterion is more important than other criteria, it should have more influence on the outcome of the decision problem. In other words, it should influence the preference structure of the decision maker more than the remaining criteria. This interpretation of the weight factors leads to a general method for incorporating the weight factors into fuzzy aggregation operators (Kaymak and van Nauta Lemke 1998). In contrast, Keeney and Raiffa (1976) interpret the weight factors as scaling factors for the criteria. In this approach, the weight factor indicates how many units of one criterion can be exchanged for one unit of another criterion. The weight factors are then incorporated in the scaling of the variables, and one need not consider them explicitly at the aggregation step. The decision makers, however, feel distinctly that the importance of the criteria cannot be all equal and they prefer models with different weight factors for different criteria. Thus, the use of an anti-symmetric decision model is appreciated. Partly for this reason, the former interpretation for the weight factors is more appropriate, and that interpretation is used in this book. Moreover, fuzzy sets (representing the decision criteria) are aggregated in fuzzy decision making. These fuzzy sets transform the information regarding the criteria to membership values which are then combined further. Hence, different criteria are made commensurable by transforming them to measurements on a single scale (the membership). In this setting, the importance of criteria is a well-defined concept in terms of the sensitivity of the aggregation step to individual membership values.

There are two main methods for increasing the influence of the important criteria on the decision function.

(1) **Transforming the decision function.** In this approach, the weights are introduced into the decision function in such a way that the contribution of the

more important criteria to the aggregate result is increased. Kaymak and van Nauta Lemke (1998) have shown that there are rigorous ways of achieving this.

(2) **Transforming the operands.** In this approach, the operands of a decision function are first transformed to different values, after which the original symmetric aggregation function is used for combining the transformed values. The modification of the operands should be selected such that the alternatives that score higher in more important decision criteria have more influence on the aggregated result.

Various authors have suggested methods for extending fuzzy aggregation operators to their weighted counterparts. Some methods use the method of transforming the decision function, while others use the method of transforming the operands. The remaining sections of this chapter describe weighted counterparts of several frequently used aggregation operators in fuzzy decision making. These operators are also used in the subsequent chapters of this book. A derivation of these operators is not given below. The interested reader is referred to (Kaymak and van Nauta Lemke 1998, Yager 1978, Dubois and Prade 1985, Yager and Filev 1994) for their derivation.

3.5.1 *Weighted counterparts of t-norms*

An important class of fuzzy t-norms is the class of Archimedean t-norms.

Definition 3.10 A t-norm T is said to be *Archimedean* if it satisfies

$$T(\mu_1, \mu_1) < \mu_1. \tag{3.58}$$

Similarly, a t-conorm S is said to be Archimedean if it satisfies

$$S(\mu_1, \mu_1) > \mu_1. \tag{3.59}$$

Kaymak, van Nauta Lemke and den Boer (1998) propose a general framework for finding the weighted counterparts of Archimedean t-norms by using sensitivity analysis on decision functions. This approach transforms the decision function according to a well-defined rule, and it has been shown that it satisfies general conditions that can be expected from a weighted aggregation operator. This sensitivity-based method leads to the following weighted counterparts of the t-norms for $w_i \in [0, 1]$.

The weighted counterpart of the product operator is given by

$$T_p^{\mathbf{w}}(\mu_1, \mu_2) = \mu_1^{w_1} \cdot \mu_2^{w_2}. \tag{3.60}$$

In Eq. (3.60), the notation $T_p^{\mathbf{w}}(\mu_1, \ldots, \mu_m)$ is short for the multivariable function $T_p(w_1, \ldots, w_m, \mu_1, \ldots, \mu_m)$.

The weighted counterpart of the Łukasiewicz t-norm is

$$T_L^{\mathbf{w}}(\mu_1, \mu_2) = \max(0, 1 - w_1 - w_2 + w_1\mu_1 + w_2\mu_2), \tag{3.61}$$

and the weighted counterpart of the Hamacher product is given by

$$T_H^{\mathbf{w}}(\mu_1, \mu_2) = \frac{\mu_1 \mu_2}{w_1\mu_2 + w_2\mu_1 + \mu_1\mu_2(1 - w_1 - w_2)}. \tag{3.62}$$

Parametric t-norms can also be generalized to their weighted counterparts. We give several examples below.

With $\mu_1, \mu_2 \in (0, 1)$, the Dombi class of parametric t-norms is given by

$$T_D(\mu_1, \mu_2) = \frac{1}{1 + \sqrt[\gamma]{\left(\frac{1-\mu_1}{\mu_1}\right)^\gamma + \left(\frac{1-\mu_2}{\mu_2}\right)^\gamma}}, \quad \gamma > 0. \tag{3.63}$$

The weighted version of the Dombi t-norm for $w_1, w_2 \geq 0$ is given by

$$T_D^{\mathbf{w}}(\mu_1, \mu_2) = \frac{1}{1 + \sqrt[\gamma]{w_1\left(\frac{1-\mu_1}{\mu_1}\right)^\gamma + w_2\left(\frac{1-\mu_2}{\mu_2}\right)^\gamma}}, \quad \gamma > 0. \tag{3.64}$$

With $\mu_1, \mu_2 \in (0, 1)$, the Yager class of parametric t-norms is given by (see also Eq. (3.6))

$$T_Y(\mu_1, \mu_2) = \max(0, 1 - [(1 - \mu_1)^\gamma + (1 - \mu_2)^\gamma]^{1/\gamma}), \quad \gamma > 0.$$

The weighted version of the Yager t-norm for $w_1, w_2 \geq 0$ is then given by

$$T_Y^{\mathbf{w}}(\mu_1, \mu_2) = \max(0, 1 - \sqrt[\gamma]{w_1(1 - \mu_1)^\gamma + w_2(1 - \mu_2)^\gamma}), \quad \gamma > 0. \tag{3.65}$$

For $\mu_1, \mu_2 \in (0, 1)$, the Schweizer and Skalar 2 class of parametric t-norms is given by

$$T_S(\mu_1, \mu_2) = \sqrt[\gamma]{\max(0, \mu_1^\gamma + \mu_2^\gamma - 1)}, \quad \gamma > 0. \tag{3.66}$$

The weighted version of the Schweizer and Skalar 2 t-norm for $w_1, w_2 \geq 0$ is then given by

$$T_S^{\mathbf{w}}(\mu_1, \mu_2) = \sqrt[\gamma]{\max(0, w_1\mu_1^\gamma + w_2\mu_2^\gamma + 1 - w_1 - w_2)}, \quad \gamma > 0. \tag{3.67}$$

For the weighted extension of the idempotent t-norm (*i.e.*, the minimum operator), the method of transforming the operands from Sec. 3.5 is suitable. Yager

(1978) has proposed the following decision function for aggregating unequally weighted criteria,

$$D_{\mathrm{Y}}^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \bigwedge_{j=1}^{n} \mu_j^{w_j} \tag{3.68}$$

with $w_j \in [0, 1]$ as the weight factors. In Eq. (3.68), the membership values are raised to the corresponding weight factors as powers. The reasoning behind Eq. (3.68) is that when the weight factors are small, the transformed membership will have a larger value (i.e., closer to 1) and it will thus be less likely to constrain the aggregation owing to the minimum operator (possibly, there are other values in more important criteria that are not close to 1, which influences the result when using the minimum operator). Hence, the less important criteria will have less influence on the aggregation result.

Yager (1984) generalizes this result to a class of decision functions which have the form

$$D^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = T[I(w_1, \mu_1), I(w_2, \mu_2), \ldots, I(w_n, \mu_n)], \tag{3.69}$$

where T is a t-norm and I is a function of two variables for transforming the judgments. The power raising

$$I(w_j, \mu_j) = \mu_j^{w_j}, \tag{3.70}$$

is a commonly applied transformation, while the minimum operator (non-interactive AND) and the product operator (interactive AND) are usually applied as the t-norm. After the generalization of the t-norms, the extension of the t-conorm aggregation can be obtained by using the duality relation Eq. (3.13).

Another transformation function has been proposed by Dubois and Prade (1986), which leads to weighted minimum (and maximum) operators that can also be applied in the setting of the possibility theory. The weighted minimum is given by

$$D^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \bigwedge_{j=1}^{n} [(1 - w_j) \vee \mu_j]. \tag{3.71}$$

The weighted maximum is again obtained from the duality relation Eq. (3.13).

Yager and Filev (1994) give a generalization of Eq. (3.71) for t-norm aggregation. Noting that 1 is the identity element for t-norms, Yager and Filev (1994) require that an operand whose importance is zero should be transformed to the identity element for the aggregation. The weighted decision function then becomes

$$D^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = T(S(\mu_1, \neg w_1), S(\mu_2, \neg w_2), \ldots, S(\mu_n, \neg w_n)), \tag{3.72}$$

where $S(\cdot)$ denotes a t-conorm, and $\neg$ is a negation operator. Note that when w_j is zero, $S(w_j, \mu_j)$, $j = 1, \ldots, n$ equals 1 (due to the t-conorm), which is the identity element for the t-norms. An example of a weighted decision function thus obtained is

$$D^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \prod_{j=1}^{n} (1 - w_j + w_j \mu_j). \tag{3.73}$$

3.5.2 *Weighted counterparts of t-conorms*

Weighted counterparts of t-conorms extend the t-conorm operators to weighted aggregation. They have not been investigated widely in the literature for two main reasons.

(i) Conjunctive and compensatory aggregation are used much more than disjunctive aggregation. Therefore, the incentive to study weighted disjunctive behavior is small.
(ii) The duality relation Eq. (3.13) between t-norms and t-conorms should also hold for the weighted operators. Hence, the weighted counterparts of t-conorms can be obtained in a straightforward manner from the weighted counterparts of the t-norms.

We present in this section weighted counterparts of several t-conorms obtained by using the duality relation Eq. (3.13). Indeed, the generalization of t-conorms to their weighted counterparts is completely analogous to the generalization of the t-norms, when the duality relation Eq. (3.13) is assumed to hold.

The weighted counterpart of the algebraic sum becomes

$$S_p^{\mathbf{w}}(\mu_1, \mu_2) = 1 - (1 - \mu_1)^{w_1} (1 - \mu_2)^{w_2}, \tag{3.74}$$

and the weighted counterpart of the Hamacher sum becomes

$$S_H^{\mathbf{w}}(\mu_1, \mu_2) = \frac{w_1\mu_1 + w_2\mu_2 - (w_1 + w_2)\mu_1\mu_2}{1 + (w_1 - 1)\mu_1 + (w_2 - 1)\mu_2 + (1 - w_1 - w_2)\mu_1\mu_2}, \tag{3.75}$$

while the weighted counterpart of the general weighted decision function Eq. (3.72) is given by

$$D^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = S(T(\mu_1, w_1), T(\mu_2, w_2), \ldots, T(\mu_n, w_n)). \tag{3.76}$$

Note that the weighted counterparts of the operators reduce to the non-weighted operators as expected, when the weight factors are equal, with $w_j = 1$, $j = 1, \ldots, n$.

3.5.3 *Weighted averaging operators*

The decision functions that have been generalized to the weighted case so far have not imposed any conditions on the weight factors, except that they should be non-negative. An additional constraint is needed for the generalization of the averaging operators such that the weight factors are normalized according to

$$\sum_{j=1}^{n} w_j = 1, \quad w_j \in [0,1],\ j = 1, \ldots, n. \tag{3.77}$$

The class of decomposable averaging operators is the most important class of operators that impose Eq. (3.77) as a necessary condition on the weight factors. The weighted form of Eq. (3.23) is then given by

$$M_f^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = f^{-1}\left\{\sum_{j=1}^{n} w_j f(\mu_j)\right\}, \tag{3.78}$$

for $\mu_j \in (0,1), j = 1, \ldots, n$ where the weight factors w_j satisfy Eq. (3.77). For the generalized mean operator in Eq. (3.24), one obtains

$$M_\gamma^{\mathbf{w}}(\mu_1, \ldots, \mu_n) = \left\{\sum_{j=1}^{n} w_j \mu_j^\gamma\right\}^{1/\gamma}, \tag{3.79}$$

as the weighted extension. Note that this extension follows the 'transforming the decision function' method. It is accepted as the most suitable weighted extension of the generalized averaging operator (Kaymak and van Nauta Lemke 1998).

3.6 Summary and concluding remarks

Many aggregation operators are available from the fuzzy set theory that can be used as decision functions in fuzzy decision making. The properties of these aggregation operators are well known, and they can be used to model different types of decision behavior. The most important classes of decision functions in fuzzy decision making are t-norms, t-conorms, the averaging operators and the compensatory operators. Other aggregation operators provide additional flexibility, but they have not yet been applied widely.

The decision function takes a central place in fuzzy decision making, as it combines decision goals and constraints and models the decision maker's preferences. The selection of the decision function is then an important step in the decision procedure which requires detailed analysis. Note that the decision function combines the fuzzy sets for different criteria into a multidimensional fuzzy set from which the best decision is determined. This leads to a similarity-based interpretation of the fuzzy multicriteria decision process, where the final multidimensional fuzzy decision set indicates the similarity of the alternatives to the best alternative. This interpretation is used in Chapter 4 for designing nonlinear controllers.

In some decision problems, the decision criteria have unequal importance. In that case, unequal weight factors are used for modeling the differences in the importance of the criteria for decision making. The weight factors can be introduced into the decision model either by a transformation of the decision functions, or by a transformation of the evaluated membership values (operands of the decision function). Weighted extension of t-norms, t-conorms and averaging operators have been given according to both methods.

Chapter 4

Fuzzy Aggregated Membership Control

We have discussed, in previous chapters, the basics of fuzzy decision making according to Bellman and Zadeh's decision model. In this chapter, we investigate the relation between decision making and control engineering further. The relation between the decision problems and the control problems is studied in detail in the remainder of this chapter. The chapter shows how the fuzzy decision making approach can be applied in control systems. The resulting control systems are part of the fuzzy control paradigm, but the design of the controller is different from that of conventional fuzzy controllers.

We show in Sec. 4.1 that there exists a one-to-one relation between control problems and decision problems. An overview of conventional fuzzy control is given in Sec. 4.2. We describe in Sec. 4.3 how nonlinear controllers can be designed in a simple but powerful way by using the fuzzy decision making paradigm. The properties of these controllers are investigated. Examples are given in Sec. 4.4, before Sec. 4.5 concludes the chapter.

4.1 Decision making and control

By using the results and the terminology from Chapter 2, a one-to-one relation between control problems and decision problems can be established as follows.

(1) In control problems, a number of control actions $u \in U$ are available to the controller. The controller must make a selection for the suitable control action out of those available for achieving the specified control objectives. In other words, the control actions correspond to the set of alternatives A.
(2) There are a set of conditions which cannot be influenced by the control system. These are typically the disturbances that act on the system from the environment. They correspond to the set Θ in decision making.
(3) The control actions and the disturbances act on the controlled system, and lead

to several consequences, such as new states, and error signals. The system description that maps its manipulated variables and the disturbances to the outputs corresponds to the mapping κ in decision making.

(4) The consequences of the control actions are evaluated within a time window by using various criteria, which are typically norms of various signals or quantities derived from the consequences, such as the squared sum of various error signals. The consequences thus correspond to the set Ξ of consequences in decision making, and they are evaluated using several criteria, just like in the multicriteria approach to decision making.

(5) The evaluations for the consequences of the control actions are combined by an objective function. The optimal control action is determined by minimization (or maximization) of the objective function. The objective function corresponds to the decision function D in decision making.

Due to the close relation between decision making and control problems, it is not surprising that decision making methods can be used in control engineering for controller design and implementation. Control engineering studies dynamic systems and develops methods for designing controllers such that the overall interaction between the dynamic system and the controller results in a behavior acceptable to the control engineer. In many cases, the controller implements a mapping tuned for the controlled system, so that the required behavior is obtained, or it is approximated as much as possible in the judgment of the control engineer. Since a decision function also implements a mapping from the decision criteria to the decision outcome, the selection and design of decision functions could also be used to determine the desired controller mapping.

Traditionally, control engineering has been concerned with the development of methods such that a required mapping for the controller is found. Off-line state-space methods (*e.g.*pole placement), frequency domain analysis, and heuristic tuning of controller parameters are examples of techniques that help the control engineer system during the design of a controller. Although the first applications of fuzzy control were motivated by the implementation of human operators' control strategy in an algorithmic form (Holmblad and Østergaard 1982), the attention has shifted towards the specification of a static mapping from the input variables of the fuzzy system to its output variables (Driankov et al. 1993). This static mapping is often called the *control surface.* Almost all fuzzy logic tools available on the market have functionality to view the control surface (or a part of it projected on a two or three dimensional subspace) in three dimensional graphs.

The tuning of a fuzzy controller shapes the control surface such that the nonlinear control law it represents leads to the desired controller performance. Due to the interaction between various parts of a conventional fuzzy controller, the control surface often exhibits (additional) nonlinearity that the control engineer would not specify directly. The influence of such nonlinearity on the control system is often unknown. However, it is desirable to have direct control on this nonlinearity, since it specifies the controller's behavior, and therefore influences its performance. As explained in Sec. 4.3, the fuzzy decision making paradigm could be used for achieving this, where a new type of fuzzy controllers called fuzzy aggregated membership (FAME) controllers allows the control engineer to influence and shape the nonlinearity of the control surface directly.

The relation between decision making and control engineering is more apparent in model-based control. Many powerful methods for controller design are based on estimating a system's behavior by using its model. Since a controller can be seen as a 'decision maker' that must decide upon the best course of action given the information regarding

- the controlled system,
- its environment,
- the control goals, and
- the control constraints,

fuzzy decision making can be applied to control when a model of the controlled process is available. In this approach, the controller uses the model of the system to estimate the consequences of various control actions, and can decide upon a control action to be taken, based on the available information. Therefore, the decision making approach essentially fulfills the role of determining 'an optimal' control action. In contrast to conventional optimization, however, the fuzzy decision making approach allows the use of linguistic goals, soft constraints and flexible preferences. This introduces additional flexibility to the controller, as the designer now has more possibilities for specifying a suitable objective function to optimize. Fuzzy decision theory is used for translating the general objectives of the designer into a mathematical function that the controller uses, as discussed in later chapters.

4.2 Conventional fuzzy controllers

A fuzzy controller implements a mapping between its inputs and its outputs. The controller inputs consist of process variables such as process outputs, errors and measured or reconstructed states. The core of the fuzzy controller implements a static mapping between its antecedent and the consequent variables. Additional dynamics are usually introduced by filtering the input as shown in Fig. 4.1. One of the most commonly used fuzzy controllers is the so-called Mamdani type controller, which is described in many publications about fuzzy control (Driankov et al. 1993, Lee 1990a, Lee 1990b, Yager and Filev 1994). Here, a summary is given of the most salient points.

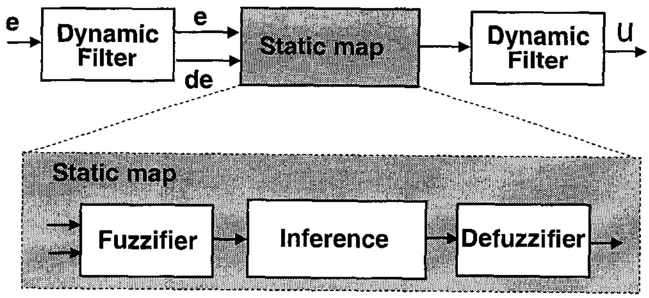


Fig. 4.1 General structure of a fuzzy controller.

4.2.1 *Basic elements of a fuzzy controller*

Figure 4.2 shows the basic elements of a (Mamdani–type) fuzzy controller. The control protocol is stored in the form of **if–then** rules in a rule base, which is a part of the knowledge base. Assume without loss of generality that the controller has n inputs $x_1, \ldots, x_n$, and one output u. The if–then rules are written as

$$R^k\text{:}\ \ \textbf{If}\ x_1 \text{ is } A_1^k \textbf{ and } x_2 \text{ is } A_2^k \textbf{ and } \ldots \textbf{ and } x_n \text{ is } A_n^k \\ \textbf{then}\ u \text{ is } B^k, \tag{4.1}$$

where A_j^k and B^k denote membership functions, and $R^k, k = 1, \ldots, K$ is the kth rule in the rule base. The if–part of the rule is often called the rule antecedent, while the then–part is called the rule consequent. The fuzzy rules describe associations between fuzzy regions in the product space of antecedent variables X and fuzzy regions in the consequent space U, as shown in Fig. 4.3. One can also say that the fuzzy region in the antecedent space is mapped to a fuzzy region in the consequent space by the mapping f. The membership functions that partition the antecedent and the consequent spaces are also part of the knowledge base.

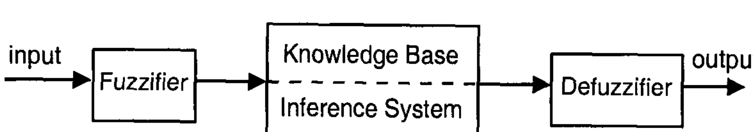


Fig. 4.2 Basic elements of Mamdani–type fuzzy controller.

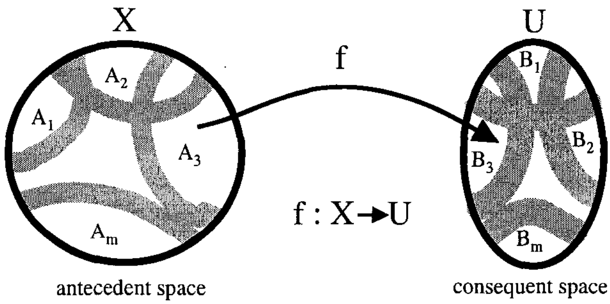


Fig. 4.3 Fuzzy rules associate fuzzy regions in the antecedent space with fuzzy regions in the consequent space.

The rules are based on qualitative knowledge, while the membership functions defining the linguistic terms provide a smooth interface to the numerical process variables and the set-points. The linguistic terms are defined in the fuzzifier, which determines the membership degrees of the controller input values in the antecedent fuzzy sets. The inference mechanism combines this information with the knowledge stored in the rules and determines what the output of the rule-based system should be, as explained in Sec. 4.2.2. In general, this output is again a fuzzy set. For control purposes, a crisp control signal is often required. The defuzzifier calculates the value of this crisp signal from the fuzzy controller outputs.

4.2.2 *Fuzzy inference mechanism*

The inference mechanism of a fuzzy controller combines its inputs with the rules stored in the rule base to determine the outputs of the fuzzy controller. The main steps in the inference mechanism can be summarized as follows (see also Sec. 5.2.1).

(1) **Fuzzy relation.** Each fuzzy rule establishes a relation between a fuzzy region in the antecedent product space and the consequent space. The fuzzy regions are described by membership functions that are defined for the antecedent and consequent variables. The membership functions are representations of the linguistic terms that are used for defining the rules. The fuzzy relation that the rule describes is often interpreted as an implication operation, although the operators used are not strictly fuzzy implication operators. The most commonly used operation for representing the fuzzy relation is Mamdani's min operator. A fuzzy rule with n antecedent variables and one consequent vari-

able can then be represented by the membership function

$$\mu_{R^k}(\mathbf{x}, u) = \bigwedge_{j=1}^{n} \mu_j^k(x_j) \wedge \mu_u^k(u), \quad k = 1, \ldots, K$$

for $\mathbf{x} \in X_1 \times \ldots \times X_n$ and $u \in U$. The overall fuzzy relation R is found by aggregating individual fuzzy relations described by the fuzzy rules. The aggregation operator for this purpose depends on the operator that is used for representing the fuzzy relation for individual rules. When Mamdani implication (minimum operator) is used, the aggregation of individual rules is found by taking the union of individual rules by using the maximum operator. The total relation is given by the membership function

$$\mu_R(\mathbf{x}, u) = \bigvee_{k=1}^{K} R^k(\mathbf{x}, u).$$

(2) **Inference.** The inference mechanism calculates the fuzzy output of the system given its inputs. In Mamdani systems, the compositional rule of inference is used for this purpose. Given the input A' to the system, the fuzzy output B' is found by composing the input with the total relation that is described by the fuzzy rules, *i.e.*,

$$B' = A' \circ R.$$

The composition operator is often the sup-t composition,

$$\mu_{B'}(u) = \sup_{\mathbf{x} \in X} \mu_{A'}(\mathbf{x}) \circledast R(\mathbf{x}, u),$$

where $\circledast$ denotes a t-norm. Usually, the minimum operator or the (algebraic) product operator is used for the t-norm.

(3) **Defuzzification.** In most applications, a crisp value is required, which implies that the fuzzy output B' must be defuzzified to determine a crisp output u^*. The defuzzification can be seen as an operator that replaces a fuzzy set by a representative value, such as the mode (element with the highest membership) or the center of area. A commonly used defuzzifier is the center of gravity defuzzifier, which is given mathematically by

$$Z_u^{\text{cog}} = \frac{\int_{u \in U} u \mu_{B'}(u) du}{\int_{u \in U} \mu_{B'}(u) du},$$

where Z_u denotes defuzzification over the domain U of the control output.

When the inputs to a fuzzy system are crisp, and the output is defuzzified, the fuzzy system represents a static mapping from the inputs to the outputs. This

Table 4.1 A typical rule base for a fuzzy PD controller.

			Δe		
e	NB	NS	ZE	PS	PB
NB	NB	NB	NS	NS	ZE
NS	NB	NS	$\boxed{\text{NS}}$	ZE	PS
ZE	NS	NS	ZE	PS	PS
PS	NS	ZE	PS	PS	PB
PB	ZE	PS	PS	PB	PB

mapping describes a hypersurface in the product space of inputs and outputs, and it is in general a nonlinear mapping. In case of fuzzy controllers, this hypersurface is also called a control surface. Example 4.1 describes a fuzzy counterpart of a PD-controller and the associated control surface.

Example 4.1 The rule base of a fuzzy counterpart of a linear PD controller with two inputs (error e and error change Δe) and one output – the control action u typically looks as shown in Table 4.1.

Five linguistic terms are used for each variable, (NB – *Negative big*, NS – *Negative small*, ZE – *Zero*, PS – *Positive small* and PB – *Positive big*). The linguistic terms are depicted in Fig. 4.4. Each combination of e and Δe in Table 4.1 defines one rule, *e.g.* the rule for the highlighted (boxed) element in Table 4.1 reads

$$R^8\text{: }\textbf{If } e \text{ is NS } \textbf{and } \Delta e \text{ is ZE } \textbf{then } u \text{ is NS.}$$

Figure 4.5 shows the resulting control surface obtained by plotting the inferred control action u for discretized values of e and Δe. In this example, max-min inference with center-of-gravity defuzzification has been used.

The shaping of the control surface directly influences the dynamics of the resulting controller. In addition to the parameters of the fuzzy controller, such as the shape and location of membership functions, as well as the approximate mapping described by the fuzzy rules and the scaling factors, the selection of the inference mechanism and the defuzzification method influence the control surface. Usually, the selection of the inference mechanism and various operators is directly related to the controlled system, such as the ease of implementation, the available inference operators in a software package and so on. If the performance of the controller is found to be unsatisfactory, controller parameters are varied to improve the performance. It is important to realize, however, that the selection of the inference and defuzzification operators has a profound influence on the control surface.

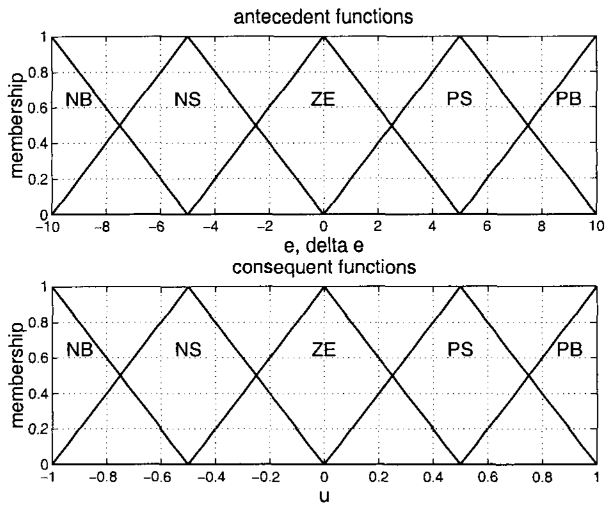


Fig. 4.4 Membership functions for the linguistic terms of a fuzzy PD controller.

Example 4.2 Figure 4.5 depicts the control surface obtained for a fuzzy PD-controller when minimum operator is used for aggregation of antecedent membership functions and max-min composition for the inference. When product operator is used for the aggregation with max-product composition for the inference, the shape of the control surface changes to that shown in Fig. 4.6. Note that the control surface has become smoother. This effect can also be seen by studying the contour lines depicted in Fig. 4.7, which indicate the antecedent conditions that lead to the same control action. The contour lines for the max-product inference are smoother compared to the contour lines obtained from the max-min inference. In addition to a smoothing effect, the product operator can also be used together with the bounded sum aggregation operator (Łukasiewicz t-conorm) to obtain a linear control surface.

4.2.3 *Nonlinearity in fuzzy controllers*

The response of the fuzzy controller is characterized by a nonlinear control surface in the product space of antecedent and consequent variables. The nonlinearity of the control surface should be related to the process characteristics in order to ensure a desirable controller response. The controller's nonlinearity can be influenced in two main ways.

(1) **By changing the rules in the rule base.** The working of the fuzzy controller is described by the fuzzy rules in the rule base. The control strategy of the

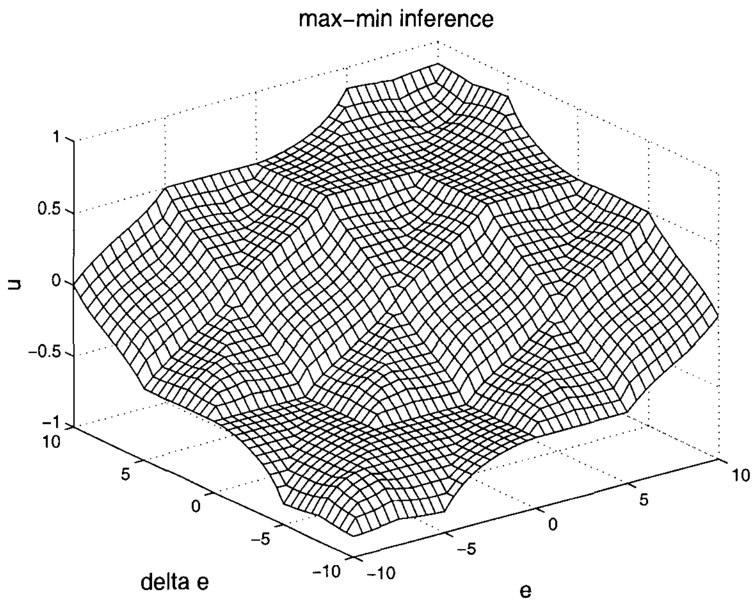


Fig. 4.5 Fuzzy PD control surface with max-min inference.

controller is coded in these rules. By changing the approximate mapping described in the rules, the nonlinear response of the controller can be modified.

(2) **By changing the parameters and the inference mechanism of the fuzzy controller.** The inference mechanism interpolates between the rules. The nature of the interpolation depends on the type of membership functions, their parameters (shape, location, support, core), the inference rule, the selected conjunction operator, the aggregation operator and the defuzzification operator. By modifying these elements, the shape of the nonlinear control surface can be influenced.

Since the fuzzy rules describe the controller's strategy, they should be used to shape the nonlinearity of the control surface, as the nonlinearity of the controller should be a result of the process properties and the control goals. However, the controller may exhibit additional nonlinearity even though the relation described by the rules shows no nonlinearity. As shown in Example 4.1, the fuzzy rule base may be an approximation of a linear PD controller. The control surface, however, is usually nonlinear because of the selected controller parameters, as shown in Fig. 4.5 and Fig. 4.6. This nonlinearity may not always be desirable and additional tuning of the controller is needed to obtain a desired response. The tuning of the Mamdani controller requires the adjustment of a large number of

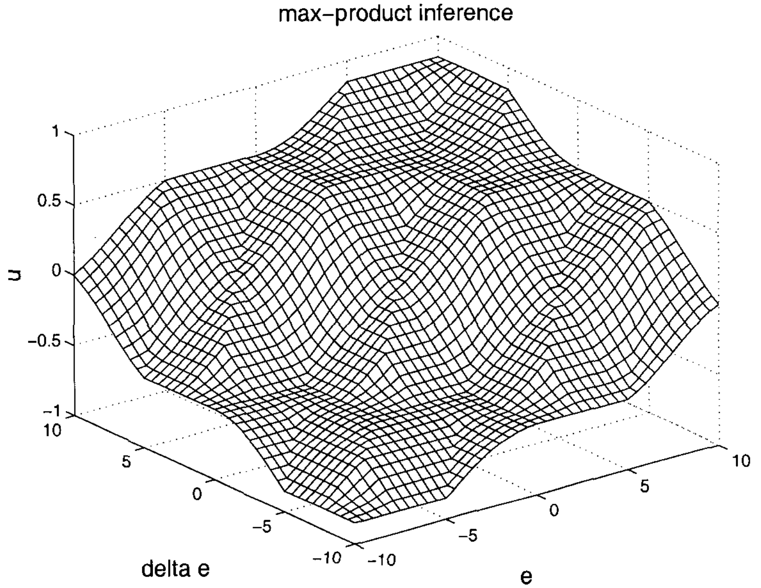
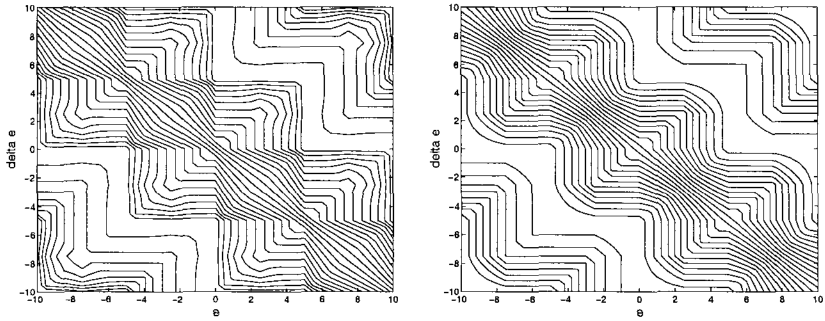


Fig. 4.6 Fuzzy PD control surface with max-product inference and product aggregation of antecedents.

parameters, especially when the number of antecedent variables is high, and it is thus usually a tedious task. Hence, it is desirable to have a controller whose nonlinearity can be explicitly designed and can be tuned with a small number of parameters.

4.3 Nonlinear controllers using decision functions

A decision function which combines n criteria is a mapping $\mathbb{R}^n \longrightarrow \mathbb{R}$. The overall mapping is defined by a combination of one dimensional membership functions which show the degree of satisfaction of a single criterion. In this sense, a fuzzy decision system can be used as a fuzzy controller, since fuzzy controllers are nonlinear systems which implement a nonlinear control law (mapping) between controller inputs and controller outputs. In conventional fuzzy controllers, the nonlinear control law is obtained from the interaction of the approximate nonlinear control behavior described by the fuzzy rules and the way information is combined by the selected inference mechanism including the defuzzification operation. This nonlinear control law is given by the static mapping that the fuzzy system implements between its inputs and its defuzzified outputs. The static map-



(a) Max-min inference *(b) Max-product inference*

Fig. 4.7 Comparison of contour lines for the control surface of two fuzzy PD controllers.

ping is represented by the control surface in the product space of the controller inputs and the outputs. Because of the interaction between the fuzzy rules and the selected inference mechanism, the designer does not have explicit control over the shaping of the control surface, since many of the parameters are determined by considerations not directly related to the properties of the controlled system. This often leads to trial-and-error tuning for obtaining an acceptable controller behavior, which could have been improved if the control surface could have been shaped by the designer explicitly. In order to deal with this problem, this section introduces a type of nonlinear controller, which is based on the aggregation of fuzzy sets using decision functions. The fuzzy sets are defined on the universe of discourse of the individual input variables of the controller, and they indicate the degree of satisfaction of a particular control goal by the input variables. The nonlinear control surface is formed by a suitable aggregation of the fuzzy sets. The nonlinearity of the control surface is determined explicitly by the selection of the membership functions and the aggregation operator. Only one fuzzy set per variable is defined, which reduces the number of controller parameters compared to a conventional fuzzy controller. However, the resulting controller belongs to a more restrictive class of fuzzy controllers.

4.3.1 *Fuzzy aggregated membership controllers*

The fuzzy decision making paradigm that is described in Chapter 2 and Chapter 3 is used in this section to obtain a new type of nonlinear controller called *fuzzy aggregated membership (FAME)* controllers (Kaymak et al. 1996). The decision function defines a decision surface in the product space of the n criteria and the decision: $\zeta_1 \times \cdots \times \zeta_n \times D$. By mapping every point on this decision surface to a unique control action (the value of the control variable), a control surface can be obtained from the decision surface. Because every point of the decision surface corresponds to an alternative with distinct values of membership for the criteria, a control action can be calculated for every combination of input values of the controller.

The proposed controller is obtained in the following way. A particular structure for the controller inputs and the outputs are selected, depending on the specific control problem. For example, if the characteristics of the controlled process require the use of a nonlinear PD controller, the error signal e and its derivative Δe should be selected as the inputs of the controller. Then, one membership function for each variable is defined. The membership function should be defined in such a way that the resulting control surface corresponds to the characteristics of the controlled process (gain factors, *etc.*). This membership function describes a particular property for the variable, such as 'large error signal'. The measured values for the inputs (*e.g.*error and error derivative) can now be converted into a membership value by using the membership functions. In this manner, different variables are brought into a decision space within which various quantities can be combined using a decision function. The decision function, which should reflect the goals of the controller, combines the membership values from different inputs into a single decision value. Finally, the output of the controller is determined by transforming the aggregated membership value into a control value. Figure 4.8 shows the structure of the proposed FAME controller.

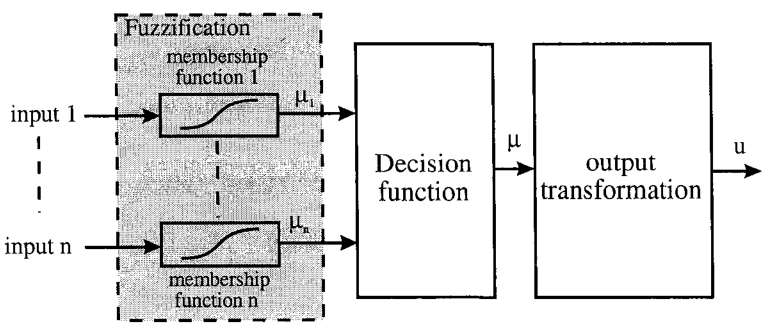


Fig. 4.8 Structure of a fuzzy aggregated membership controller. Reproduced from (Kaymak et al. 1996), ©1996 IEEE.

Consider a FAME controller with n inputs. The steps that the controller takes for calculating the output can be summarized as follows.

(1) Determine the membership values μ_j for each input variable using the mea-

sured values of the variables and the membership functions defined for each variable. Here, the membership function is a mapping $\mathbb{R} \longrightarrow [0,1]$, and it need not have a height of one or a bounded support.

(2) Determine the aggregated membership value μ using the decision function and possibly the weight factors

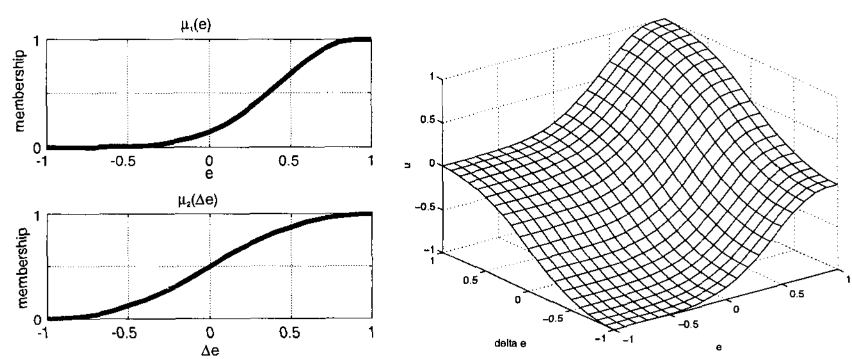
$$\mu = D^{\mathbf{w}}(\mu_1, \ldots, \mu_n). \tag{4.2}$$

(3) Calculate the controller output u using an appropriate function of μ

$$u = h(\mu). \tag{4.3}$$

This transformation is typically a scaling of the closed interval $[0,1]$ to the interval of operating values using a linear mapping.

Example 4.3 Figure 4.9 depicts a nonlinear PD controller using the proposed scheme. The controller has as inputs the error signal e and its derivative Δe. The membership functions that are defined for the two inputs and the resulting control surface of the controller are shown in the figure. The aggregation operator that is used is the arithmetic mean. Note that a linear PD controller would have been obtained if the membership functions for the inputs had been trapezoidal.



(a) Membership functions *(b) Control surface*

Fig. 4.9 A nonlinear PD-controller implemented using arithmetic mean as the fuzzy decision function. (a) Input membership functions. (b) Control surface of the controller. Reproduced from (Kaymak et al. 1996), ©1996 IEEE.

4.3.2 Decomposability of control surface

A disadvantage of the proposed scheme is that the decision surface of the controller can consist only of 'decomposable' functions. Therefore, an arbitrary nonlinear mapping cannot be obtained using the proposed scheme. The function approximation capability of the FAME controllers is considered in more detail in Sec. 4.3.3 and Sec. 4.3.4. In this section, the decomposability of the control surface is discussed. Decomposability means in this context that the final n-dimensional control surface can be found by combining n functions of one variable by using a fuzzy aggregation operator. The following relations are then established. Suppose that the controller has n inputs $x_1, \ldots, x_n$ and one output u. A membership function defined on input x_j, $j = 1, \ldots, n$ is a mapping

$$\mu_j(x_j) : \mathbb{R} \longrightarrow [0, 1],$$

and denotes how much input x_j satisfies the jth fuzzy criterion. For n inputs with (possibly) different weights w_j, the aggregation function is a mapping

$$D^{\mathrm{w}}(\mu_1(x_1), \ldots, \mu_n(x_n)) : [0, 1]^n \longrightarrow [0, 1], \tag{4.4}$$

where D^{w} is increasing in its operands. The final output is found by a third mapping

$$h(D^{\mathrm{w}}) : [0, 1] \longrightarrow \mathbb{R}. \tag{4.5}$$

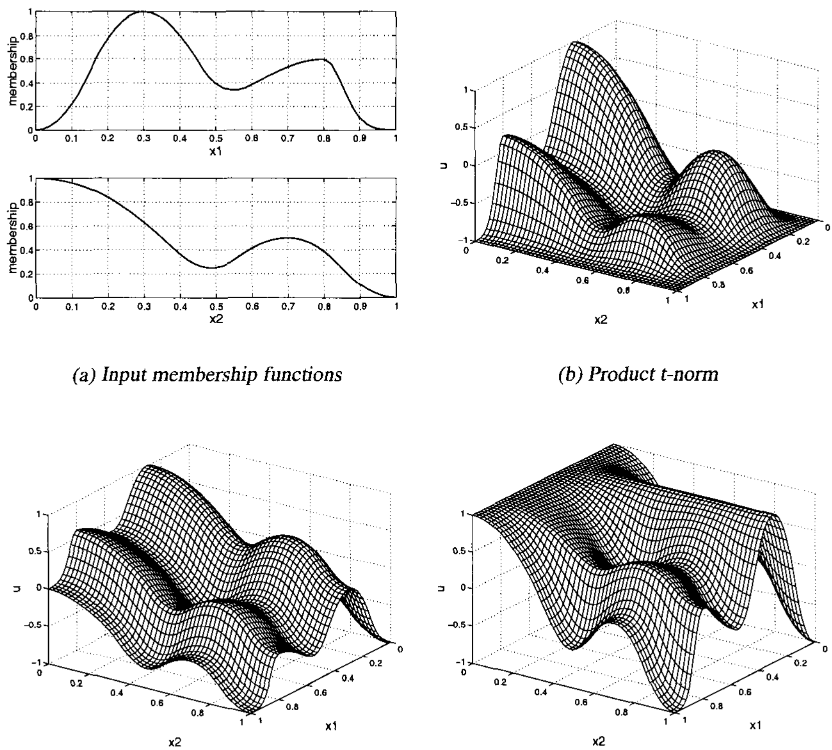
Typically, Eq. (4.5) is a linear mapping, so that it scales the closed interval $[0, 1]$ to another closed interval of real numbers. The output of the fuzzy system is thus given by

$$u = h(D^{\mathrm{w}}(\mu_1(x_1), \ldots, \mu_n(x_n))). \tag{4.6}$$

Example 4.4 Allowing a large degree of freedom in the specification of the membership functions $\mu_j(x_j)$ and in the selection of the aggregation operator, a large variety of control surfaces can be generated. Knowledge about the fuzzy aggregation operators and the requirements for the control surface are used to select a required decision function D^{w} and the shape of the membership functions. Suppose that for a fuzzy controller with two inputs, the membership functions are defined as shown in Fig. 4.10a. When the output mapping h is selected such that

$$h(\mu) = 2(\mu - 0.5),$$

where μ is given by Eq. (4.4), the resulting control surface looks as shown in Fig. 4.10b for the product t-norm. The control surface for the arithmetic mean is shown in Fig. 4.10c, while the control surface for the algebraic sum is shown in Fig. 4.10d. Note the large degree of influence the aggregation operator has on the shape of the control surface, leading to different control surfaces from the same set of membership functions.



(a) Input membership functions

(b) Product t-norm

(c) Arithmetic mean

(d) Algebraic sum

Fig. 4.10 Membership functions and control surfaces for a FAME controller with two inputs and various aggregation methods.

An important class of controllers in control engineering possesses a decomposable form. Consider a nonlinear PID controller given by

$$u(t) = K_1(x_1)e(t) + K_2(x_2)\frac{de(t)}{dt} + K_3(x_3)\int_0^t e(t)dt, \tag{4.7}$$

where u is the controller output, and the controller 'constants' K_1, K_2 and K_3 are functions of the variables x_1, x_2 and x_3, respectively. Assume in the following without loss of generality that the controller output is given by

$$u(t) = K_1(e)e(t) + K_2\left(\frac{de}{dt}\right)\frac{de(t)}{dt} + K_3\left(\int_0^t edt\right)\int_0^t e(t)dt \quad (4.8)$$

$$= K_1(e)e(t) + K_2(e')e' + K_3(E)E, \quad (4.9)$$

where $e(t) = r(t) - y(t)$ is the error signal between the desired output $r(t)$ and the system output $y(t)$, $e'(t)$ is its time derivative and $E(t)$ is its time integral. In Eq. (4.8), K_1, K_2 and K_3 change with time as they are also functions of the error signal. The time-invariant PID controller is obtained when K_1, K_2 and K_3 are constants. Although the PID controller according to Eq. (4.8) does not put bounds on the error signal, in practice this signal and its derivatives are bounded. Hence, it is assumed that e, e' and E take on values from a closed interval. Note that the integrated signal E is also kept bounded in order to avoid wind-up problems (Åström and Hägglund 1995).

The nonlinear PID controller from Eq. (4.8) can be designed by using the nonlinear controller design method described above. Let h be an affine mapping so that the aggregated membership is given by

$$\mu = h^{-1}(u) = \frac{u-b}{c}, \quad (4.10)$$

where b and c are constant. Let the aggregation operator for the FAME controller be the arithmetic mean. Further, suppose that the membership functions $\mu_1(e)$, $\mu_2(e')$ and $\mu_3(E)$ are given where the membership functions are mappings defined as $\mu_j : \mathbb{R} \longrightarrow [0, 1]$. The aggregated value is then given by

$$\mu = \frac{\mu_1(e) + \mu_2(e') + \mu_3(E)}{3}. \quad (4.11)$$

Since $u = c\mu + b$, one finds that

$$u = \frac{c}{3}(\mu_1(e) + \mu_2(e') + \mu_3(E)) + b. \quad (4.12)$$

Comparing Eq. (4.9) with Eq. (4.12), one sees that they can be made equivalent by selecting the membership functions in a special way. One way of achieving the equivalence is to divide Eq. (4.12) into three parts according to $u = u_1 + u_2 + u_3$ with

$$\begin{aligned} u_1 &= \frac{c}{3}\mu_1(e) + \frac{b}{3}, \\ u_2 &= \frac{c}{3}\mu_2(e') + \frac{b}{3}, \\ u_3 &= \frac{c}{3}\mu_3(E) + \frac{b}{3}. \end{aligned} \quad (4.13)$$

Provided that the membership functions are chosen such that

$$\mu_1(e) = \frac{3}{c}K_1(e)e - \frac{b}{c},$$
$$\mu_2(e') = \frac{3}{c}K_2(e')e' - \frac{b}{c}, \quad \text{and} \tag{4.14}$$
$$\mu_3(E) = \frac{3}{c}K_3(E) - \frac{b}{c},$$

the nonlinear FAME controller becomes equivalent to a nonlinear PID controller. Note that the gradient of the membership functions is proportional to the corresponding PID constants. Hence, a large gradient for the membership function $\mu_1(e)$ corresponds to a large proportional action, and so on. By selecting the form of the membership functions, the controller can be tuned for a particular system.

Within the working region of the PID controller, a linear controller is obtained when the membership functions μ_1, μ_2 and μ_3 are linear functions (*e.g.*triangular or trapezoidal membership functions). Suppose that the membership function μ_1 is given by

$$\mu_1(e) = \frac{e + k_1}{2k_1}. \tag{4.15}$$

Figure 4.11 depicts $\mu_1(e)$ graphically. The proportional action of the resulting linear PID controller can then be tuned by varying the value of k_1. A similar observation can also be made for the derivative and the integral actions. *Increasing* the value of k_1 *decreases* the proportional action, and vice versa. Note that k_1 indicates the limits of the working region of the PID controller, and it corresponds to the scaling factors of a Mamdani fuzzy controller. Therefore, the scaling factors influence the gain factors of the controller (Åström and Hägglund 1995).

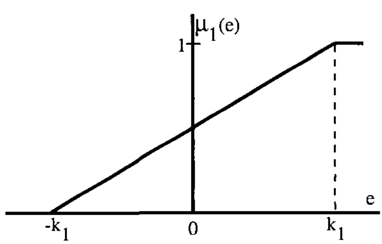


Fig. 4.11 A trapezoidal membership function from which a linear PID controller can be obtained.

The decomposition properties of specific classes of fuzzy systems have recently been studied in literature. The analysis often concentrates on fuzzy systems which use *center–average defuzzifier* (Zeng and Singh 1996a, Zeng and Singh

1996b). In that case, the output of the fuzzy controller can be represented as a sum of basis functions, similar to the radial basis function networks from neural network theory. Then, the aggregation of fuzzy sets leads to the basis functions that span the input product space of the fuzzy system, which are summed with various weights to yield the system output. Zeng and Singh (1996b) show that a multidimensional mapping from an n-dimensional space to a single dimensional space can be decomposed with such a fuzzy system into a collection of simpler fuzzy subsystems that describe the mapping defined on a subspace of the original input space, similar to the composition of a multidimensional mapping from functions of a single variable, as explained above.

Additivity plays an important role in theoretical considerations of decomposability. For the type of decomposability considered in this book, where a nonlinear combination of basis functions is considered, the analysis is more complicated. Instead of the additivity, one must then consider additive generators of the aggregation functions. A more interesting question for the control engineer, however, is not the mathematical properties of such systems, but it is the methods for synthesizing them from some observation data. A heuristic method proposed below for doing this is based on studying the projections of the available data points onto the variable axes, and determining the general shape of membership functions from these projections. Then, the type of fuzzy aggregation that leads to the desired output is considered. The algorithm, which can also be used to design FAME controllers, can be summarized as follows.

(1) Study projections of the available data onto a single input variable. Take as a membership function for that variable either the upper envelope or the lower envelope. If necessary, some curve fitting can be made at this stage. The upper or the lower envelope are used instead of average values in order to obtain a mapping that reaches the extremes in the data.
(2) Use a linear transformation for the output. This implies that the aggregated membership value is translated and scaled for obtaining the output value. This is a mapping from the unit interval to the full range in which the values of the output variable are found.
(3) Using the mathematical properties of fuzzy aggregation operators, determine a class of aggregation operators for use. If the values tend to average one another, averaging operators should be used for the aggregation. If one of the variables dominate along the edges of the input space, t-norms or t-conorms can be used.
(4) Having selected a class of aggregation operators (*e.g.*t-norms, averaging operators or t-conorms), select a parametric family of aggregation operators.
(5) Tune the parameters of the aggregation operator, possibly including the

weight factors, which can be used to introduce some asymmetry into the aggregation.

This method is especially useful when the number of inputs is relatively small (up to four inputs). When there are more inputs, it may be more suitable to decompose the input product space into several subspaces of lower dimension. The systems designed from the reduced spaces can then be combined at a higher level (either additively or otherwise), resulting in a hierarchy of decision functions.

4.3.3 *Relation to rule-based systems*

One of the disadvantages of the FAME controllers is that the link to rule-based linguistic systems is less apparent. Strictly speaking, FAME controllers are not rule based systems, as they are based on an aggregation of several membership functions according to a formula. Since one of the advantages of fuzzy systems is their transparency revealed by the linguistic rules, an explicit link to the rule-based systems is desirable. Note first that the FAME controllers are not completely non-interpretable, since the membership functions represent a particular property that one of the input variables should satisfy. Depending on the application domain, these membership functions can be given at least the following three interpretations.

(1) **A fuzzy criterion.** The membership value represents a particular property that the input signal should attain, *e.g.* small output error.
(2) **A gradual rule.** (Dubois and Prade 1992) The membership function denotes how elements of the input domain relate to the elements of the output domain, given a relation between two typical elements in the input and output domains. For example, 'the more positive the error, the more positive the control action'.
(3) **Nonlinear gain factor.** It has been explained above that a nonlinear PID controller can be implemented using the method based on decision functions. The membership functions are then related to the gain factors of the controller.

Despite these interpretations and the transparency they introduce, a description of the fuzzy system in terms of fuzzy rules can be sometimes more desirable in order to identify the combination of inputs in which a particular condition described by the rule is valid. Note that given a rule-based fuzzy system, one may generate the mapping it represents, provided that the inference and the defuzzification methods are known. A fuzzy system based on decision functions can then be designed, for example, by using the heuristic method described above. The resulting fuzzy system approximates the mapping described by the rule based fuzzy system. The inverse mapping from decision function based fuzzy systems to rule based fuzzy systems is also possible under certain conditions, as explained below.

Let x be the input of a fuzzy system, and let $y = f(x)$ be a mapping that the fuzzy system describes from the input x to the output y. As mentioned above, such a mapping can be interpreted as a nonlinear gain factor in a nonlinear controller based on fuzzy decision functions. Further, assume that $f(x)$ is monotonic and continuous in x. Then, the following fuzzy system can be designed such that the system describes the mapping exactly. Let the input domain be defined by $x \in X = [a, b]$. Due to monotonicity and continuity, the output will also lie in a closed interval, for example given by $y \in Y = [f(a), f(b)]$. The case where $f(x)$ is decreasing so that $Y = [f(b), f(a)]$ is analogous. Divide both the input domain and the output domain into two fuzzy intervals and establish two rules

R^1: **If** x is A^1 **then** y is B^1

R^2: **If** x is A^2 **then** y is B^2.

Further, let A^1 and A^2 be two triangular fuzzy sets such that the sum of membership values $\mu_{A^1}(x)$ and $\mu_{A^2}(x)$ equals 1.0. Figure 4.12 shows the partition of the input domain. The fuzzy sets on the input domain are given by

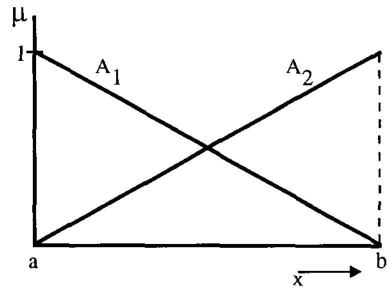


Fig. 4.12 Partition of the input domain for a minimal fuzzy system.

$$\mu_{A^1}(x) = \frac{b-x}{b-a}, \quad \text{and} \quad \mu_{A^2}(x) = \frac{x-a}{b-a}. \tag{4.16}$$

Further, let the implication be given by

$$R^k(x,y) = \begin{cases} 1, \; \mu_{A^k}(x) \le \mu_{B^k}(y), & k = 1,2 \\ 0, \; \text{otherwise.} \end{cases} \tag{4.17}$$

For the total relation described by the rule base, the conjunction of the rules is used, *i.e.*, $R = R^1 \cap R^2$. The question that must now be answered is how to select the output fuzzy set membership functions $\mu_{B^1}(y)$ and $\mu_{B^2}(y)$, so that the fuzzy system describes the mapping $f(x)$ exactly. Let the output membership function be defined using the inverse mapping f^{-1} as

$$\begin{aligned}\mu_{B^1}(y) &= \frac{b - f^{-1}(y)}{b-a}, \quad \text{and}\\ \mu_{B^2}(y) &= \frac{f^{-1}(y) - a}{b-a} = 1 - \mu_{A^1}(x).\end{aligned} \tag{4.18}$$

Since $x = f^{-1}(y)$, Eq. (4.17) and Eq. (4.18) lead to

$$y \geq f(b - \mu_1(x)(b-a)), \tag{4.19}$$

for the output set of rule R^1, and to

$$y \leq f(b - \mu_1(x)(b-a)) \tag{4.20}$$

for the output set of rule R^2. Since the total relation is a conjunction, Eq. (4.19) and Eq. (4.20) give the solution

$$y = f(b - \mu_1(x)(b-a)) = f(x), \tag{4.21}$$

where Eq. (4.16) is also used. Hence, the output of the rule-based system is equal to the final (desired) mapping. Note that this system is a rule-based equivalent of the fuzzy extension of a crisp mapping according to the fuzzy extension principle. Given such a rule based system, any monotonic function of a single variable can be realized. By combining several systems, one can find a rule-based description for any function of the form

$$\sum_{j=1}^{n} w_j f_j(x_j). \tag{4.22}$$

To see how Eq. (4.22) relates to nonlinear controllers based on fuzzy decision functions, suppose that an operator which possesses an additive generator function representation is used for the aggregation. Remember that this class includes many of the commonly used aggregation operators. It is observed from Eq. (4.5) that the overall mapping for the controller with n inputs is given by

$$u = h(D^{\mathbf{w}}(\mu_1(x_1), \ldots, \mu_n(x_n))). \tag{4.23}$$

One obtains, by considering weighted aggregation in additive generator function representation, that

$$u = h\left(\bar{f}^{-1}\left(\sum_{j=1}^{n} w_j \bar{f}(\mu_j(x_j))\right)\right),$$

$$\Rightarrow u = g_0 \left(\sum_{j=1}^{n} w_j g_j(x_j) \right) . \tag{4.24}$$

Comparing Eq. (4.24) and Eq. (4.22) it is seen that the rule-based system can describe the argument of g_0. For linguistic interpretability, a rule-based system can also be designed to match g_0, or it can be used to transform the consequents of the original rule based system to the output domain.

In the analysis above, it is assumed that the one dimensional mappings are monotonic. If this is not the case for the modeled mapping, then the non-monotonic mapping $f(x)$ can be subdivided into a number of monotonic regions which are then combined together. Each monotonic region is modeled with one rule based system, each of which again leads to a minimal fuzzy system. Kosko (1997) has proposed a method based on similar concepts for designing Mamdani systems with a minimal number of rules.

4.3.4 *Function approximation capability*

Since they can only approximate decomposable functions, the FAME controllers cannot approximate an arbitrary nonlinear function. If general approximation capability is required for achieving more complicated dynamic behavior, several FAME controllers must be combined in a hierarchical system with at least two levels in the hierarchy. Zeng and Singh (1996b) have shown that a fuzzy system can be divided into a number of subsystems, each of which describe a mapping that is monotonic in its operands. Each such system can be approximated by a FAME controller as described above, and their union describes an arbitrary non-linear continuous system. In other words, each FAME controller can be interpreted as a rule which describes the system behavior in a particular part of the input product space. The union of the individual rules describes the desired non-linear mapping. The function approximation capability of these systems can also be seen from Eq. (4.24) which shows that a nonlinear controller based on fuzzy decision functions can be compared to a neuron in a neural network (Hunt et al. 1992, Kosko 1992). The nonlinear function g_0 is equivalent to the nonlinearity of a neuron which acts on a summation of its inputs. In the above case, the inputs to the neuron are modified through the functions $g_j(x_j)$. Suppose that K of these elements are combined linearly as shown in Fig. 4.13. The system in Fig. 4.13 describes a mapping from n inputs to p outputs. The g_{jk}, $j = 1, \ldots, n$, $k = 1, \ldots, K$ are the nonlinear functions from Eq. (4.24) obtained for the FAME controller. They are the nonlinear transforms for the input layer of a neural network. The outputs transformed through g_{jk} are summed up weighted by w_{jk}, and the output of each summation is passed to a nonlinear membership function g_{0k}, whose outputs are the outputs of the hidden layer of the neural network. They are summed up weighted by v_{ik}, $i = 1, \ldots, p$, to yield the system outputs (third layer). Cybenko (1989) has shown that such a configuration can approximate any continuous mapping, provided that a sufficient number of nonlinear elements are available. Therefore, several FAME controllers can be used to approximate an arbitrary continuous function.

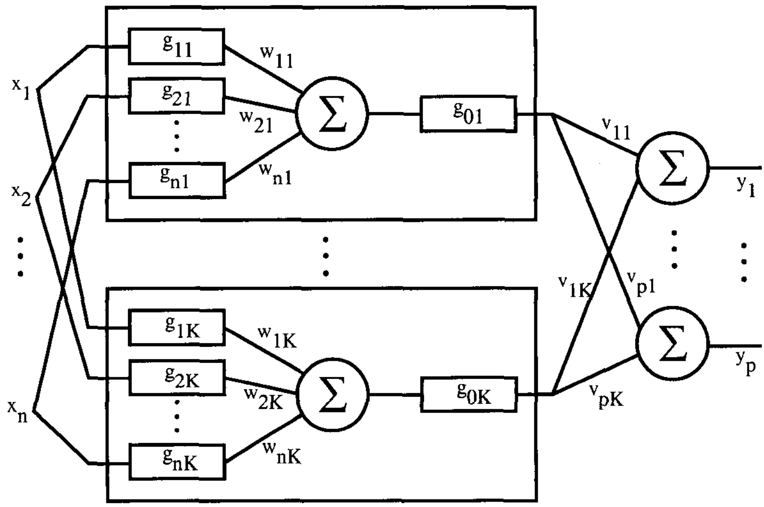


Fig. 4.13 Several FAME controllers can be combined in a feedforward network structure for approximating a general nonlinear continuous function.

4.4 Examples of fuzzy aggregated membership control

In this section, a couple of examples of nonlinear systems based on fuzzy decision functions (FAME controllers) are given. FAME controllers can be used in existing control systems to improve their performance, as discussed in Sec. 4.3.1. In this section, two additional examples are given of the use of FAME controllers in modeling and nonlinear PID control.

4.4.1 *Parameter estimation of nonlinear parity equations in aircraft*

One of the approaches to actuator failure detection and identification in aircraft is the parity space approach, which utilizes the redundancy contained in the static and dynamic relationships amongst the actuator commands and measured outputs

(Patton et al. 1989). Parity equations from linear model-based detection systems are applicable only in one operating point. Schram et al. (1998) have suggested a method whereby the parameters of the parity equations are scheduled between different operating points by using a fuzzy system, so that the failure detection and identification becomes more robust to changing flight conditions. One of the important aspects of their method is modeling the parameter values of the parity equations as a function of the changing flight conditions. The parity equations typically have many parameters, each of which must be modeled separately. Below, the modeling of only one of these parameters is considered. Schram et al. (1998) use a zero-order Takagi–Sugeno system for the modeling. The value of the *gain* parameter is determined from the airplane's air speed and its total mass. The gain is measured for 15 different operating conditions, each of which is represented as a rule in a rule base. Figure 4.14 shows the input membership functions as used by Schram et al. (1998) and the resulting relation between the inputs of the system and the parameter of the parity equation.

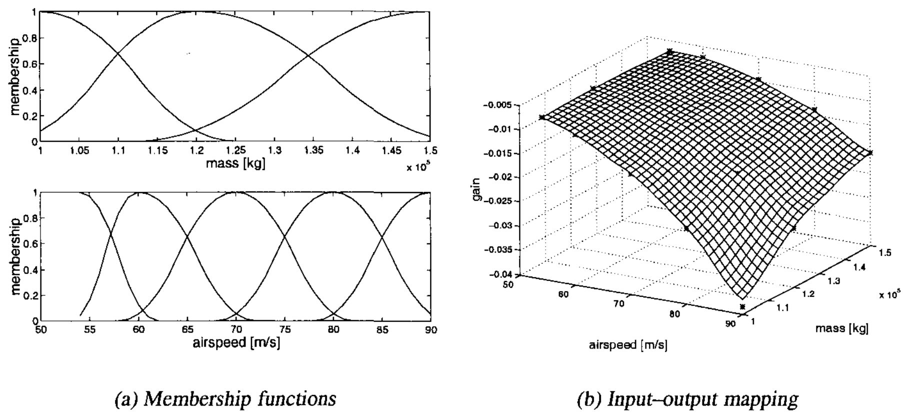
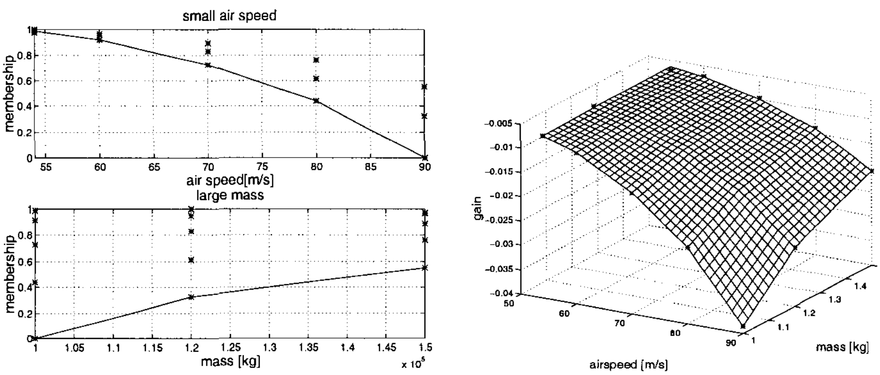


Fig. 4.14 Takagi–Sugeno system used by Schram et al. (1998) for estimating the parity equation gain; (∗) are the measured data points.

The modeling of the gain by using fuzzy decision functions starts by studying the projections of the measured points onto the input variables. The lower envelopes of the projections of the measured data points on to the 'airspeed' and 'mass' axes are chosen as the membership functions as shown in Fig. 4.15a. In this example, the membership functions are not parameterized, but they are derived directly from the measurements with linear interpolation between the measurements. Since the output becomes low only when both the airspeed and the mass are low, a t-conorm operation is used for the aggregation. Hamacher t-conorm (Mizumoto 1989a) in its parametric form has been chosen for this purpose. The parameter γ of the t-conorm is optimized for the best fit to the 15 points, and the aggregated result is scaled to the output domain by using an affine transformation. The optimal value of the parameter is found to be 1.18. Figure 4.15b shows the resulting mapping from the input domain to the output domain. The sum squared error is calculated to be 1.0×10^{-6}, which is lower than 2.7×10^{-5}, the sum squared error obtained by Schram et al. (1998). Note that the number of parameters in the system based on fuzzy decision functions (two membership functions and the t-conorm parameter γ) is smaller than the number of parameters in the Takagi–Sugeno system with 15 rules (8 membership functions). Thus, the system with fuzzy decision functions achieves good approximation while decreasing the number of free parameters in the system. Hence, it is a good modeling tool for this system.



(a) Membership functions (b) Input–output mapping

Fig. 4.15 System based on fuzzy decision functions for estimating the parity equation gain. The input membership functions are aggregated using Hamacher t-conorm; $(*)$ are measured data points.

4.4.2 *Nonlinear PID control of a laboratory propeller setup*

A laboratory propeller setup is shown in Fig. 4.16. It consists of two propellers, one rotating along a horizontal axis and the other one rotating along the vertical axis. The two propellers are coupled together by a rigid rod that can rotate in the horizontal plane (yaw ϕ) and in the vertical plane (pitch θ). The goal of the control system is to bring the rod to a desired yaw angle and pitch angle by using the horizontal and the vertical motors. The horizontal and the vertical motors show a strong coupling.

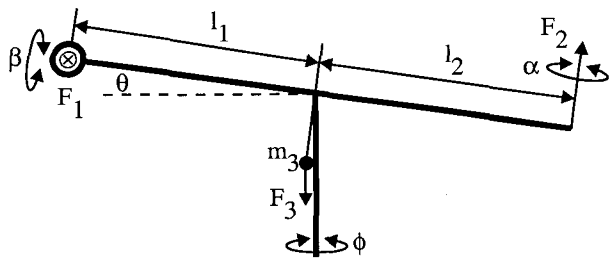


Fig. 4.16 Schematic diagram of a laboratory setup.

The two subsystems can be controlled by two PID controllers. However, the tuning is difficult, especially for the horizontal system, because the overshoot becomes very large if the proportional action is large. If it is small and the derivative action is large, then the response time is very long, *i.e.*, the system response is too slow. Figure 4.17 shows the yaw and pitch responses for a PID controller tuned for the considered response. A consideration of the system indicates that the gain factors should preferably be small when the system is far away from the set point (*i.e.*, when the error and its derivative is large), and it should be large around the steady-state in order to improve the response. A FAME controller is designed to implement this strategy. The nonlinear PID controller analysis presented in Sec. 4.3.1 is used for designing the controller. Since the derivatives of the membership functions are proportional to the gain factors, sine curves are used as the membership functions to model the effect described above. These membership functions are depicted in Fig. 4.18a.

The sine curves are represented in parametric form and the slope of the membership functions are tuned by changing the scaling factors. The three membership functions for the proportional, derivative and integrals actions are combined together with weighted arithmetic mean. A cross-section of the resulting control surface for the PD combination is shown in Fig. 4.18b. Figure 4.17 shows the response of the system, which illustrates the improvement due to the nonlinearity introduced in the PID controller. Note that the computational load of this controller consists only of evaluating three membership functions and a linear combination of them.

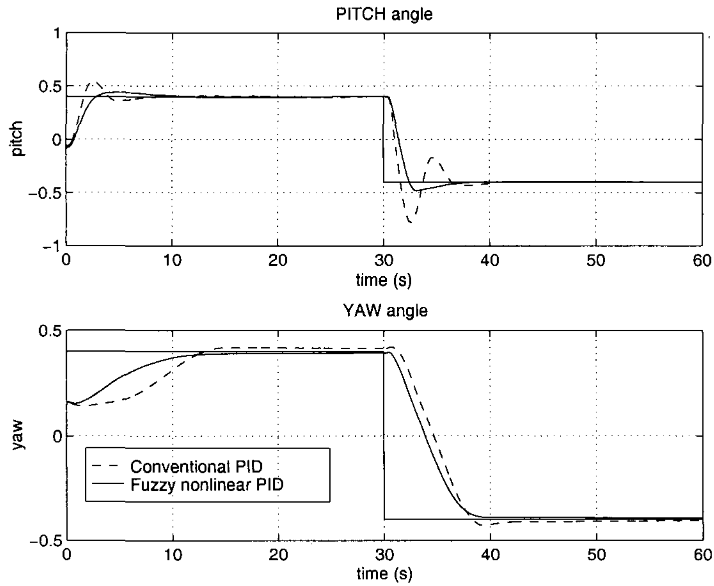
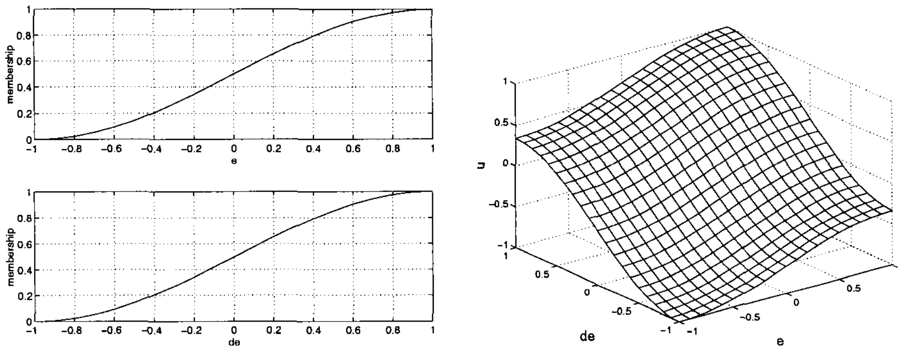


Fig. 4.17 Closed loop response of a conventional PID controller and a nonlinear PID controller based on fuzzy decision functions for the laboratory propeller setup.

4.5 Summary and concluding remarks

This chapter has considered the use of a new type of nonlinear controller that is based on a fuzzy decision making paradigm and fuzzy decision functions. In contrast to fuzzy logic controllers, whose nonlinearity may be the result of a number of factors, some of which are not explicit, the proposed FAME controller allows the specification of its nonlinearity in an explicit way. The controller uses aggregation operators from fuzzy set theory and one membership function per input variable to determine its output. Tuning this controller requires relatively less effort because of the reduced number of parameters and the fact that it is based on aggregation operators whose properties are well-known from the fuzzy set theory. However, the controller is not as general as a fuzzy logic controller, due to the reduced degrees of freedom. In particular, it can only approximate mappings that can be decomposed into a number of single dimensional functions, which can be combined by a fuzzy decision function to compose the original mapping.



(a) Membership functions *(b) PD cross-section*

Fig. 4.18 Nonlinear PID controller based on fuzzy decision functions for the laboratory propeller setup.

Chapter 5

Modeling and Identification

In this chapter, we turn our attention to model-based control. In model-based control, the controller determines the current control actions by using a model of the controlled process. Some of the modeling and identification techniques used in this book, and that are particularly suitable for fuzzy model-based control (FMBC) are presented in this chapter. The goal of the chapter is to present fuzzy modeling and identification techniques in so far they are used for obtaining the fuzzy models used in this book.

The development of a model of the process is essential for fuzzy model-based control. Traditional first-principles models are used, based on a deep knowledge of the nature of the system, and on a suitable mathematical treatment. This type of models is usually known as 'white-box' models or mechanistic models. Some systems are almost linear and can be approximated by a linear model. This is, however, generally not the case, and linear models are derived presenting the system around a working point. This approach has the disadvantage of restricting the control to a certain operating region of the system. When the model is used in a model-based control scheme outside the region around the working point, it usually leads to poor control performance.

If a first-principles approach is not possible, linear identification techniques can be used. The linear system is often represented as a state-space model of the form

$$\begin{aligned} \mathbf{x}(\tau+1) &= \mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{u}(\tau), \\ \mathbf{y}(\tau) &= \mathbf{C}\mathbf{x}(\tau), \end{aligned} \tag{5.1}$$

for $\tau = 0, 1, \ldots$.

For some nonlinear processes, the system's behavior can be described by suitable mathematical laws, leading to a model, which can be used for control purposes, *i.e.*, the model is not too complex and does not involve heavy computational effort. These so-called nonlinear white-box models are highly desirable because they can describe the system not only around a working point, but in the whole range of the system under control. However, many processes are complex and only partly understood. Generally, a good mathematical description of the underlying physics of the system is not possible.

As this white-box approach is laborious, and inefficient for complex and partially known systems, a nonlinear model based on soft computing techniques, *e.g.* fuzzy, neural, or neuro-fuzzy methods, can be used. These methods can represent highly nonlinear processes in an effective way, due to their general function approximation properties. Fuzzy modeling is an attractive modeling technique because it is possible to combine different types of knowledge and data such as first principles, knowledge obtained from linguistic rules describing the system and/or measurements.

Linear models are mainly used in this book as test cases. Nonlinear white-box models are used as the 'real' process in some simulation tests. Fuzzy models are widely used throughout this book, also for real-time implementations. The general formulation of the modeling problem is presented in Sec. 5.1. The basic principles of fuzzy modeling are described in Sec. 5.2. When only data of the system under control is available, fuzzy identification, as presented in Sec. 5.3, can be used. The identification of fuzzy models using product-space fuzzy clustering, which is the type of fuzzy identification used in this book, is briefly described in Sec. 5.4.

5.1 Formulation of the modeling problem

Modeling is a technique that derives a model of the system under control. Discrete nonlinear autonomous open loop models are considered in this book. For the discrete case, the general form of these multiple input multiple output (MIMO) models is given by

$$\begin{aligned} \mathbf{x}(\tau+1) &= \mathbf{h}(\mathbf{x}(\tau), \mathbf{u}(\tau)), \\ \mathbf{y}(\tau) &= \mathbf{g}(\mathbf{x}(\tau)). \end{aligned} \tag{5.2}$$

Let $\mathbf{h}'$ and $\mathbf{g}'$ be the functions exactly describing the system. The objective of modeling is to approximate the functions $\mathbf{h}'$ and $\mathbf{g}'$, by the functions $\mathbf{h}$ and $\mathbf{g}$, respectively, such that the approximations are as close as possible to the functions describing the system.

Suppose now that a black-box model must be identified from input–output data. In this case, the state vector $\mathbf{x}$ can be obtained from the inputs and outputs of the system, joining them in a vector,

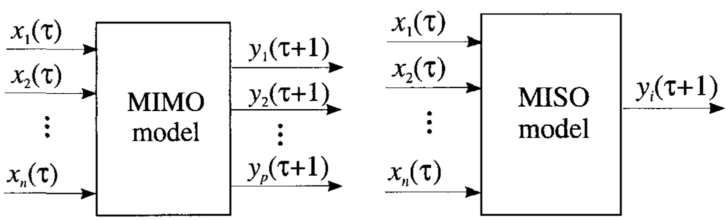
$$\mathbf{x}(\tau) = [y_1(\tau), \dots, y_1(\tau - p_1 + 1), \dots,$$

$$
\begin{aligned}
&y_p(\tau), \ldots, y_p(\tau - p_p + 1), \\
&u_1(\tau), \ldots, u_1(\tau - m_1 + 1), \ldots, \\
&u_m(\tau), \ldots, u_m(\tau - m_m + 1)]^T.
\end{aligned} \tag{5.3}
$$

The parameters $m_1, \ldots, m_m$ are the orders of the inputs $u_1, \ldots, u_m$, and the parameters $p_1, \ldots, p_p$ are the orders of the outputs $y_1, \ldots, y_p$, respectively. Note that the dimension of the state vector is given by $n = \sum_{j=1}^{m} m_j + \sum_{j=1}^{p} p_j$. With this state vector, Eq. (5.2) can be reduced to

$$\hat{\mathbf{y}}(\tau + 1) = \mathbf{f}(\mathbf{x}(\tau)). \tag{5.4}$$

The state variables $\mathbf{x}$ are called the regressor and the predicted outputs $\hat{\mathbf{y}}$ the regressand. The static nonlinear regression in Eq. (5.4) is widely used for the modeling of nonlinear dynamic systems, either in input–output or state-space form. The general MIMO system in Eq. (5.4) is depicted in Fig. 5.1a. This model can



(a) Complete MIMO model.

(b) The ith MISO model of the total MIMO system.

Fig. 5.1 Generic MIMO model and one of its ith MISO component.

be decomposed in a collection of MISO systems, if each MISO system is represented by a MISO Nonlinear Auto Regressive with eXogenous input (NARX) model. Denote with $y_i, i = 1, \ldots, p$, an output considered for a particular MISO NARX model i. This MISO system, shown in Fig. 5.1b, is given by

$$\hat{y}_i(\tau + 1) = f_i(\mathbf{x}(\tau)). \tag{5.5}$$

The coupled MIMO system is given by the collection of all the p MISO systems. Note that the definition of the state vector as in Eq. (5.3) allows that *any* previous input or output of the system can be a state for each particular MISO model i, with $i = 1, \ldots, p$. A collection of the MISO systems, as in Eq. (5.5), defines thus a general MIMO system as in Eq. (5.4).

5.2 Fuzzy modeling

From the modeling techniques based on soft computing, fuzzy modeling is one of the most appealing. In fact, when the process under control is nonlinear, and the system can not be totally described by first principles, but is only partly known, it is advantageous to use fuzzy modeling as a way to combine first principles, knowledge obtained from experts describing the system's behavior and linguistic rules obtained from measurements. This approach is also called *gray-box* modeling. If no a priori knowledge (physical models of parts of the system or linguistic rules) is available, the rules and membership functions can be directly extracted from process measurements, using various techniques, such as fuzzy clustering, neural learning methods or orthogonal least squares (see Guillaume (2001) for an overview). Fuzzy models provide a transparent, gray-box description of the process dynamics that reflects the nature of the process nonlinearity for low-order nonlinear systems (Babuška 1998). Many processes are partly known, where first principles and measurements on the system can be synergistically combined.

The theory of fuzzy sets can be applied to the modeling of systems in different ways. Traditionally, rule-based fuzzy systems are used (Zadeh 1973, Driankov et al. 1993). In computational terms, fuzzy models are flexible mathematical structures that, in analogy with neural networks and radial basis functions, are known to be universal function approximators (Wang 1992, Kosko 1994, Zeng and Singh 1995). In fuzzy modeling, the fuzzy If–Then rules take the following general form,

If *antecedent proposition* **then** *consequent proposition.*

Fuzzy models use 'If–Then' rules and logical connectives to establish relations between the variables defined for the model of the system. The fuzzy sets in the rules serve as an interface amongst qualitative variables in the model, and the input and output numerical variables. The rule-based nature of the model allows for a linguistic description of the knowledge, which is captured in the model. The fuzzy modeling approach has several advantages when compared to other nonlinear modeling techniques, such as neural networks. In general, fuzzy systems can provide a more transparent representation of the system under study and can also give a linguistic interpretation in the form of rules.

Depending on the form of the propositions and on the structure of the rule base, different types of rule-based fuzzy models can be distinguished. Two different types are used in this book.

(1) *Linguistic or Mamdani fuzzy model* (Zadeh 1973, Mamdani 1977), where both the antecedent and consequent are fuzzy propositions (see Sec. 5.2.1).

(2) *Takagi–Sugeno (TS) fuzzy model* (Takagi and Sugeno 1985, Sugeno and Tanaka 1991), where the consequents are crisp functions of the antecedent variables rather than a fuzzy proposition. TS models are described in Sec. 5.2.3.

The singleton fuzzy model, where the consequent is a singleton, can be seen as a particular case of both linguistic and TS fuzzy models, and is presented in Sec. 5.2.2.

5.2.1 *Linguistic fuzzy models*

The linguistic fuzzy model (Zadeh 1973, Mamdani 1977) consists of rules where both the antecedent and the consequent are fuzzy propositions. Linguistic fuzzy models represent static mapping of systems. A general rule of a linguistic or Mamdani fuzzy model is given by

$$R^k\colon\ \textbf{If } \mathbf{x} \text{ is } A^k \textbf{ then } \mathbf{y} \text{ is } B^k, \quad k = 1, 2, \ldots, K, \tag{5.6}$$

where R^k denotes the kth rule and K is the number of rules. The antecedent variable is given by $\mathbf{x} \in X \subset \mathbb{R}^n$ and represents the input of the fuzzy system. Similarly, $\mathbf{y} \in Y \subset \mathbb{R}^p$ is a consequent variable representing the output of the fuzzy system. Note that these symbols are conform to states and outputs of systems presented in Sec. 5.1. A^k and B^k are fuzzy sets described by the membership functions $\mu_{A^k}(\mathbf{x})\colon X \to [0,1]$ and $\mu_{B^k}(\mathbf{y})\colon Y \to [0,1]$, respectively. Fuzzy sets A^k define regions in the antecedent space X, and fuzzy sets B^k define regions in the consequent space Y. The antecedents are usually defined as a combination of simple fuzzy propositions for each x_j, $j = 1, \ldots, n$ of the vector $\mathbf{x}$, instead of using multidimensional fuzzy sets. In a similar way, the consequents can also be divided in simple fuzzy propositions y_i, $i = 1, \ldots, p$. This decomposition has the advantage that the linguistic interpretability of the model increases. For the fuzzy propositions it is usual to attribute linguistic meanings such as 'high temperature', 'small velocity', and so on. As the antecedent and consequent fuzzy sets take on linguistic meanings, they are called linguistic labels of the *linguistic variables*. For instance, if the linguistic variable is 'temperature', several fuzzy sets can be defined for this variable, *e.g.* 'low', 'medium', 'high'. Different fuzzy logic operators, such as conjunction, disjunction and complement can be used to combine the antecedent propositions. The most commonly used form is, however, the conjunctive form, given by

$$\begin{aligned} R^k\colon\ &\textbf{If } x_1 \text{ is } A_1^k \textbf{ and } x_2 \text{ is } A_2^k \textbf{ and } \ldots \textbf{ and } x_n \text{ is } A_n^k \\ &\quad \textbf{then } y_1 \text{ is } B_1^k \textbf{ and } y_2 \text{ is } B_2^k \textbf{ and } \ldots \textbf{ and } y_p \text{ is } B_p^k, \end{aligned} \tag{5.7}$$

where one-dimensional fuzzy sets are defined for each component of the antecedent and consequent vectors. Note that the conjunctive form divides the antecedent space in a lattice of axis-orthogonal hyperboxes.

Given the rules and the known inputs, the inference mechanism derives the outputs of the fuzzy model. The *compositional rule of inference* (Zadeh 1973) performs the fuzzy inference for linguistic models. Each rule in Eq. (5.6) is a fuzzy relation R^k: $X \times Y \to [0,1])$, which is computed by

$$\mu_{R^k}(\mathbf{x},\mathbf{y}) = \mathrm{I}(\mu_{A^k}(\mathbf{x}), \mu_{B^k}(\mathbf{y})), \tag{5.8}$$

where the operator I can be a fuzzy implication or a conjunction operator (t-norm). The entire rule base is represented by combining the K relations R^k of the individual rules into a global relation R. If I is an implication, R is obtained by making a conjunction of all the R^k, and if I is a conjunction operator, R is computed as a disjunction of the individual relations R^k. For a given input 'x is A'', and the relation R, the corresponding output fuzzy set B' is derived by

$$B' = A' \circ R, \tag{5.9}$$

where $\circ$ denotes the sup-t composition (Klir and Yuan 1995). The norm used most in this composition is the minimum t-norm, leading to the following composition

$$\mu_{B'}(\mathbf{y}) = \max_{\mathbf{x}} \min_{\mathbf{x},\mathbf{y}} (\mu_{A'}(\mathbf{x}), \mu_R(\mathbf{x},\mathbf{y})) \,. \tag{5.10}$$

When the implication I in Eq. (5.8) is chosen to be the minimum conjunction operator, μ_R becomes the minimum of μ_{A^k} and μ_{B^k}, and the compositional rule of inference is simplified to the so called max-min or Mamdani inference (Driankov et al. 1993), which can be summarized in the following steps.

(1) The degree of fulfillment β^k of the antecedent is computed for each rule k as

$$\beta^k = \mu_{A_1^k}(x_1) \wedge \mu_{A_2^k}(x_2) \wedge \ldots \wedge \mu_{A_n^k}(x_n), \quad k = 1, \ldots, K \,. \tag{5.11}$$

(2) For each rule derive the output fuzzy set ${B'}^k$ using the minimum t-norm.

$$\mu_{B'^k}(\mathbf{y}) = \beta^k \wedge \mu_{B^k}(\mathbf{y}). \tag{5.12}$$

(3) Aggregate the output fuzzy sets by taking the maximum.

$$\mu_{B'}(\mathbf{y}) = \max_{k=1,2,\ldots,K} (\mu_{B'^k}(\mathbf{y})). \tag{5.13}$$

The application of this fuzzy set algorithm gives as solution the fuzzy set B'. However, in many cases a numerical output value is required, and the output fuzzy set must be defuzzified. The defuzzification transforms a fuzzy set to a single representative numerical value. Two common defuzzification methods are the center-of-gravity (COG) and the mean-of-maxima (MOM). For discrete domains Y, the COG method computes for each coordinate y_i, $i = 1, \dots, p$ the center of gravity for the fuzzy set B' as a weighted sum

$$Z_{y_i}^{\text{cog}}(B') = \frac{\sum_{q=1}^{N_q} \mu_{B'}(\mathbf{y}_q)\, y_{i,q}}{\sum_{q=1}^{N_q} \mu_{B'}(\mathbf{y}_q)}, \tag{5.14}$$

where N_q is the cardinality of the discretized domain Y (*i.e.*, number of quantization levels used for the discretization of Y). The point $\mathbf{y}_q$ is the qth discrete point in the quantization of Y. The MOM method computes the mean value of the interval with the largest membership degree. The MOM method is normally used with the inference based on fuzzy implications, to select the 'most possible' output. A broader discussion of defuzzification is given in Sec. 6.2.

5.2.2 *Singleton fuzzy model*

A special case of the linguistic fuzzy model is obtained when the consequent sets B^k are reduced to fuzzy singletons. This is possible if the dimension of the output is reduced to one, which is represented now by y, and $Y \subset \mathbb{R}$. Singleton sets can be represented as real numbers c^k, yielding the rules

$$R^k\text{: } \textbf{If } \mathbf{x} \text{ is } A^k \textbf{ then } y = c^k, \quad k = 1, 2, \dots, K. \tag{5.15}$$

This model is called the *singleton fuzzy model.* For this model, the COG defuzzification method is reduced to the *fuzzy mean* method, *i.e.*,

$$y = \sum_{k=1}^{K} \frac{\beta^k}{\sum_{j=1}^{K} \beta^j} c^k \,. \tag{5.16}$$

This defuzzification depends on the number of rules K, and not on the number of fuzzy sets for a certain output y_i, $i = 1, \dots, p$, like in Eq. (5.14). The singleton model can also be seen as a special case of the Takagi–Sugeno fuzzy model discussed in Sec. 5.2.3.

Contrary to the linguistic fuzzy model, the consequent parameters c^k of the singleton model can easily be estimated from data by using least squares techniques. Moreover, the singleton model belongs to a general class of function approximators, called the basis functions expansion (Friedman 1991), taking the form

$$y = \sum_{k=1}^{K} \Phi^k(\mathbf{x}) c^k \,. \tag{5.17}$$

Most of the structures used in nonlinear system identification, such as artificial neural networks, radial basis functions or splines belong to this class of systems. In the singleton model, the basis functions $\Phi^k(\mathbf{x})$ are given by the normalized degrees of fulfillment of the rule antecedents, and the constants c^k in Eq. (5.17) are the consequents in Eq. (5.15).

Beyond the fact that singleton fuzzy models are relatively easy to identify, they also have other attractive properties. When the antecedent membership functions are triangular, form a partition of unity and the product t-norm is used to represent the logical **and** connective in the rule antecedents, a multilinear interpolation between the rule consequents is obtained (Brown and Harris 1994). Under certain conditions, this singleton model can be *exactly* inverted, providing a control law based on the inverse of the process model. The inversion of singleton fuzzy models is presented in Sec. 7.2.

5.2.3 *Takagi–Sugeno fuzzy models*

Takagi and Sugeno (1985) introduced a fuzzy rule-based model that consists of a generalization of the singleton model, where the rule consequents are not constants, but crisp functions of the model input according to

$$R^k\text{: } \textbf{If } \mathbf{x} \text{ is } A^k \textbf{ then } y^k = f^k(\mathbf{x}), \quad k = 1, 2, \ldots, K\,, \tag{5.18}$$

where R^k denotes the kth rule, K is the number of rules, $\mathbf{x}$ is the antecedent variable, y is the one dimensional consequent variable and A^k is the (multidimensional) antecedent fuzzy set of the kth rule, as for the linguistic model in Sec. 5.2.1. Each rule k has a different function f^k yielding a different value y^k for the output. This fuzzy model can be generalized for p outputs as

$$R^k\text{: } \textbf{If } \mathbf{x} \text{ is } A^k \textbf{ then } \mathbf{y}^k = \mathbf{f}^k(\mathbf{x}), \quad k = 1, \ldots, K\,. \tag{5.19}$$

Note that the index in the outputs $\mathbf{y}^k$ and the functions $\mathbf{f}^k$ corresponds to the kth rule. For the sake of simplicity, the form in Eq. (5.18) is used in the following. In fact, Eq. (5.19) is the representation of a MIMO fuzzy model, which is decomposed in several MISO systems as in Eq. (5.18). When the states $\mathbf{x}$ are defined as in Eq. (5.3), the MIMO fuzzy model in Eq. (5.19) can be decomposed into a collection of MISO fuzzy models as in Eq. (5.18), without loss of generality. The antecedent proposition A^k can again be a combination, usually in a conjunctive form, of simple propositions for each x_j, $j = 1, \ldots, n$, as in Eq. (5.7). The consequent functions f^k in Eq. (5.18) can be chosen as parameterized functions, where the structure remains the same for all the rules. The most simple and widely used function is the affine linear form, which yields the rules

$$R^k\colon \textbf{If } \mathbf{x} \text{ is } A^k \textbf{ then } y^k = (\mathbf{a}^k)^T\mathbf{x} + b^k\,, \tag{5.20}$$

where $\mathbf{a}^k$ is a parameter vector and b^k is a scalar offset. This model is called an *affine TS model.* The consequents of the affine TS model are hyperplanes in the product space of the inputs and the output, *i.e.*, $\mathbb{R}^n \times \mathbb{R}$. Note that when $\mathbf{a}^k = \mathbf{0},\ k = 1, \ldots, K$, the consequents in model Eq. (5.20) are constant functions, and the model is a *singleton model* as presented in Sec. 5.2.2.

For the sake of simplicity of notation, let $\beta^k \triangleq \mu_{A^k}(\mathbf{x})$ denote the membership function of the antecedent A^k, *i.e.*, the degree of fulfillment of rule k. The inference mechanism proposed by Takagi and Sugeno (1985) is reduced to a straightforward extension of the fuzzy-mean defuzzification presented in Eq. (5.16),

$$y = \sum_{k=1}^{K} \hat{\beta}^k y^k = \sum_{k=1}^{K} \hat{\beta}^k \left((\mathbf{a}^k)^T\mathbf{x} + b^k\right), \tag{5.21}$$

where $\hat{\beta}^k$ is the normalized degree of fulfillment of the kth rule's antecedent given by

$$\hat{\beta}^k = \frac{\beta^k}{\sum_{j=1}^{K} \beta^j}\,. \tag{5.22}$$

When the supports of the antecedent fuzzy sets overlap in the antecedent space, the TS model can be regarded as an approximation of a nonlinear function by local linear functions that are combined.

5.3 Fuzzy identification

The previous sections reviewed the structures and inference mechanisms of different rule-based fuzzy models. The construction of the fuzzy models, usually known as *fuzzy identification*, is now discussed. It is assumed that the structure of the system, *i.e.*, the input and outputs variables, are determined beforehand. For dynamic systems as in Eq. (5.4), the choice of the model structure determines the representation of the dynamics within the fuzzy model. When considering a fuzzy modeling approach, one has to choose the type of the fuzzy model a priori, which depends on the particular application. The inference and defuzzification methods must be chosen afterwards. Finally, the rule base and the membership functions must be derived. In general, TS fuzzy models are more suitable for the identification of nonlinear systems from measured data, while linguistic fuzzy models give a more qualitative description of the system, and as such can be used when dealing with process knowledge. Therefore, it is often useful to develop models of different types for the same system, where each model serves a different purpose such as control design, simulation, prediction, fault detection and user interfaces. A survey of various identification techniques for dynamic systems can be found in Guillaume (2001). In the following sections, we discuss construction of fuzzy models in so far as they are relevant to the material in this book.

Several techniques using neuro-fuzzy identification, such as fuzzy-logic based neurons (Pedrycz 1985) or spline adaptive techniques (Brown and Harris 1994) can be used in fuzzy identification. Local approaches to fuzzy modeling and identification are also increasingly being used (Murray-Smith and Johansen 1997). One of these local modeling techniques is *product-space fuzzy clustering* where local linear models are derived to approximate a nonlinear regression problem by using fuzzy clustering methods. Fuzzy clustering algorithms are unsupervised algorithms that partition a number of data points into a given number of clusters (Höppner et al. 1999). The information regarding the distribution of data can be captured by the fuzzy clusters, which can be used to identify relations between various variables regarding the modeled system. Bezdek and Pal (1992) and Babuška (1998) discuss methods for applying fuzzy clustering methods to obtain fuzzy models. By applying fuzzy clustering on the data obtained from measurements on dynamic systems, fuzzy models of these systems can also be obtained, as discussed in Babuška (1998), Sugeno and Yasukawa (1994), Yoshinari et al. (1993) and Zhao et al. (1997). Modeling of dynamic systems by fuzzy clustering generally entails the following steps.

(1) Determine the model structure suitable to the problem by identifying the relevant system variables. These may be data regarding the system states, output errors or others.
(2) Collect data from the system by measuring, computing or constructing the relevant system variables.
(3) Select a clustering algorithm and determine values of the parameters relevant to the clustering method used.
(4) Select the number of required clusters.
(5) Cluster the data with the selected clustering algorithm.
(6) Obtain membership functions from clusters by projection or otherwise.
(7) Determine a fuzzy rule from each cluster by using the membership functions obtained.
(8) Validate the model.

Typically, the modeling procedure will not follow the above steps successively.

Often, the user iterates several times through different parts of the modeling procedure to come up with an adequate model of the system.

Gustafson and Kessel (1979) have proposed a powerful fuzzy clustering algorithm that is based on adaptive distance measures. It can be used to obtain a nonlinear regression model from a collection of local linear models. This technique has a model structure that is easy to understand and interpret, and can integrate various types of knowledge, such as empirical knowledge, derived from first-principles and measured data (Kaymak et al. 1997). The data of the system can be used to fine tune the parameters of an already existing fuzzy model, which is for instance derived from expert knowledge expressed in a collection of If–Then rules. Another approach, such as clustering in the product space of variables, must be used when no prior knowledge about the system is available, and the fuzzy model is then constructed, based only on measurements. Product-space fuzzy clustering is the identification method used in this text for the identification of fuzzy models, and it is briefly presented in the next section.

5.4 Identification by product-space fuzzy clustering

Assuming that the input and output variables are known, the nonlinear identification problem is solved in two steps.

(1) Structure identification.
(2) Parameter estimation.

These two steps are briefly reviewed below, with attention to the parameter estimation problem. The identification procedure presented below is for affine TS models as in Eq. (5.20). The identification of singleton models, also used in this book and presented in Sec. 5.2.2, is just a particular case of the affine TS model. Product-space clustering can also be advantageously used in the identification of linguistic and fuzzy relational models, as discussed in (Babuška 1998).

5.4.1 *Structure identification*

Structure identification allows us to transform the dynamic identification problem into a static nonlinear regression. Suppose that the structure of the model is given by Eq. (5.4). For the sake of simplicity, let each MISO system be identified separately. As described in Sec. 5.1, the total MIMO system can be derived as a collection of MISO systems. A MISO system can thus be described by

$$\hat{y}(\tau + 1) = f(\mathbf{x}(\tau)). \tag{5.23}$$

Product-space fuzzy clustering is based on the data in the product space $X \times Y$ of the regressor and the regressand. Let N denote the number of data samples, selected from the input and output data sequences. This number must be much larger than the number of states in the system, *i.e.*, $N \gg n$. Let N_d be the number of points actually used in the identification. Also, let τ_h denote the highest order of the inputs and outputs in Eq. (5.3). Then, $N_d = N - \tau_h$. Let Φ denote the regressand matrix in $\mathbb{R}^{N_d \times n}$ having the state vectors $\mathbf{x}(\tau)^T$ in its rows, and Υ denote the vector in $\mathbb{R}^{N_d}$ containing the regressands $y(\tau+1)$, with $\tau = \tau_h, \ldots, N-1$

$$\Phi = \begin{bmatrix} \mathbf{x}(\tau_h)^T \\ \vdots \\ \mathbf{x}(N-1)^T \end{bmatrix}, \quad \Upsilon = \begin{bmatrix} y(\tau_h+1) \\ \vdots \\ y(N) \end{bmatrix}. \tag{5.24}$$

The matrix Φ contains shifted versions of the input and output data, as in Eq. (5.3). An example is presented in Example 5.1. In this example $\tau_h = 2$, and $N_d = N-2$.

Example 5.1 Let an NARX model be given by $\hat{y}(\tau+1) = f(y_1(\tau), y_2(\tau), y_2(\tau-1), u(\tau))$, and let the considered output of the MISO model be y_1: $y \triangleq y_1$. Having N data samples for y_1, y_2, u_1 and u_2, the regressor matrix and the regressand vector are given by

$$\Phi = \begin{bmatrix} y_1(2) & y_2(2) & y_2(1) & u_1(2) \\ y_1(3) & y_2(3) & y_2(2) & u_1(3) \\ \vdots & \vdots & \vdots & \vdots \\ y_1(N-1) & y_2(N-1) & y_2(N-2) & u_1(N-1) \end{bmatrix}, \quad \Upsilon = \begin{bmatrix} y_1(3) \\ y_1(4) \\ \vdots \\ y_1(N) \end{bmatrix}.$$

Assuming that the structure is correctly chosen, the unknown nonlinear mapping between Υ and Φ can be estimated from the data set. The structural parameters $m_1, \ldots, m_m$ and $p_1, \ldots, p_p$, as in Eq. (5.3), are chosen either on the basis of prior knowledge or automatically by comparing different structures in terms of some suitable criteria (Sugeno and Kang 1988).

5.4.2 *Parameter estimation*

At this step, the number of rules K, the antecedent fuzzy sets A^k, and the consequent parameters $\mathbf{a}^k$, b^k for $i = 1, \ldots, K$, as in Eq. (5.20), are determined. Fuzzy clustering in the Cartesian product space $X \times Y$ is applied to partition the data into subsets, which can be approximated by local linear models (Babuška 1998). Cluster analysis classifies objects according to similarities among them. In system identification, clustering finds relationships between the system variables.

The data set $\mathbf{Z}$ to be clustered is formed by appending Υ to Φ,

$$\mathbf{Z} = [\Phi, \Upsilon]^T . \tag{5.25}$$

The columns of $\mathbf{Z}$ are denoted by $\mathbf{z}_\ell, \ell = 1, \ldots, N_d$. Let $\mathbf{U} = [\mu_{k\ell}] \in [0,1]^{K \times N_d}$ denote a fuzzy partition matrix of $\mathbf{Z}$. Let $\mathbf{V}$ be a vector of cluster prototypes (centers) to be determined, defined by $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K]$, and let $\mathbf{F}$ be a set of cluster covariance matrices $\mathbf{F} = [\mathbf{F}_1, \ldots, \mathbf{F}_K]$, where $\mathbf{F}_k$ are positive definite matrices in $\mathbb{R}^{(n+1)\times(n+1)}$. The GK algorithm searches for an optimal fuzzy partition $\mathbf{U}$, the prototype matrix of cluster means $\mathbf{V}$, and a set of cluster covariance matrices $\mathbf{F}$. In other words,

$$(\mathbf{Z}, K) \xrightarrow{\text{clustering}} (\mathbf{U}, \mathbf{V}, \mathbf{F}) . \tag{5.26}$$

The optimization minimizes the following objective function,

$$J(\mathbf{Z}, \mathbf{U}, \mathbf{V}) = \sum_{k=1}^{K} \sum_{\ell=1}^{N_d} (\mu_{k\ell})^\alpha d_{k\ell}^2 , \tag{5.27}$$

where α is a weighting parameter. The function $d_{k\ell}$ is the distance of a data point $\mathbf{z}_\ell$ to the cluster prototype $\mathbf{v}_k$. In the Gustafson–Kessel clustering algorithm, the distance is computed from the covariance matrices according to

$$d_{k\ell}^2 = (\mathbf{z}_\ell - \mathbf{v}_k)^T \frac{\mathbf{F}_k^{-1}}{|\mathbf{F}_k|^{1/(n+1)}} (\mathbf{z}_\ell - \mathbf{v}_k) , \tag{5.28}$$

where $|\mathbf{F}_k|$ is the determinant of the covariance matrix $\mathbf{F}_k$. The GK algorithm is summarized in Algorithm 5.1.

Algorithm 5.1 Gustafson–Kessel algorithm.

Given the data set $\mathbf{Z}$, choose the number of fuzzy rules (clusters) $1 < K \ll N$, the weighting exponent $\alpha > 1$ and the termination tolerance $\epsilon > 0$. Initialize the partition matrix randomly.

Repeat for $l = 1, 2, \ldots$

Step 1: **Compute cluster means (prototypes):**

$$\mathbf{v}_k^{(l)} = \frac{\sum_{\ell=1}^{N_d} (\mu_{k\ell}^{(l-1)})^\alpha \mathbf{z}_\ell}{\sum_{\ell=1}^{N_d} (\mu_{k\ell}^{(l-1)})^\alpha}, \quad 1 \le k \le K .$$

Step 2: **Compute covariance matrices:**

$$\mathbf{F}_k = \frac{\sum_{\ell=1}^{N_d} (\mu_{k\ell}^{(l-1)})^\alpha (\mathbf{z}_\ell - \mathbf{v}_k^{(l)})(\mathbf{z}_\ell - \mathbf{v}_k^{(l)})^T}{\sum_{\ell=1}^{N_d} (\mu_{k\ell}^{(l-1)})^\alpha}, \quad 1 \le k \le K .$$

Step 3: **Compute distances:**

$$d_{k\ell}^2 = (\mathbf{z}_\ell - \mathbf{v}_k^{(l)})^T |\mathbf{F}_k|^{\frac{1}{n+1}} \mathbf{F}_k^{-1} (\mathbf{z}_\ell - \mathbf{v}_k^{(l)}) \,.$$

Step 4: **Update partition matrix:**

if $d_{k\ell} > 0$ for $1 \leq k \leq K, \quad 1 \leq \ell \leq N_d$,

$$\mu_{k\ell}^{(l)} = \frac{1}{\sum_{j=1}^{K} (d_{k\ell}/d_{j\ell})^{2/(\alpha-1)}},$$

otherwise

$$\mu_{k\ell}^{(l)} = 0 \text{ if } d_{k\ell} > 0, \text{ and } \mu_{k\ell}^{(l)} \in [0,1]$$
$$\text{with } \sum_{k=1}^{K} \mu_{k\ell}^{(l)} = 1 \,.$$

until $||\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}|| < \epsilon$.

Given the triplet $(\mathbf{U}, \mathbf{V}, \mathbf{F})$ obtained by the GK algorithm, the antecedent membership functions A^k can be computed, and hence the consequent parameters $\mathbf{a}^k$ and b^k can be calculated, as explained in the following.

5.4.2.1 *Number of clusters*

The number of clusters determines the number of rules in the obtained fuzzy model. This number is an important parameter that influences the accuracy and transparency of the fuzzy models, and it has thus been considered extensively in the literature (Setnes, Babuška, Kaymak and van Nauta Lemke 1998, Kaymak and Setnes 2000, Frigui and Krishnapuram 1996). Two main strategies to determine the appropriate number of clusters in data can be distinguished.

- Cluster the data for different values of K and then use a mathematical expression to assess the goodness of the obtained partitions. This approach is called the *validity measures* approach. Different validity measures have been proposed in connection with adaptive distance clustering techniques (Gath and Geva 1989).
- Start with a sufficiently large number of clusters and reduce this number successively by combining clusters that are compatible with respect to some predefined criteria. This is a cluster merging approach. A cluster merging approach called *compatible cluster merging* is discussed in Chapter 6.

5.4.2.2 *Antecedent membership functions*

Each cluster represents one TS fuzzy rule, as in Eq. (5.20). The multidimensional membership functions A^k are given analytically by computing the distance of $\mathbf{x}(\tau)$ from the projection of the cluster center $\mathbf{v}_k$ onto X, and then computing the membership degree in an inverse proportion to the distance. Denote with $\mathbf{F}_k^x = [f_{jl}]$, $1 \le j, l \le n$, the submatrix of $\mathbf{F}_k$. This matrix describes the form of the cluster in the antecedent space X. Let $\mathbf{v}_k^x = [v_{1k}, \ldots, v_{nk}]^T$ denote the projection of the cluster center onto the antecedent space X. From the GK algorithm, the inner-product distance norm, given by

$$d_{k\ell} = (\mathbf{x}(\tau) - \mathbf{v}_k^x)^T |(\mathbf{F}_k^x)|^{1/n} (\mathbf{F}_k^x)^{-1} (\mathbf{x}(\tau) - \mathbf{v}_k^x) . \tag{5.29}$$

is converted into the membership degree by

$$\mu_{A^k}(\mathbf{x}(\tau)) = \frac{1}{\sum_{j=1}^{K} (d_{k\ell}/d_{j\ell})^{2/(\alpha-1)}} , \tag{5.30}$$

where α is the fuzziness parameter of the GK algorithm given in Algorithm 5.1.

5.4.2.3 *Consequent parameters*

Optimal consequent parameters are estimated by the least-squares method. Let $(\theta^k)^T = \left[(\mathbf{a}^k)^T, b^k\right]$, let $\mathbf{\Phi}_e$ denote the matrix $[\mathbf{\Phi}, \mathbf{1}]$, and let $\mathbf{\Gamma}^k$ denote a diagonal matrix in $\mathbb{R}^{N_d \times N_d}$ having the membership degree $\mu_{A^k}(\mathbf{x}(\tau))$ as its ℓth diagonal element. Denote $\mathbf{\Phi}'$ the matrix in $\mathbb{R}^{N_d \times K(n+1)}$ composed from matrices $\mathbf{\Gamma}^k$ and $\mathbf{\Phi}_e$ as follows

$$\mathbf{\Phi}' = \left[(\mathbf{\Gamma}^1 \mathbf{\Phi}_e), (\mathbf{\Gamma}^2 \mathbf{\Phi}_e), \ldots, (\mathbf{\Gamma}^K \mathbf{\Phi}_e)\right] . \tag{5.31}$$

Denote θ' the vector in $\mathbb{R}^{K(n+1)}$ given by

$$\theta' = \left[(\theta^1)^T, (\theta^2)^T, \ldots, (\theta^K)^T\right]^T . \tag{5.32}$$

The resulting least squares problem, $\Upsilon = \mathbf{\Phi}'\theta' + \epsilon$, has the solution

$$\theta' = \left[(\mathbf{\Phi}')^T \mathbf{\Phi}'\right]^{-1} (\mathbf{\Phi}')^T \Upsilon . \tag{5.33}$$

The optimal parameters $\mathbf{a}^k$ and b^k are given by

$$\begin{aligned} \mathbf{a}^k &= [\theta'_{s+1}, \theta'_{s+2}, \ldots, \theta'_{s+n}]^T , \\ b^k &= [\theta'_{s+n+1}], \text{ where } s = (k-1)(n+1) . \end{aligned} \tag{5.34}$$

With the determination of the parameters $\mathbf{a}^k$ and b^k, the fuzzy model identification procedure is completed. If several outputs are considered, the procedure must be repeated for each output.

5.5 Summary and concluding remarks

This chapter presented the modeling and identification techniques used in this manuscript. These techniques are used as tools. In the next chapters, models based on first-principles or mathematical descriptions of a system are used when this is possible. The latter type of models, also known as white-box models, are used for control simulations or test case purposes. Unfortunately, this type of models can usually not be obtained because of the poor knowledge of the system, or they are too complex to be used in control applications. For this type of problem, which is mainly considered in this text, models extracted from data of several inputs and outputs of the system, possibly combined with other sources of knowledge, such as empirical or mathematical laws, are derived.

The formulation of the modeling problem is described in the beginning of this chapter in order to introduce the necessary notation definitions. Fuzzy modeling is chosen from several modeling techniques based on soft computing due to its interesting properties. In fact, fuzzy models can incorporate different types of knowledge, and a gray-box model can be derived, *i.e.*, a model that is not described by mathematical principles, but that can be easily interpretable, because it is a rule-based linguistic model. Linguistic fuzzy models, singleton models and TS fuzzy models are briefly described.

The identification of fuzzy models from different types of knowledge, is usually known as fuzzy identification. When only data is available, product-space fuzzy clustering has several advantages over other fuzzy identification techniques, such as the possible combination of different types of knowledge, and it is usually easy to interpret. The identification procedure of a TS fuzzy model has been described. It starts by defining the structure of the system. The parameters are estimated by using a clustering algorithm, such as the GK algorithm. After determining the number of clusters, the identification process must run the clustering algorithm, and extract the antecedent membership functions and the consequent parameters. Several examples of the application of this identification algorithm are presented in the next chapters.

Chapter 6

Fuzzy Decision Making for Modeling

Fuzzy decision making methods can be applied to support the identification and construction of fuzzy inference systems in fields related to control engineering, such as systems modeling. This chapter considers applications of fuzzy decision making in fuzzy modeling by using product-space fuzzy clustering and in defuzzification.

Various model parameters must be determined to obtain a good fuzzy model when the models are obtained from data. An important parameter for fuzzy models is the number of rules in the rule base. This parameter is a reflection of the trade-off between the required model accuracy and the reduction of the model complexity. Fuzzy clustering is a widely used method for obtaining fuzzy models, as discussed in Chapter 5. Determination of a relevant number of rules corresponds to the determination of a correct number of clusters for adequately describing the modeled system. We describe in Sec. 6.1 a method for determining a relevant number of clusters when identifying Takagi–Sugeno fuzzy models by using fuzzy clustering. Starting from a system description with a large number of rules, a multicriteria decision step determines whether the number of rules in the rule base can be decreased, leading to the simplification of models. This is one application of fuzzy decision making.

Defuzzification is an important part of fuzzy systems, since many fuzzy systems must eventually provide the user with a crisp outcome. Defuzzification can be seen as an operation that replaces a fuzzy set by its representative crisp value. In terms of decision making, the problem is formulated as the selection of a crisp value that best represents a fuzzy set, given the goals of the fuzzy system. Many defuzzification methods have been suggested in literature. These defuzzification methods have in common that the defuzzification operation is equally sensitive to different elements in the domain over which defuzzification takes place. Hence, the defuzzification operator is not biased for different elements of the domain. The decision making interpretation of defuzzification points out, however, that defuzzification is not an operation void of any context, but that it must be selected to match and satisfy the goals for which the fuzzy system is designed. Section 6.2 describes a fuzzy decision making approach to defuzzification, which leads to a new defuzzification operator that is not equally sensitive to the elements over which the defuzzification takes place. An application to the security analysis of power networks, where the new defuzzification operator is used, is given as an example in Sec. 6.3 before the chapter concludes with a summary and several concluding remarks regarding the applications.

6.1 Fuzzy decisions in fuzzy modeling

Many algorithms in control engineering require the determination of the values of certain parameters in order to obtain satisfactory results. A number of criteria must be considered, and a suitable parameter value is determined from the degree to which different performance criteria are satisfied. Many soft computing methods require the specification of parameters such as the learning rate in the back-propagation learning rule (Rumelhart et al. 1986). The successful use of methods depends on a correct specification of such parameters. The determination of fuzzy models from system measurements also requires the correct specification of a number of parameters, such as the number of rules and the definitions of membership functions. The values of these parameters are sometimes specified by the control engineer based on experience, on previous knowledge, and on trial and error. However, the designers can benefit significantly from methods that automate most of the fuzzy modeling. A decision making algorithm for selecting relevant values of the modeling parameters can then prove to be a useful aid to the designer, by reducing the effort of obtaining fuzzy models from data-driven approaches and thereby reducing the time needed for the design procedure.

6.1.1 *Fuzzy models from clustering*

Fuzzy modeling by using product-space clustering has been discussed in Sec. 5.4. Fuzzy modeling techniques are becoming increasingly popular for modeling complex systems to which standard linear methods cannot be applied due to insufficient knowledge about the underlying physical mechanisms, process nonlinearity and parameter uncertainty. Most fuzzy models are based on the structure proposed by Mamdani (1974) or Takagi and Sugeno (1985). Takagi–Sugeno (TS) models differ from Mamdani models in that their consequents are linear functions of the antecedent variables instead of fuzzy sets. Recall from Chapter 5 that the rule base in MISO Takagi–Sugeno models have the following structure,

$$R^k\text{: } \textbf{If } x_1 \text{ is } A_1^k \textbf{ and } x_2 \text{ is } A_2^k \textbf{ and } \dots \textbf{ and } x_n \text{ is } A_n^k$$
$$\textbf{then } y^k = \sum_{j=1}^{n} a_j^k x_j + b^k, \tag{6.1}$$

where R^k is the kth rule in the rule-base, $x_1, \dots, x_n$ are the premise variables, y^k is the output of the kth rule and $A_1^k, \dots, A_n^k$ are the fuzzy sets defined over their respective universes of discourse. The overall output y^* of the model is calculated by a convex weighted sum of each of the rule consequents,

$$y^* = \frac{\sum_{k=1}^{K} y^k \beta^k}{\sum_{k=1}^{K} \beta^k} \tag{6.2}$$

where K is the total number of rules, β^k is the non-normalized degree of fulfillment of the kth rule premise and y^k is the output of rule k.

TS models can be obtained by applying the Gustafson–Kessel clustering algorithm (Gustafson and Kessel 1979) for clustering data in the product space of the antecedent and the consequent variables, as discussed in Sec. 5.4. One of the important parameters that must be determined when applying fuzzy clustering methods is the number of required clusters. This number determines the number of rules in the model obtained. Hence, correct specification of this parameter is important, because a large number increases unnecessarily the model complexity and redundancy, while a small number decreases model accuracy. Unfortunately, fuzzy clustering algorithms such as fuzzy c–means (Bezdek 1981) or the Gustafson–Kessel algorithm do not give an indication of the correct number of clusters needed. They just partition the data into the specified number of clusters, no matter whether the partition obtained is meaningful or not. Consequently, methods have been suggested for determining the 'optimal' number of clusters in a clustering problem.

The conventional approach to determine a correct number of clusters in cluster analysis is based on validity measures. In general, clustering algorithms aim at locating well-separated and compact clusters. When the number of clusters is chosen equal to the number of groups that actually exist in the data, it can be expected that the clustering algorithm will identify them correctly. When this is not the case, some misclassification can be made and the clusters are not likely to be well separated and compact. A cluster validity measure can quantify the separation and the compactness of the clusters. However, as (Bezdek 1981, p. 98) points out, the formulation of the cluster validity problem in a mathematically tractable manner is extremely difficult. The definition of compactness and separation for a specific data set, as well as the definition of a 'good' cluster is open to interpretation and can be formulated in different ways. Consequently, the literature contains many validity measures, some of which can be found in Bezdek (1981), Backer (1995) and Gath and Geva (1989).

Another method to determine the correct number of clusters is to start with a large number of clusters and to reduce this number by merging similar clusters until no more clusters can be merged. The compatible cluster merging (CCM) techniques suggested by Krishnapuram and Freg (1992) are an example of this approach. CCM is especially attractive when an upper bound on the number of clusters may be estimated. In this section, the application of the CCM algorithm to fuzzy modeling is considered and a reduction method is described which uses a decision making step to determine the relevant number of clusters (Kaymak and Babuška 1995). An overview of various reduction methods for the simplification of fuzzy models and the determination of a relevant number of rules is found in Kaymak et al. (1997).

6.1.2 *Compatible cluster merging*

Denote the number of clusters in a clustering problem with K, since the number of clusters equals the number of rules in the fuzzy modeling scheme described in Sec. 5.4. When an upper limit on the number of required clusters can be estimated, a correct number for K may be determined by using the compatible cluster merging technique. The cluster merging technique evaluates the clusters for their compatibility (*i.e.*, similarity) to one another and merges the clusters that are found to be compatible. Then, the clustering is performed again with the new number of clusters. Although similar to the validity approach, the technique differs in that an upper estimate K_m on the number of clusters is made and the number of clusters is gradually reduced by merging, until an appropriate number is found.

A compatible cluster merging (CCM) technique for GK clustering has been introduced by Krishnapuram and Freg (1992). In the following, we present a modified version of this technique based on the material from Kaymak and Babuška (1995).

Let the centers of two clusters be $\mathbf{v}_i$ and $\mathbf{v}_j$. Let the eigenvalues of the covariance matrices of the two clusters be $\{\lambda_{i1}, \ldots, \lambda_{ip'}\}$ and $\{\lambda_{j1}, \ldots, \lambda_{jp'}\}$ respectively, both arranged in descending order. Let the corresponding eigenvectors be $\{\phi_{i1}, \ldots, \phi_{ip'}\}$ and $\{\phi_{j1}, \ldots, \phi_{jp'}\}$. We define the following compatibility criteria.

$$\zeta_{ij}^1 = |\phi_{ip'} \cdot \phi_{jp'}|, \quad \zeta_{ij}^1 \text{ close to } 1, \tag{6.3}$$

$$\zeta_{ij}^2 = ||\mathbf{v}_i - \mathbf{v}_j||, \quad \zeta_{ij}^2 \text{ close to } 0. \tag{6.4}$$

Figure 6.1 shows the graphical illustration of the compatibility criteria in Eq. (6.3) and Eq. (6.4). Equation (6.3) states that the parallel hyperplane clusters should be

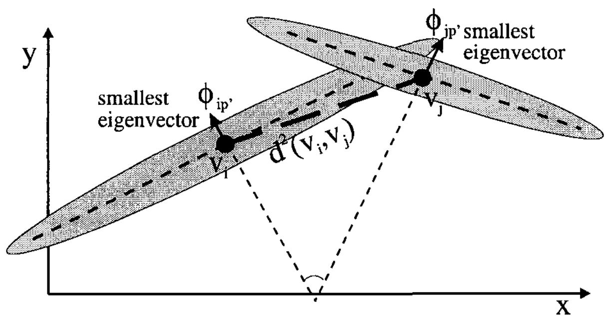


Fig. 6.1 Graphical illustration of compatible cluster merging criteria. Reproduced from (Kaymak and Babuška 1995), ©1995 IEEE.

merged. Equation (6.4) states that the cluster centers should be sufficiently close for merging. Figure 6.2 depicts various cases that these criteria cover. Clusters 1 and 4 are parallel, but they are not close. Hence, they are not compatible. Clusters 1 and 2, or 3 and 4 are not compatible either, since they are close to each other but not parallel. Clusters 2 and 3 are compatible since they are both parallel and close to each other.

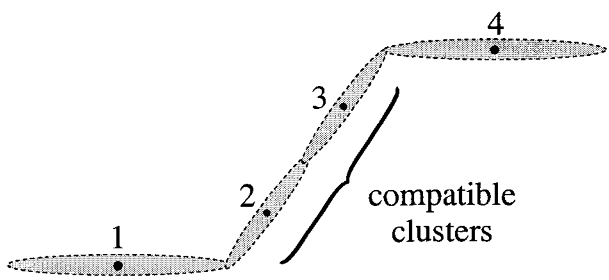


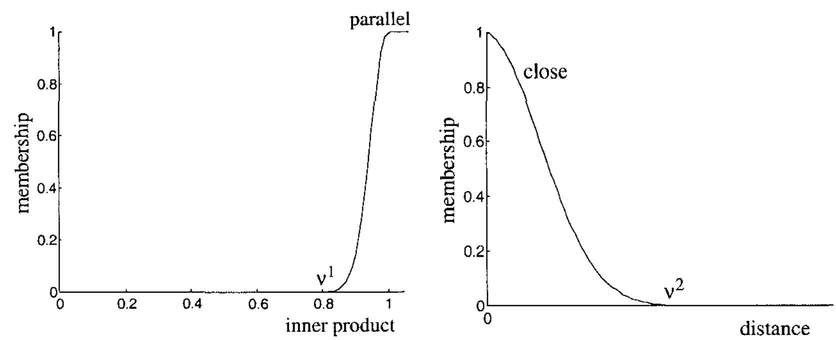
Fig. 6.2 Examples of compatible and incompatible clusters.

6.1.3 *The decision making algorithm*

Although the compatibility criteria quantify various aspects of the similarity of clusters, the overall cluster compatibility is obtained through the aggregation of the compatibility criteria. A fuzzy decision making algorithm is used for the aggregation of the criteria. The fuzzy decision making is especially useful as parallelism and closeness are gradual concepts, and different types of aggregation behavior can be studied by using different operators from the fuzzy set theory.

The goal of the decision making step is to determine which pairs of clusters can be merged. The decision alternatives are the possible pairs of clusters. When K clusters are considered, the total number of decision alternatives is $K(K-1)/2$. The compatibility criteria in Eq. (6.3) and Eq. (6.4) are evaluated for each pair of clusters. This leads to the matrix $\mathbf{Z}^1$ for Eq. (6.3), and to the matrix $\mathbf{Z}^2$ for Eq. (6.4). Both $\mathbf{Z}^1$ and $\mathbf{Z}^2$ are symmetric matrices. $\mathbf{Z}^1$ has all 1's on its main diagonal, since a cluster is always parallel to itself. $\mathbf{Z}^2$ has all 0's on its main diagonal, as the distance of a cluster center to itself is zero.

Following the fuzzy decision making approach, the decision goals for each criterion must be defined by using a fuzzy set. In this problem, this means that the fuzzy set 'close to 1' must be defined for Eq. (6.3), which represents the parallelism of the clusters, and 'close to 0' must be defined for Eq. (6.4), which represents the closeness of the clusters. Figure 6.3 shows the membership functions defined for the two criteria. The important parameters for the membership functions are the limits of their support, characterized by the elements $(\nu^1, 0)$ and $(\nu^2, 0)$.



(a) Parallelism *(b) Closeness*

Fig. 6.3 Membership functions for parallelism and closeness of clusters.

The values of ν^1 and ν^2 are found by averaging compatibility values, except for the elements on the main diagonal of $\mathbf{Z}^1$ and $\mathbf{Z}^2$, according to

$$\nu^1 = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \zeta_{ij}^1, \tag{6.5}$$

$$\nu^2 = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \zeta_{ij}^2. \tag{6.6}$$

The elements on the main diagonal are not considered, as a cluster is always fully compatible with itself, and hence these elements do not provide any information. The varying support for the membership functions ensures the adaptability of the algorithm to specific merging problems that are encountered in practice.

Evaluating the membership functions in Fig. 6.3 with the values of ζ_{ij}^1 and ζ_{ij}^2, one obtains the degree of parallelism μ_{ij}^1 and closeness μ_{ij}^2 for the cluster pair i and j. Having determined the degree of parallelism and the closeness of the clusters, the overall cluster compatibility is determined by the aggregation of the two criteria. A fuzzy aggregation operator is used for this purpose. Note that the parallelism and the closeness of clusters partially compensate each other. In other words, two clusters that are not quite parallel but very close may need to be merged. The same also applies to the clusters that are parallel but somewhat far from each other. Taking this fact into account, the generalized averaging operator of Eq. (3.24) is a good candidate as the aggregation operator.

The outcome of the decision procedure is the overall compatibility matrix $\mathbf{S}$. The elements s_{ij} of the compatibility matrix $\mathbf{S}$ are given by

$$s_{ij} = \left[\frac{(\mu_{ij}^1)^\gamma + (\mu_{ij}^2)^\gamma}{2} \right]^{1/\gamma}, \quad \gamma \in \mathbb{R}. \tag{6.7}$$

The selection of the parameter γ is discussed in Sec. 6.1.6. The compatibility matrix $\mathbf{S}$ is a symmetric matrix whose main diagonal consists of 1s by definition. The element s_{ij} denotes the compatibility of cluster i with cluster j.

6.1.4 *Merging clusters*

Given the compatibility matrix $\mathbf{S}$, the clusters that will be merged must be identified and combined. Clusters can be merged in several ways. One possibility is to merge the most similar pair of clusters, as long as the value of the corresponding s_{ij} is above a threshold s^*. In practice, this method merges two clusters at each merger. The disadvantage of the method is that clustering must be made for all K, $K' \leq K \leq K_m$, where K' is the 'correct' number of clusters for describing the data, and K_m is the initial number of clusters used by the algorithm.

Another merging method is based on fuzzy relational clustering (Dunn 1974, Yang 1993). Relational clustering computes the transitive closure of the matrix $\mathbf{S}$ by applying the max-min composition successively. The transitive closure of a fuzzy relation R is the smallest relation that is transitive and contains R. Given a fuzzy relation R, its max-min transitive closure R_T can be calculated by using the following iterative algorithm (Klir and Yuan 1995).

(1) Let $R_T^{(0)} = R$.
(2) Repeat for iteration number $l = 1, 2, \ldots$

- $R_T^{(l)} = R^{(l-1)} \cup \left(R^{(l-1)} \circ R^{(l-1)}\right)$

until $R_T^{(l)} = R_T^{(l-1)}$,

where $\circ$ denotes the max-min composition of fuzzy relations.

The transitive matrix determined by the algorithm above indicates groups of clusters that are similar at least to the degree denoted by the matrix elements. By applying a pre-determined and problem dependent threshold s^*, the groups of clusters that need to be merged are identified.

Example 6.1 Consider the following compatibility matrix.

$$\mathbf{S} = \begin{pmatrix} 1 & 0.4 & 0.8 & 0.3 \\ 0.4 & 1 & 0.6 & 0 \\ 0.8 & 0.6 & 1 & 0 \\ 0.3 & 0 & 0 & 1 \end{pmatrix}.$$

The transitive closure $\mathbf{S}_T$ of $\mathbf{S}$ is equal to

$$\mathbf{S}_T = \begin{pmatrix} 1 & 0.6 & 0.8 & 0.3 \\ 0.6 & 1 & 0.6 & 0.3 \\ 0.8 & 0.6 & 1 & 0.3 \\ 0.3 & 0.3 & 0.3 & 1 \end{pmatrix}.$$

If s^* is chosen to be 0.7, one obtains the following partition

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

which means that the clusters 1 and 3 are to be merged.

Note that more than two clusters may also be merged, depending on the threshold s^*. In this way, computationally intensive calculations are avoided, which reduces the computation time.

Example 6.2 Consider the compatibility matrix discussed in Example 6.1. When a threshold is applied on it with a s^* value of 0.5, one obtains the partition

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

which implies that clusters 1, 2 and 3 can be merged.

In general, lowering the value of s^* leads to increased cluster merging. For $s^* = 0$, all clusters will be merged, while for $s^* = 1$, no clusters will be merged. Hence, for $s^* = 1$, the CCM algorithm reduces to the Gustafson–Kessel algorithm with $K = K_m$, the maximum number of clusters. The selection of the threshold value s^* requires tuning, but a value between 0.3 and 0.7 works for most problems. After the merging step, the clustering algorithm is re-initialized with the merged partition and the data are clustered again with the reduced number of clusters.

6.1.5 *Heuristic step*

Sometimes, it is possible that the merging process is not sufficiently discriminating to uniquely determine which clusters should be merged. The problem arises from the fact that the Gustafson–Kessel clusters possess a particular shape, which is not considered by the criteria of Eq. (6.3) and Eq. (6.4). This is illustrated in Example 6.3.

Example 6.3 Consider three clusters that are depicted in Fig. 6.4. Clusters 1 and 2 are parallel. Further, cluster center $\mathbf{v}_3$ is slightly closer to cluster center $\mathbf{v}_1$ than cluster center $\mathbf{v}_2$. Suppose that Eq. (6.3) and Eq. (6.4), together with the membership functions in Fig. 6.3, lead to the membership values

$$\begin{pmatrix} 1 & 1 & 0.7 \\ 1 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 0.6 & 0.7 \\ 0.6 & 1 & 0.5 \\ 0.7 & 0.5 & 1 \end{pmatrix},$$

respectively. Assuming that the geometric mean is used for the aggregation as originally proposed by Kaymak and Babuška (1995), the overall compatibility matrix becomes

$$\mathbf{S} = \begin{pmatrix} 1 & 0.77 & 0.7 \\ 0.77 & 1 & 0.59 \\ 0.7 & 0.59 & 1 \end{pmatrix},$$

whose transitive closure is equal to

$$\mathbf{S}_T = \begin{pmatrix} 1 & 0.77 & 0.7 \\ 0.77 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{pmatrix}.$$

When the threshold s^* is chosen as 0.75, one obtains the crisp partition matrix

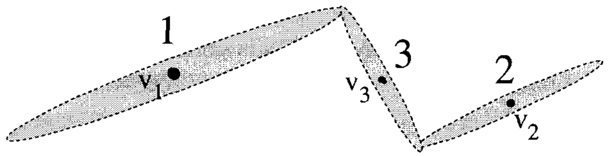


Fig. 6.4 Example of cluster merging where the merging criteria may lead to over-merging.

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which implies that clusters 1 and 2 should be merged. However, cluster 3 is located between them, geometrically, and hence clusters 1 and 2 should not be merged.

A heuristic step is introduced to identify the cases where the merging of clusters would be impermissible. Note that the compatible clusters should not be merged when there is an incompatible cluster in their mutual neighborhood. The *mutual neighborhood* is defined as the region of the antecedent product space of the fuzzy model, which is located within a certain distance of the compatible cluster centers. Mathematically, the merging condition for the compatible clusters is

$$\min_{\mathbf{v}_k \notin M} \max_{\mathbf{v}_i \in M} d_{ik} > \max_{\mathbf{v}_i \in M} \max_{\mathbf{v}_j \in M} d_{ij}, \tag{6.8}$$

where M is a group of compatible clusters, and

$$d_{ij} = |\mathrm{P}(\mathbf{v}_i) - \mathrm{P}(\mathbf{v}_j)|$$

with $\mathrm{P}(\cdot)$ representing the projection of the cluster centers onto the antecedent product space. The heuristic states that the compatible clusters are merged when

Eq. (6.8) is satisfied. The merging process is iterated until no more clusters can be merged.

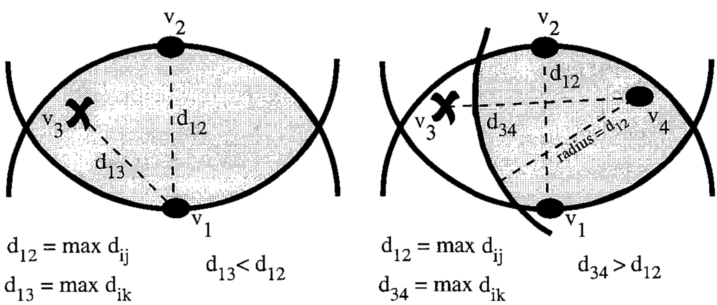
Example 6.4 Figure 6.5a illustrates the concept of mutual neighborhood for two-dimensional antecedent product space. There are two compatible clusters denoted by centers $\mathbf{v}_1$ and $\mathbf{v}_2$. The shaded region is the mutual neighborhood given by Eq. (6.8). Since cluster center $\mathbf{v}_3$ is within the mutual neighborhood of $\mathbf{v}_1$ and $\mathbf{v}_2$, the clusters are not merged. Figure 6.5b shows a similar situation, but this time with three compatible clusters. The mutual neighborhood of the clusters 1, 2 and 4 is defined by the distance d_{12}, since

$$\max_{\mathbf{v}_i \in \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4\}} \max_{\mathbf{v}_j \in \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4\}} d_{ij} = d_{12}.$$

The left hand side of Eq. (6.8) is given by

$$\min_{\mathbf{v}_k \in \{\mathbf{v}_3\}} \max_{\mathbf{v}_i \in \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4\}} d_{ik} = \min_{\mathbf{v}_k = \mathbf{v}_3} \max(d_{31}, d_{32}, d_{34}) = d_{34}.$$

The mutual neighborhood is now more restricted, and the compatible clusters may be merged, since $d_{34} > d_{12}$.



(a) Compatible clusters may not be merged. *(b) Compatible clusters may be merged.*

Fig. 6.5 A group of compatible clusters are not merged if an incompatible cluster center is within their mutual neighborhood (the shaded region) in the antecedent product space.

6.1.6 *Selection of the decision function*

Since the overall compatibility is determined by the aggregation operation, the aggregation function influences the merging process. Because trade-off is required between the criteria for the parallelism and the closeness, the generalized averaging operator is a likely candidate for the aggregation. The geometric mean is used for the aggregation initially. However, experiments have suggested that the geometric mean allows very little compensation between the criteria. Therefore, the parametric form of the generalized averaging operator in Eq. (6.7) is used, and the influence of changing the parameter value is studied as follows.

Various data sets are clustered, and the number of clusters that is found by the compatible cluster merging algorithm is compared for different values of the parameter γ and the threshold s^*. The heuristic step is not used in this study. It is found that $\gamma = 0$ (geometric mean) leads to too little compensation, while $\gamma = 1$ (arithmetic mean) leads to too much compensation and over-merging. The value of $\gamma = 0.5$ is empirically determined to give satisfactory results for most problems. Figure 6.6 shows the results of one such study, which involves the modeling of the pressure dynamics of a feed-batch fermentor. The details of the system are described in Sec. 7.6. It is known that this particular system can be modeled with three fuzzy rules, and therefore the cluster merging algorithm should find three clusters. Figure 6.6 depicts the number of clusters that the compatible cluster merging algorithm finds for the feed-batch fermentor data with different values of the aggregation operator's parameter γ and the threshold s^*. In each case, the algorithm started with 10 clusters. In general, more clusters are merged as the value of s^* decreases. Similarly, more clusters are also merged for increasing values of γ. As seen in Fig. 6.6, there is a sharp transition region from under-merging behavior to over-merging behavior between $\gamma = -1$ and $\gamma = 2$. For $\gamma = 0.5$ the sensitivity to the threshold s^* is not very high (the number of clusters determined does not change for $0.4 < s^* < 0.55$). Therefore, it is a good candidate for the aggregation. The aggregation operator becomes

$$s_{ij} = \left(\frac{\sqrt{\mu_{ij}^1} + \sqrt{\mu_{ij}^2}}{2} \right)^2 . \tag{6.9}$$

6.1.7 *Compatible cluster merging algorithm*

Various steps of the compatible cluster merging algorithm have been explained in previous sections. The total algorithm is summarized in Algorithm 6.1.

Algorithm 6.1 Modified compatible cluster merging (CCM) algorithm. Given a data set $\mathbf{z}_\ell$, $\ell = 1, \ldots, N_d$ with $\mathbf{z}_\ell = (x_{\ell 1}, \ldots, x_{\ell n}, y_\ell)^T$, choose the maximum number of clusters K_m and the merging threshold s^*.

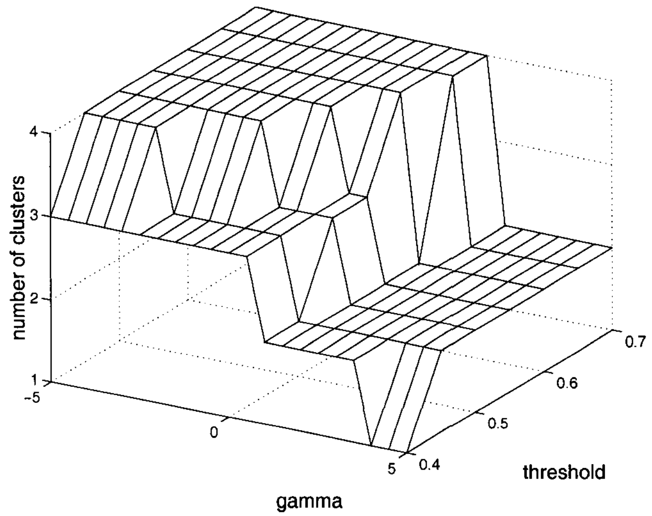


Fig. 6.6 The number of clusters determined by the CCM algorithm for modeling a feed-batch fermentor. The results are depicted for different averaging operators and different similarity thresholds without the heuristic step.

Choose the parameters for the GK algorithm, *i.e.* the stopping criterion ϵ and the fuzziness parameter α. Set $K = K_m$.
Repeat:

1. Perform GK clustering with K clusters.
2. Evaluate the compatibility criteria in Eq. (6.3) and Eq. (6.4) for every pair of clusters.
3. Calculate the overall compatibility s_{ij} by using the membership functions from Fig. 6.3, the aggregation function Eq. (6.9) and the evaluations of the compatibility criteria.
4. Compute the transitive closure of the overall compatibility matrix **S** and threshold with s^*.
5. Merge compatible clusters if Eq. (6.8) is satisfied.
6. Decrease K accordingly.

Until: No clusters can be merged.

Compatible cluster merging has a number of advantages compared to the simple clustering approach without merging. The cluster merging approach can reduce the computation time for modeling considerably, compared to the validity approach. First of all, the data need not be clustered for all $2 \leq K \leq K_m$, but only down to the relevant number of clusters. Secondly, more than two clusters can be merged at one step, reducing the computational load even further. In addition to the computational advantages, the CCM method may lead to better partitions due to better initialization of the clustering algorithm. In the absence of additional knowledge, the initial partition is selected randomly. In CCM, however, the initial partition for subsequent clustering stages is formed by merging compatible clusters. Hence, the information from a finer partition is used for the initialization, which can guide the clustering algorithm to better local optima. Moreover, when the data are grouped into a large number of clusters, interesting regions in which only a few data points are found can be located. By a careful merging process involving the heuristic step for preventing over-merging of clusters, the identified clusters in the interesting small regions can be preserved, while the clusters in other regions are merged. If the algorithm is initialized randomly, it is very unlikely that these regions may be located without increasing the number of clusters.

6.1.8 *Influence of the heuristic step*

The heuristic step prevents the over-merging of clusters that may result from the inadequate discrimination power of Eq. (6.3) and Eq. (6.4). To study the influence of the heuristic step, its influence is observed in the same experiments as in the feed-batch fermentor example in Sec. 6.1.6 using the same data set. Figure 6.7 shows the number of clusters that the cluster merging algorithm with the heuristic step finds for different values of parameter γ and the threshold s^*. Comparing Fig. 6.6 with Fig. 6.7, it is seen that the heuristic step has an influence, especially for large values of γ, and small values of s^* which lead to over-merging. When the heuristic step is active, it prevents merging in this region. When γ is about 0.5 and s^* is about 0.5, the heuristic step has little influence, and the algorithm finds three as the optimal number of clusters. Therefore, the heuristic step is especially used when there is a danger of over-merging (*e.g.*for high values of γ). However, the heuristic step also influences the response for medium values of γ ($0 < \gamma < 2$) increasing the sensitivity to the parameter γ, and hence the tuning of the threshold s^* requires more effort.

6.1.9 *Example*

Consider a system that is described by the static function

$$y = \sin(0.0015x^2)\frac{x^{2.9}}{10000}, \quad x \in [0, 100]. \tag{6.10}$$

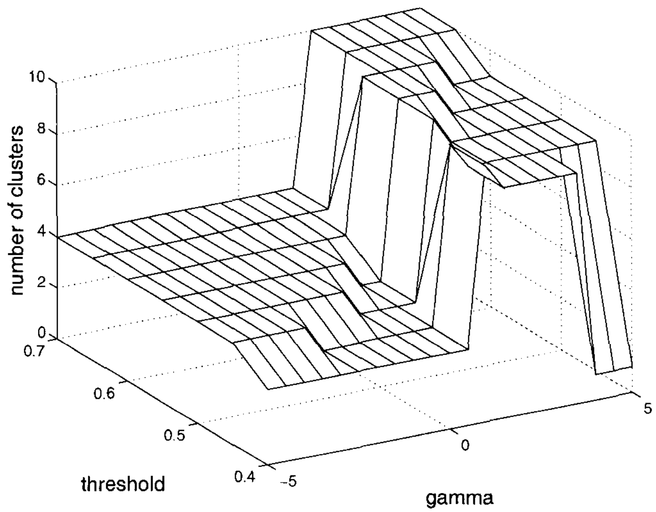


Fig. 6.7 The number of clusters determined by the CCM algorithm with the heuristic step for modeling a feed-batch fermentor. The results are depicted for different averaging operators and different similarity thresholds.

Data is generated for $x = 1, 2, \ldots, 100$. Uniformly distributed random noise with an amplitude of 2 is added to the function values. A TS–model is generated for this system by using the GK clustering algorithm. The CCM algorithm is applied to this system with $s^* = 0.5$. Starting with 10 clusters initially, 3 clusters are merged in the first step, 2 in the second and 2 in the third, resulting in six clusters finally. The results of clustering with 10 randomly initialized clusters are shown in Fig. 6.8. As expected, the clustering algorithm has distributed the clusters evenly across the input x. Figure 6.9 shows the local models and the clusters that are determined by the compatible cluster merging. As Fig. 6.9 shows, the algorithm has located six clusters that give good local descriptions of the system. Note the large cluster on the left and the small cluster on the right, which would not have been found if the clustering algorithm had been initialized randomly.

6.1.10 *Similarity and rule base simplification*

The evaluation of compatible clusters can be interpreted as the evaluation of similar clusters. The compatible cluster merging algorithm reduces the redundancy in a partition by combining similar clusters. The decision making step relates to the evaluation of the similarity. In general, similarity can be evaluated using one of the two methods.

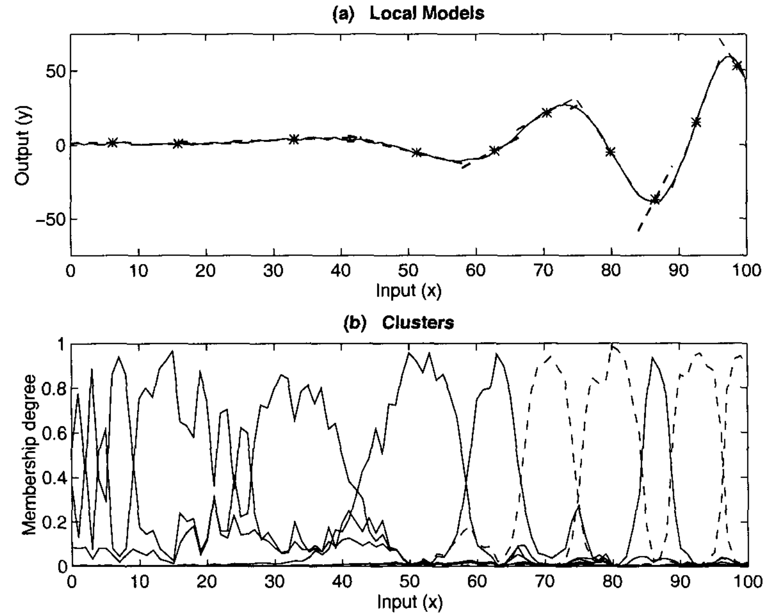


Fig. 6.8 The local models (a) and the clusters (b) that are identified by the GK-clustering algorithm after randomly initializing 10 clusters. Reproduced from (Kaymak and Babuška 1995),©1995 IEEE.

(1) By using explicit similarity measures.

(2) By using multicriteria evaluation of features that relate to similarity.

Explicit similarity measures typically assess the similarity between two fuzzy sets based on the (pointwise) similarity of their membership. Setnes, Babuška, Kaymak and van Nauta Lemke (1998) have proposed simplifying rule based systems by evaluating the similarity of membership functions with explicit similarity measures. The authors have applied the method to the modeling of a washing process and observed encouraging results. The advantage of simplifying fuzzy models by assessing the similarity of fuzzy sets is that the redundant information in each variable can be reduced, which also increases the interpretability of the fuzzy model. However, the combination of the reduced number of one-dimensional fuzzy sets by the inference mechanism may lead to results which are not supported by the data. Then, model validation techniques determine the applicability of the model determined.

Kaymak and Setnes (2000) have also proposed a cluster merging approach based on explicit similarity measures applied during the optimization stage of the clustering algorithm. The advantage of this method is that it can be applied to any objective function based clustering algorithm (*e.g.*fuzzy c–means), and one need

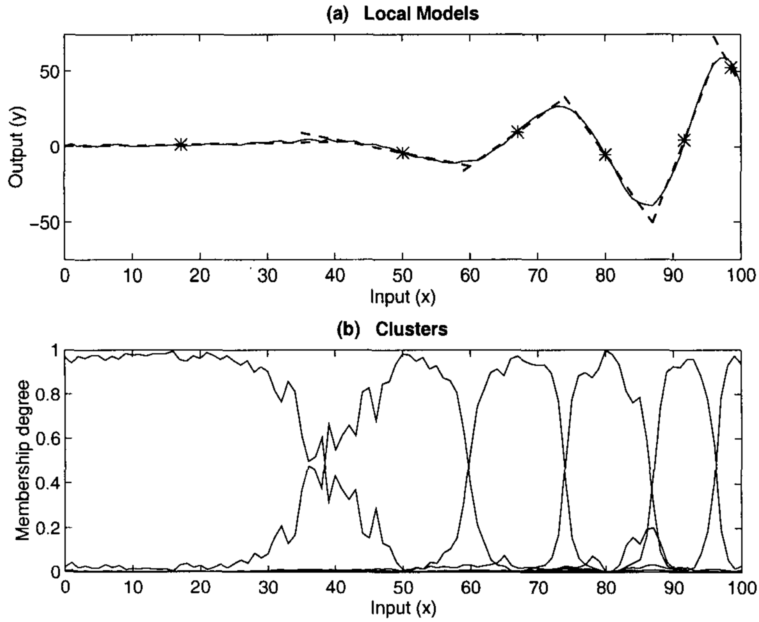


Fig. 6.9 The local models (a) and the clusters (b) that are identified by the *Compatible Cluster Merging* algorithm. Reproduced from (Kaymak and Babuška 1995), ©1995 IEEE.

not try to identify explicit criteria to assess similarity. In the future, the use of explicit similarity measures in fuzzy clustering can be expected to increase.

The compatible cluster merging can be categorized under multicriteria evaluation of similarity. In this approach, the similarity is formulated as a decision problem, and it is assessed by evaluating different features that are important for the similarity of the objects considered. The overall similarity is then obtained by aggregating the individual similarity criteria. Bonissone (1979) has also proposed such an approach for similarity assessment. It can be expected that the compatible cluster merging benefits from a similar approach, where an extension of the similarity criteria leads to improved merging behavior.

6.2 Defuzzification as a fuzzy decision

The reasoning mechanism of fuzzy systems manipulates fuzzy data, information and knowledge to determine its outputs. When fuzzy systems must interact with other (non-fuzzy) systems, or when a crisp output is required from the fuzzy system, the processed fuzzy information must be defuzzified to determine the crisp equivalent. This process is called defuzzification. The defuzzification may take place at different stages for different reasoning mechanisms. Sometimes, the fuzzy system propagates the fuzziness all the way to its outputs, such as a Mamdani type fuzzy rule based system. The defuzzification then takes place at the outputs of the fuzzy system. Other reasoning mechanisms defuzzify the results at an earlier stage. For instance, Takagi–Sugeno systems with constant rule consequents are essentially fuzzy rule based systems where the rule outputs are defuzzified at an early stage of the reasoning, before the consequent modification of the rules takes place.

Defuzzification reduces the fuzzy systems into nonlinear crisp mappings from the system inputs to the system outputs. Often, the defuzzification performs an interpolation, although this is not always the case. The selection of the defuzzification method is not context independent. The goals regarding the problem for which the fuzzy system is designed determine the selection of the defuzzification operation. When designing fuzzy PID controllers, for example, the defuzzification operation must perform some kind of interpolation so that the control signal is an intermediate value between different rule outputs when multiple rules are activated. For a mobile robot's path planner, however, a defuzzification method that returns an element with maximal membership (height of the membership function) may be more suitable in case two different rules follow different strategies while trying to avoid an obstacle (*e.g.*turn left and turn right). Therefore, the problem context has a direct influence on the selection of the defuzzification operation.

Let B be a fuzzy set defined by the membership function $\mu : y \in Y \longrightarrow [0,1]$. The two most commonly used defuzzification methods in the literature are the center-of-gravity (cog) defuzzification (Driankov et al. 1993)

$$y^{\text{cog}} = \frac{\int_Y y \, \mu_B(y) dy}{\int_Y \mu_B(y) dy}, \tag{6.11}$$

and the mean-of-maxima (mom) defuzzification

$$y^{\text{mom}} = \frac{y_{\min} + y_{\max}}{2}, \tag{6.12}$$

where $y_{\min}$ and $y_{\max}$ are given by

$$y_{\min} = \inf_y \{y \in Y : \mu_B(y) = \text{height}(B)\} \tag{6.13}$$

$$y_{\max} = \sup_y \{y \in Y : \mu_B(y) = \text{height}(B)\}. \tag{6.14}$$

The defuzzification operators in Eq. (6.11) and Eq. (6.12) basically determine the first moment of the fuzzy set where the domain elements are weighted by the membership functions. More extended and generalized defuzzification methods have also been proposed in the literature (Filev and Yager 1991, Runkler and

Glesner 1993, Yager and Filev 1993). For example, Filev and Yager (1991) have proposed a parametric class of defuzzification operators known as BADD (*basic defuzzification distributions*) operators and given by

$$y^{\mathrm{BADD}}(\delta) = \frac{\int_Y y\,[\mu_B(y)]^\delta dy}{\int_Y [\mu_B(y)]^\delta dy}, \quad \delta \geq 0, \tag{6.15}$$

where the parameter δ can be used to adapt the defuzzification operator to specific problems. The extension of the method arises from the availability of δ to the user for influencing the way the domain elements in Y are weighted by the membership values. Following the same line of thought, Yager and Filev (1993) have introduced the SLIDE (*semi-linear defuzzification*) operators given by

$$y^{\mathrm{SLIDE}}(\delta_1, \delta_2) = \frac{(1-\delta_2)\int_{Y_L} y\,\mu_B(y)dy + \delta_2 \int_{Y_H} y\,\mu_B(y)dy}{(1-\delta_2)\int_{Y_L} \mu_B(y)dy + \delta_2 \int_{Y_H} \mu_B(y)dy} \tag{6.16}$$

with $\delta_1 \in [0, \mathrm{height}(B)]$, $\delta_2 \in [0,1]$ and

$$Y_L = \{y \in Y : \mu_B(y) < \delta_1\} \tag{6.17}$$

$$Y_H = \{y \in Y : \mu_B(y) \geq \delta_1\}. \tag{6.18}$$

Most popular defuzzification methods have been developed for applications in control engineering and they reflect the interpolative characteristics of control problems. However, other fields of application or the way information is used by certain fuzzy systems may require other types of defuzzification operators. Zimmermann (1996) proposes four criteria for selecting the defuzzification operator in control problems,

(1) computational effort,
(2) representation of objective,
(3) continuity,
(4) plausibility.

In a more general decision making setting, especially criterion 2 and criterion 4 are important for selecting the defuzzification operator. The consideration of these criteria in particular problems may lead to new defuzzification operators as discussed in Sec. 6.2.2.

The defuzzification operators that have been proposed in literature do not differentiate amongst different elements of the domain on which the fuzzy set is defined. There is not any preference for one domain element over another and the defuzzification is mainly influenced by the corresponding membership values of the domain elements. In this sense, the defuzzification operator is equally sensitive to all the domain elements. Some applications, however, may require a method which is not equally sensitive to all the domain elements. This section discusses a new defuzzification method with unequal sensitivity to the elements of the domain over which the defuzzification takes place. The proposed method is developed for a fuzzy security assessment system for power distribution networks and has been applied successfully (Kaymak, Babuška, van Nauta Lemke and Honderd 1998).

6.2.1 *Sensitivity of defuzzification to domain elements*

A defuzzification method can be interpreted as an aggregation operation that replaces a fuzzy set by a representative crisp value. Since most implementations of defuzzification operators discretize the continuous domains, only the quantized versions of the defuzzification operators are considered in the following. Consider a fuzzy set discretized into N_q elements as shown in Fig. 6.10. The fuzzy set B is determined by the ordered pairs $(y_q, \mu_B(y_q))$, $q = 1, \ldots, N_q$. Defuzzification is the determination of the best crisp value that represents $\mu_B(y)$ given the ordered pairs. Hence, the problem is similar to the selection of a best alternative given the data regarding the satisfaction of the decision criteria. However, it is possible that the defuzzified value is not a member of the set of alternatives, and therefore the defuzzification problem is more of an aggregation problem rather than a choice problem. The defuzzification operator aggregates the values $(y_q, \mu_B(y_q))$, $q = 1, \ldots, N_q$ into a final defuzzified value y^*, and it can be expected that modified versions of fuzzy aggregation operators can be used for this purpose.

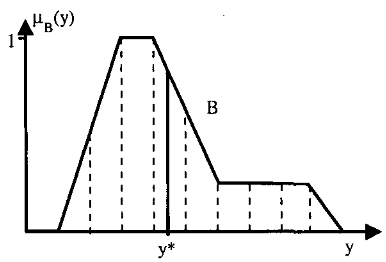


Fig. 6.10 Defuzzification of a discretized fuzzy set.

A typical defuzzification operator combines the values for the domain elements with the (modified) membership values for obtaining the defuzzified value. In general, the membership function is used as a distribution function for nonlinear weighting of the domain elements. Different domain elements with equal membership value have equal influence on the defuzzification operation. In this sense, the defuzzification method is equally sensitive to different domain elements.

Definition 6.1 Let Y be a discretized domain, *i.e.*, $Y = \{y_q | q = 1, \ldots, N_q\}$ and B a fuzzy set that is defined on Y. A defuzzification operator Z is said to be equally sensitive to the domain elements y_q if it satisfies

$$\left.\frac{\partial Z}{\partial y_q}\right|_{\mu_B(y_q)=\mu^*} = \left.\frac{\partial Z}{\partial y_{q'}}\right|_{\mu_B(y_{q'})=\mu^*}, \qquad \mu^* \in [0,1], \tag{6.19}$$

for all q, q' with $\mu_B(y_q) = \mu_B(y_{q'})$.

Definition 6.1 states that the contribution of element y_q towards the defuzzified value is independent of the value of y_q and that the influence on the defuzzification operation of different domain elements with equal membership is equal. One consequence of Definition 6.1 is that the defuzzification of a crisp set in $\mathbb{R}$ (*i.e.*, a crisp interval) is equal to the mid-point of the interval when the defuzzification operator is equally sensitive to its domain elements. If the defuzzification operator is not equally sensitive to the domain elements, the defuzzified value may be another element in the domain.

The center-of-gravity defuzzification of a fuzzy set B is given in the discretized case by

$$Z^{\text{cog}}(B) = \frac{\sum_{q=1}^{N_q} \mu_B(y_q)\, y_q}{\sum_{q=1}^{N_q} \mu_B(y_q)} \tag{6.20}$$

with N_q the cardinality of the domain on which B is defined. Substituting Eq. (6.20) in Eq. (6.19) shows that the condition in Eq. (6.19) is satisfied. Hence the center-of-gravity defuzzification is equally sensitive to domain elements. Similarly, it can be found for the mean-of-maxima defuzzification that

$$\frac{\partial Z^{\text{mom}}}{\partial y_q} = \begin{cases} \frac{1}{N_m}, & \mu_B(y_q) = \text{height}(B) \\ 0 & \text{otherwise,} \end{cases} \tag{6.21}$$

where N_m is the number of domain elements with maximal membership.

The generalizations and extensions to the existing defuzzification methods have been suggested in literature such as the BADD defuzzification method from Filev and Yager (1991), the SLIDE method from Yager and Filev (1993) and the XCOA method from Runkler and Glesner (1993). These methods differ in the way the membership values are used for defuzzification, but they remain equally sensitive to the domain elements, which need not be suitable for all applications.

6.2.2 *A defuzzification method with unequal sensitivity*

This section discusses a new defuzzification operator that has unequal sensitivity to the domain elements. The new defuzzification operator that will be called cogus (center-of-gravity with unequal sensitivity) is based on the generalized averaging operator in Eq. (3.24). It is described by

$$Z^{\text{cogus}}(B) = \left(\frac{\sum_{q=1}^{N_q} w_q \left[\mu_B(y_q)\right]^{\delta} y_q^{\gamma}}{\sum_{q=1}^{N_q} w_q \left[\mu_B(y_q)\right]^{\delta}} \right)^{1/\gamma}, \qquad \gamma \in \mathbb{R},\ \delta > 0, \tag{6.22}$$

with $w_q > 0$, $q = 1, \ldots, N_q$. The weight factors w_q form one way of introducing unequal sensitivity to domain elements. They influence the relative contribution of each domain element to the defuzzified value. The other method for introducing unequal sensitivity to the domain elements is the use of the parameter γ. For negative values of γ, the result of the defuzzification moves towards the smallest element in the support of the defuzzified set. For positive values of γ, the defuzzified value moves towards the largest element in the support. The operator δ changes the sensitivity to membership values. When δ increases, the influence of elements with high membership value increases on the defuzzification operation. The influence of the elements with lower membership value increases as δ approaches zero.

The proposed defuzzification method is similar to the BADD defuzzification method except for the introduction of the weight factors w_q and the parameter γ. Evaluating Eq. (6.19) with Eq. (6.22), one finds that the influence of the domain elements on the defuzzification is different because of the introduction of these parameters. Hence, the result is a defuzzification operator that is unequally sensitive to the domain elements. It should be mentioned that the transformation of y_q must be allowed, which means that Eq. (6.22) cannot be used for domains with negative elements. In this case, the domain must be transformed to non-negative values, and the defuzzified value must then be transformed back to obtain the defuzzified value on the original scale.

6.3 Application example: fuzzy security assessment

This section describes an application for which the output of the fuzzy system needs to be defuzzified by a method that is unequally sensitive to the domain elements. The application is related to the operation of a power distribution network. An important task in the operation of power distribution networks is the assessment of the network's security with the goal of determining which actions to take to control the process. The security assessment involves classifying the system's state into one of the security classes, based on the predictions of its behavior after the failure of a component in the network. For this purpose, extensive power-flow simulations of the network are done, the results of which are presented to the operators who interpret them for determining the security class. The operators are expected to take proper actions if the simulations predict violations of the safety margins. However, a lot of ambiguity still exists in determining the security class since not all the relevant data are available, the severity of the violations is different, the effects of the violations are open to interpretation and the goals of the system are usually expressed in linguistic terms. Hence, a fuzzy logic based security assessment system presents advantages as it can deal with the uncertainty in the system. Moreover, the fuzzy logic system can circumvent extensive simulations — which take a long time and require powerful, usually specialized hardware — by using the experience of the network.

6.3.1 *Security class determination*

A fuzzy knowledge-based system has been developed for assessing the security of the Dutch 380 kV distribution network (Heydeman et al. 1996, Kaymak, Babuška, van Nauta Lemke and Honderd 1998). For assessing the severity of possible violations, the security assessment system divides the security of the transmission network into four distinct classes,

(1) very secure (VS),
(2) normal secure (NS),
(3) slightly insecure (SI),
(4) very insecure (VI).

The security of the network decreases from VS to VI. Figure 6.11 shows how the information is used to determine the security class. The security class is determined first at the level of components such as transformers, transmission lines and generators in the network. The results are then aggregated at the network level. The inference at the component level determines the security class of the network when considering only one type of component in the network. Each rule base thus determines a separate outcome for the security class. At this level, max–min composition with the Mamdani minimum operator (Driankov et al. 1993) is used for inference. The output of the inference is a fuzzy set with four elements, where the membership value of each element denotes the membership of the network to a particular security class. The resulting fuzzy classifications (outputs of each rule base) are then combined at the network level for arriving at an overall classification of the network security. In general, this aggregation can be done by using one of the many decision functions available from the fuzzy decision theory.

The selection of the operator should comply with the goals and the properties of the particular decision problem. The maximum operator is a suitable aggregation operator for security analysis, since there is not a specific preference for one of the rule bases and this operator allows the analysis of a variety of security classifications, including the extreme considerations such as the worst-case analysis.

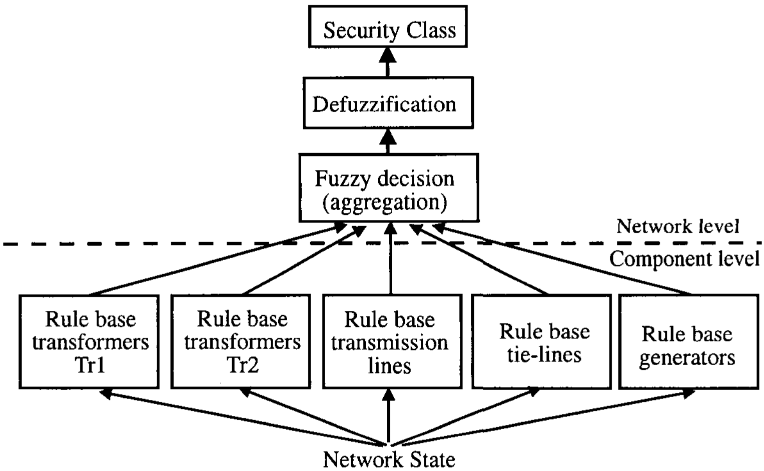


Fig. 6.11 Determination of the security class. Reprinted from (Kaymak, Babuška, van Nauta Lemke and Honderd 1998) by permission of IOS Press, ©1998 IOS Press.

The result of the aggregation is an overall fuzzy security class, which is represented by a fuzzy set that is defined on the discrete space $\{1, 2, 3, 4\}$ with VS corresponding to 1 and VI corresponding to 4. This fuzzy set needs to be defuzzified to determine the security class of the network. Figure 6.12 shows a fuzzy security classification that is obtained from the fuzzy security analysis system. A couple of points are important for the defuzzification of this fuzzy set. First, the contribution of a security class to the overall security is not the same. The insecure classes SI and VI are much more important in determining the overall security class than the secure classes VS and NS and should have more influence on the defuzzification. Secondly, because of the way information is represented in the system and the way it is used, the membership of the secure classes will be high in general. Indeed, most of the components in the network are almost always in the secure classes and thus the membership of the secure classes is high. A defuzzification method should thus exhibit lower sensitivity to the secure classes so that the influence of these high membership values is limited. The next section describes a defuzzification method with unequal sensitivity to domain elements when Eq. (6.22) is applied to the fuzzy security assessment system.

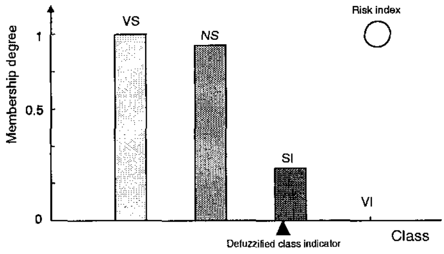
6.3.2 *Defuzzification for fuzzy security assessment*

The overall security class of the power network is determined after the defuzzification of the fuzzy security class. Classical defuzzification methods such as the center-of-gravity cannot be used because of the way the fuzzy security class is determined. As mentioned in Sec. 6.3, the influence of the insecure classes on the defuzzification should be larger. Also, the membership values for the classes VS and NS will usually be high (close to 1) and a direct averaging amongst the domain elements biases the defuzzification towards the more secure classes. One way of decreasing this bias is the introduction of an unequal weighting of the domain elements (different security classes). Another method is the introduction of a parameter which increases the influence of the insecure classes on the defuzzification. A new defuzzification method that uses both these methods is obtained as a special case of Eq. (6.22),

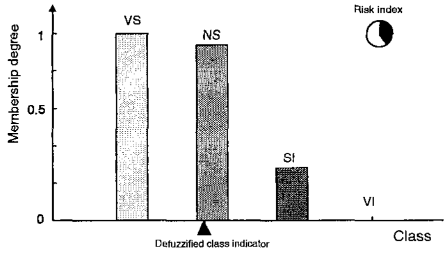
$$y^* = \left(\frac{\sum_{q=1}^{4} w_q \mu_q y_q^{\gamma}}{\sum_{q=1}^{4} w_q \mu_q} \right)^{1/\gamma}, \qquad \gamma \in \mathbb{R} \tag{6.23}$$

where $y^* \in [1,4]$ is the defuzzified value (class indicator) related to the security class of the network and $y_q \in \{1,2,3,4\}$ correspond to the security classes VS, NS, SI and VI respectively. μ_q indicate the membership degree that the security of the network is classified as class y_q, and w_q are the weight factors related to the importance of the particular class for the defuzzification purpose. In general, the weights of the secure states are lower since more attention needs to be paid to the components that result in an insecure class. Notice that the contribution of a particular class towards the indicator y^* is determined both by the membership to that class and the weight of the class.

The value of the class indicator (defuzzified value) can be moved towards or away from the largest element in the support of the fuzzy security class by varying the value of the parameter γ. This parameter can be interpreted as an index of risk awareness (van Nauta Lemke et al. 1983). For negative values of γ, the influence of the secure states on the defuzzification increases. The human operator then takes less corrective action, which corresponds to more economic operation at the expense of taking more risk. For positive values of γ, the influence of the insecure states on the defuzzification increases. This corresponds to an increased awareness for risk aversion and very secure operation, at the expense of increased operating costs. Therefore, by changing the value of γ, the sensitivity of the defuzzification to the insecure classes can be varied. Figure 6.12a shows the security class for a particular state of the network when the operation is completely risk-aware. Since this is a worst-case situation, the risk index is zero. The final classification corresponds to SI. There are no components in the VI class. Figure 6.12b shows the network in the same state when the risk awareness of the operation is reduced. The final security class now becomes NS. Since the network is classified as NS (while some components in the network are classified in SI), some risk is being taken, as indicated by the black portion of the risk index.



(a) $\gamma \to \infty$



(b) $\gamma = 0$

Fig. 6.12 Fuzzy security classification with (a) complete risk avoidance, and (b) reduced risk awareness. Reprinted from (Kaymak, Babuška, van Nauta Lemke and Honderd 1998) by permission of IOS Press, ©1998 IOS Press.

6.4 Summary and concluding remarks

Fuzzy decision making methods can be applied to support the identification and construction of fuzzy inference systems. In fuzzy modeling, one needs to determine the values of the parameters in various algorithms. Fuzzy decision making can be used to automate these selection procedures. A compatible cluster merging algorithm is described in which the required number of clusters for describing a system is described. The clustering algorithm can be used to obtain a fuzzy model of a system. Since the number of rules in the fuzzy rule base is determined by the number of clusters, the decision algorithm leads to the determination of an 'optimal' number of rules for describing the system. In this way, fuzzy modeling can be automated to a large degree, as the interaction from the user can be limited to a single parameter.

Many operations in fuzzy systems theory can be formulated as a decision making algorithm. The formulation of the defuzzification operation for fuzzy systems as a decision problem leads to new insights about defuzzification. Using this insight, a new defuzzification method (cogus defuzzification) has been introduced. Cogus defuzzification is unequally sensitive to the domain elements of the defuzzified fuzzy set. An example of fuzzy security assessment is included in this chapter to show an application of this defuzzification method.

This page is intentionally left blank

Chapter 7

Fuzzy Model-Based Control

In this chapter, we study various fuzzy model-based control schemes, where fuzzy process models are used in different model-based controllers. Several approaches for designing a controller based on a fuzzy model of the process have been investigated by various authors. Braae and Rutherford (1979) derived a fuzzy controller based on a linguistic fuzzy model. The technique suffers from a major limitation, in that the model could not deal directly with the linguistic aspects of the FLC. Pedrycz (1993) has investigated methods for deriving a control law using fuzzy relational models. Off-line controllers are synthesized based on one-step ahead prediction of the corresponding local fuzzy models. These methods are not applied in this book, because fuzzy relational models are computationally more complex than linguistic or TS fuzzy models, implying larger computational effort, and loss of linguistic meaning for the fuzzy rules in the model. An adaptive fuzzy controller based on fuzzy relational models is applied in Graham and Newell (1988) to a laboratory-scale liquid level rig. Driankov et al. (1993) present another example of the application of model-based control methodologies, where local design techniques derived from linear control theory have been applied to Takagi–Sugeno models with linear consequents (Kuipers and Åström 1994, Sugeno and Takagi 1983, Tanaka and Sugeno 1992, Palm et al. 1997).

The simplest way to control a process by using a fuzzy model is to invert the model and use it in an open-loop (feedforward) configuration. The obtained inverse model is used as a controller, and under special conditions stable control can be guaranteed for minimum phase systems. This type of control can only be applied if the inverse of a fuzzy model exists. If this inversion is not unique, some additional criteria most be added to the controller in order to choose the best control action at a given moment. Since this is a feedforward configuration, this 'ideal' control configuration can not be directly applied in practice because the model is never a perfect mapping of the system, *i.e.*, model–plant mismatches are present. Moreover, the system must cope with disturbances, and some variables of the process (more often the control actions) can be subject to level and/or rate constraints. Therefore, if one wants to implement a controller based on the inversion of a fuzzy model, the inversion must exist, some criteria must be added to choose a control action if more than one is obtained by the inversion, and the problems of model–plant mismatch, influence of disturbances and constraints must be overcome.

This chapter presents an approach that benefits from the convenient mathematical structure of certain types of rule-based fuzzy models to invert them. First, the problems related to the inversion of fuzzy models, and their use in control schemes are presented in Sec. 7.1. This section also considers an adaptive internal model control scheme, where the fuzzy model is adapted to deal with model–plant mismatch. In order to cope with constraints, this inversion can be used for control purposes, when combined with a predictive control structure. This scheme can prevent overshoots, and reduce rise and settling times. The combination of inverse model control with a predictive control structure is given in Section 7.5. A simulation example of a fermentor covering all the proposed control schemes is presented in Sec. 7.6. This example presents the application of inverse model control and its combination with predictive control to semi-realistic problems. Inverse model control based on TS fuzzy models is applied in real-time control to an air-conditioning system presented in Chapter 12. Sometimes, model-based control techniques cannot eliminate steady-state errors due to model–plant mismatch, or due to disturbances, depending on the number of integrators in the process, the type of disturbances, *e.g.*offset or process disturbances, and the required reference tracking accuracy. Internal model-based control could be used to deal with this problem, but it may lead to sluggish response due to the linear filter in the scheme (see Appendix B). We present in Sec. 7.7 a fuzzy compensation scheme that can quickly eliminate the steady-state errors under certain circumstances. The chapter ends with concluding remarks in Sec. 7.8.

7.1 Inversion of fuzzy models

The simplest way to control a process, when an inverse model is available, is to use this inverse model in an open-loop configuration. Considering an ideal model $\mathbf{M}$ mapping the control actions $\mathbf{u}$ to the system's outputs $\mathbf{y}$, the control actions are simply given by $\mathbf{u} = \mathbf{M}^{-1}\mathbf{r}$, where $\mathbf{r}$ are the references to be followed (see Fig. 7.1).

If an ideal model of the process is available, *i.e.*, the model is equal to the process, and both model and controller (inverse model) are input–output stable, the control is *perfect*, and input–output stable (Economou et al. 1986). This situ-

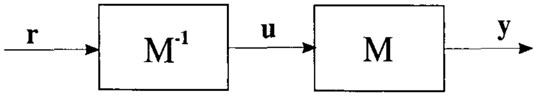


Fig. 7.1 Mapping and perfect inversion of a system.

ation of perfect control is not possible to achieve, since an exact inversion of the process can only be found in special situations, and the model is never equal to the process, resulting in model–plant mismatches. Moreover, the variables of the process can be subjected to level and rate constraints. Furthermore, disturbances acting on the process are present and these are not taken into account in the controller. Moreover, when the system has a delay of d steps, the inversion must be done for d steps ahead. All these problems must be overcome, in order to apply inverse model control in practice. The problems related to model–plant mismatch and disturbances are often dealt with the internal model control (IMC) scheme summarized in Appendix B.

Fuzzy modeling is often used in the identification of the process dynamics in order to cope with nonlinear and complex systems, giving good approximations of nonlinear systems (see Chapter 5). Moreover, special types of fuzzy models can be analytically inverted, and used for control purposes. The definition of an inverse fuzzy model is discussed in Sec. 7.1.1. The most common methods of inverting fuzzy models are presented in Sec. 7.1.2. Fuzzy models with certain structures can be *exactly* inverted, and this inversion can be used for control purposes. This book considers two different fuzzy model structures for which the exact inversion of the model can be achieved.

(1) Singleton fuzzy models, for which the inversion is presented in Sec. 7.2. This type of models belongs to a general class of function approximators (Friedman 1991), which is at least as accurate as a linguistic fuzzy model.
(2) Takagi–Sugeno fuzzy models with affine inputs $\mathbf{u}(\tau)$, whose inversion is described in Sec. 7.3. Constraining the model to be affine on $\mathbf{u}(\tau)$ usually reduces the model accuracy.

Both inversions are computationally very fast, and hence they can be used in systems with small sampling times. The system under control is sometimes time-variant and changes in the process parameters can occur. Moreover, significant model–plant mismatches due to permanent or temporary changes in the operating conditions are frequent in industrial processes. For this type of systems, the model can be adapted on-line in order to cope with these phenomena. An adaptation algorithm based on recursive least-squares is presented in Sec. 7.4, where the singleton fuzzy model is adapted. If the adaptation is done such that the invertibility of the model remains valid, this scheme can be used for control purposes.

7.1.1 *Problem definition*

Consider a global MIMO fuzzy model. For the structure presented in Sec. 5.1, any MIMO model can be decomposed in p MISO models without lack of generality. The MISO fuzzy system with n states given by Eq. (5.4) can be represented as in Fig. 7.2. Note that the state variables x_j in Eq. (5.4) are for input–output models, and they can generally be different inputs or outputs at delayed times of the process.

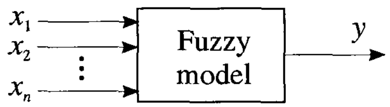
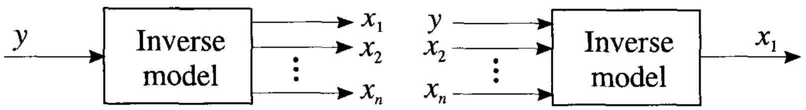


Fig. 7.2 General MISO fuzzy model.

There are two ways of inverting this fuzzy model.

(1) *Global inversion* of the model, where all states become outputs of the inverted model, and the output of the original model becomes the state of the inverted model, as presented in Fig. 7.3a. Thus, this inversion computes all the state variables when the original output is given. The solution of this inversion is normally not unique, and it is given by a family of solutions.
(2) *Partial inversion* of the model; only one of the states of the original model becomes an output of the inverted model and the other states together with the original output are the inputs of the inverted model (see the example in Fig. 7.3b).



(a) Global inversion. (b) Partial inversion.

Fig. 7.3 Inversion of fuzzy models.

The partial inversion usually has a unique solution, which is a big advantage compared to the global inversion. For the partial inversion, the inverted state is called a *controllable* input, and it is one of the inputs u_j, $j \in \{1, \ldots, m\}$ of the original system. Let this input be called $u_1 \triangleq x_1 \triangleq u$, to simplify the notation. The other states that are not inverted are called *non-controllable*, even if they are inputs of the system that can be manipulated by the process operator. When several control actions are needed at the same time, this method cannot be directly applied, but it can give the several input actions independently, by applying the partial inversion for each state that corresponds to an input of the system. The desired control action can be found by optimizing a given criterion, as for instance, in predictive control. The results obtained from the partial inversions can be given as initial values for the optimization algorithm to be applied. Only partial inversion is used in this book. For the sake of simplicity, *partial inversion* is often simply called *inversion*.

Note that partial inversion can only be applied when the inverse of the considered fuzzy model exists. If this inversion is not unique, some additional criteria must be added to find the best solution. When the inverted model is used as a controller, these criteria must determine the best control action. A fuzzy model is invertible if the model is represented by a function

$$y = f(u, x_2, \ldots, x_n), \tag{7.1}$$

and the inverted function exists, such that

$$u = f^{-1}(y, x_2, \ldots, x_n). \tag{7.2}$$

This statement implies that the function describing the original fuzzy model must be strictly monotone with respect to u. The translation of the invertibility conditions for the singleton fuzzy model and for the affine TS fuzzy model are discussed in Sec. 7.2 and Sec. 7.3, respectively.

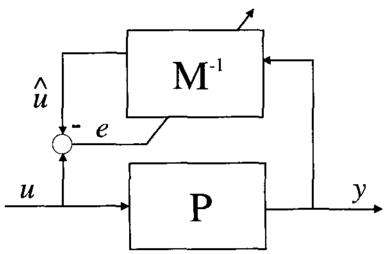
7.1.2 *Inversion methods*

Several methods can be applied to obtain the inverse model of a given process (Boullart et al. 1992, Hunt et al. 1992). The following two are used the most:

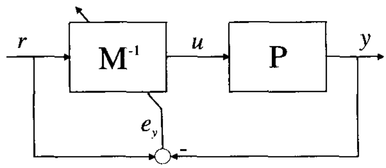
(1) Identification of the inverse model from input–output data
(2) Inversion of the original model

The first method is perhaps the most intuitive approach to inverse modeling, and it tries to fit the data in an inverse function f^{-1} (Batur et al. 1993). Two major approaches can be distinguished in this approach: *direct inverse learning* and *specialized inverse learning* (Fischer et al. 1998).

In direct inverse learning, the process is excited with a training signal and the fuzzy system reconstructs the input signal of the process from the given output signal, see Fig. 7.4a. Different identification algorithms can be used to derive



(a) Direct inverse learning.



(b) Specialized inverse learning.

Fig. 7.4 Inverse learning.

the inverse model, such as the one developed by Nelles (1998). Following Hunt et al. (1992), two major drawbacks can be found in direct inversion. First, the dynamics of the system can be a many-to-one mapping, and several values for u are possible for the same output of the process. If a least-squares approach is used, the identification algorithm maps y to the mean value of all the u, which can lead to a meaningless inverse model. Secondly, it is difficult to obtain an appropriate training signal for direct inverse learning, because the inverse model is supposed to work over a wide range of input amplitudes on y and for a large bandwidth. However, the excitation of the system is introduced as the activation of u, and a persistent excitation of y can not be guaranteed.

Both drawbacks of direct inverse learning can be overcome by using specialized inverse learning, see *e.g.*Jordan and Rumelhart (1992). The inverse model is cascaded with the process, as in Fig. 7.4b, or with a forward plant model. The parameters of the inverse model $\mathbf{M}^{-1}$ are adapted in order to minimize the deviation between the reference r and the output y. Thus, the adaptation is a goal-oriented scheme, since the objective is the same as the general control goal, and the process is automatically excited with the right signal if a typical reference trajectory must be followed. Moreover, level and rate constraints can also be considered in the learning phase.

Although specialized inverse learning overcomes the problems of excitation and possible non-invertibility, it is still difficult to use this inverse model in a control scheme, due to the model–plant mismatch and the influence of disturbances. A scheme as a disturbance observer developed by Fischer et al. (1998) can be implemented, but this scheme needs some parameter tuning, and uses a linearization of the inverse model at a certain point. Therefore, an exact inversion of the nonlinear fuzzy model is not obtained. Another possibility is to invert a feedforward fuzzy model numerically, when it is invertible, *i.e.*, when a unique mapping from the output to the inputs of the process is possible to obtain. The inverted model can be obtained with a desired accuracy, depending on the chosen number of discretized points. However, even for a small number of points, the computational costs are too high, and this solution cannot be considered as a feasible one. Therefore, the best solution seems to be to invert a fuzzy model *exactly*, by using some analytical method. If this inversion is possible, the computational operations can be done by using standard matrix operations and linear interpolations, apart from the computation of the degrees of fulfillment. Thus, the inversion is computationally very fast, making it suitable for applications in real-time control. Another advantage is that both the model and its inversion are available, allowing their use in the nonlinear internal model control scheme presented in Sec. B.1.

7.2 Inversion of a singleton fuzzy model

The inversion of singleton fuzzy models was introduced in Babuška et al. (1995). A special structure of the singleton fuzzy model, which is presented in this section, is necessary to perform this inversion.

7.2.1 *Linguistic fuzzy models with singleton consequents*

Assume that a SISO singleton model of the process is available. Such a model can be constructed directly from process measurements. A general fuzzy rule R^k has the following form.

$$\begin{aligned} R^k : \textbf{If } & y(\tau) \text{ is } A_1^k \textbf{ and } \dots \textbf{ and } y(\tau - p_y + 1) \text{ is } A_{p_y}^k \\ & \textbf{and } u(\tau) \text{ is } B_1^k \textbf{ and } \dots \textbf{ and } u(\tau - m_u + 1) \text{ is } B_{m_u}^k \\ & \textbf{then } \hat{y}(\tau + 1) = c^k, \quad k = 1, 2, \dots, K, \end{aligned} \tag{7.3}$$

where $A_1^k, \dots, A_p^k$ and $B_1^k, \dots, B_m^k$ are fuzzy sets and c^k are singletons. p_y and m_u are the orders of the output and the input, respectively. To simplify the notation, the rule index and the subscript of the input and output orders will be omitted below. The considered fuzzy rule is then given by the following expression.

$$\begin{aligned} & \textbf{If } y(\tau) \text{ is } A_1 \textbf{ and } y(\tau - 1) \text{ is } A_2 \textbf{ and } \dots y(\tau - p + 1) \text{ is } A_p \\ & \textbf{and } u(k) \text{ is } B_1 \textbf{ and } u(\tau - 1) \text{ is } B_2 \textbf{ and } \dots u(\tau - m + 1) \text{ is } B_m \\ & \textbf{then } \hat{y}(\tau + 1) \text{ is } c\,. \end{aligned} \tag{7.4}$$

Let a state vector $\mathbf{x}(\tau)$ containing the $m - 1$ past inputs, the $p - 1$ past outputs and the current output, *i.e.*, all the antecedent variables in Eq. (7.4) except $u(\tau)$, be defined as

$$\mathbf{x}(\tau) = [y(\tau), \ldots, y(\tau - p + 1), u(\tau - 1), \ldots, u(\tau - m + 1)]^T . \tag{7.5}$$

Multidimensional fuzzy sets for $\mathbf{x}(\tau)$ are defined on $\mathbb{R}^{p+m-1}$, the Cartesian product of the individual universes of discourse. When conjunctive aggregation is used, the multidimensional fuzzy set A is given by

$$A = A_1 \circledast \cdots \circledast A_p \circledast B_2 \circledast \cdots \circledast B_m ,$$

where $\circledast$ represents a t-norm. By introducing the formal substitution of B_1 by U for notational clarity, the fuzzy rule in Eq. (7.4) can be written as

$$\textbf{If } \mathbf{x}(\tau) \text{ is } A \textbf{ and } u(\tau) \text{ is } U \textbf{ then } \hat{y}(\tau + 1) \text{ is } c . \tag{7.6}$$

Note that the rule base of Eq. (7.4) is equivalent to the rule base of Eq. (7.6), since the order of the model dynamics is the same, taking into account that $\mathbf{x}(\tau)$ is a vector and A is a multidimensional fuzzy set. Let N denote the number of different fuzzy sets A_i defined for the state $\mathbf{x}(\tau)$ and M the number of different fuzzy sets U_j defined for the input $u(\tau)$. If the rule base consists of all possible combinations of A_i and U_j (the rule base is complete), the total number of rules is $K = N \times M$. The entire rule base can be represented as a table

	$u(\tau)$			
$\mathbf{x}(\tau)$	U_1	U_2	$\ldots$	U_M
A_1	c_{11}	c_{12}	$\ldots$	c_{1M}
A_2	c_{21}	c_{22}	$\ldots$	c_{2M}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_N	c_{N1}	c_{N2}	$\ldots$	c_{NM}

(7.7)

The logical *and* connective is assumed to be represented by the product t-norm operator, because this is a necessary condition to perform the inversion, and the degree of fulfillment of the rule antecedent $\beta_{ij}(\tau)$ is calculated as

$$\beta_{ij}(\tau) = \mu_{A_i}(\mathbf{x}(\tau)) \cdot \mu_{U_j}(u(\tau)) , \tag{7.8}$$

where $\mu_{A_i}(\mathbf{x}(\tau))$ is the membership degree of a particular state $\mathbf{x}(\tau)$ in the fuzzy set A_i and $\mu_{U_j}(u(\tau))$ is the membership degree of an input $u(\tau)$ in the fuzzy set U_j.

The predicted output $\hat{y}(\tau+1)$ of the model is computed by the fuzzy-mean defuzzification, where an average of the consequents c_{ij} is weighted by the degrees of fulfillment β_{ij}, so that

$$\hat{y}(\tau+1) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{ij}(\tau)\,c_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\beta_{ij}(\tau)} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mu_{A_i}(\mathbf{x}(\tau))\,\mu_{U_j}(u(\tau))\,c_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M}\mu_{A_i}(\mathbf{x}(\tau))\,\mu_{U_j}(u(\tau))}\,. \tag{7.9}$$

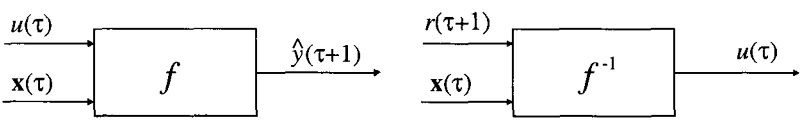
If this type of singleton fuzzy models also has triangular membership functions in the antecedents, and form a partition, *i.e.*, $\sum_{i=1}^{N}\mu_{A_i}(\mathbf{x}) = 1$, $\forall \mathbf{x}$, and $\sum_{j=1}^{M}\mu_{U_j}(u) = 1$, $\forall u$, the above singleton model provides piecewise linear interpolation between the rule consequents.

7.2.2 *Inversion of the singleton model*

The rule-based model of Eq. (7.6) corresponds to a nonlinear regression model

$$\hat{y}(\tau+1) = f(\mathbf{x}(\tau), u(\tau)), \tag{7.10}$$

shown schematically in Fig. 7.5a. The model inputs are the current state $\mathbf{x}(\tau)$ and the current input $u(\tau)$ and the output is the system's predicted output at the next sampling instant $\hat{y}(\tau+1)$.



(a) Model of the system. *(b) Derived controller.*

Fig. 7.5 Fuzzy model and a controller based on the model inverse.

Given the current system state $\mathbf{x}(\tau)$ and the desired system output (reference) at the next sampling time $r(\tau+1)$, the objective of the control algorithm is to find $u(\tau)$, such that the system output $y(\tau+1)$ is as close as possible to the desired output $r(\tau+1)$. This can be achieved by inverting the plant model, as indicated in Fig. 7.5b, substituting the reference $r(\tau+1)$ for $\hat{y}(\tau+1)$ in the static function

$$u(\tau) = f^{-1}(\mathbf{x}(\tau), r(\tau+1)). \tag{7.11}$$

This technique has been proposed in (Babuška et al. 1998). The multivariate mapping of the fuzzy model in Eq. (7.10) can be reduced to the univariate mapping $\hat{y}(\tau+1) = f_{\mathbf{x}}(u(\tau))$ by making use of the model structure. The subscript $_{\mathbf{x}}$

denotes that $f_{\mathbf{x}}$ is obtained for the particular state $\mathbf{x}(\tau)$. If the model is invertible, the inverse mapping $u(\tau) = f_{\mathbf{x}}^{-1}(r(\tau+1))$ can be obtained. The concept of invertibility and the respective conditions for the fuzzy model are related to the monotonicity of the model's input–output mapping. A fuzzy model f given by the rule base from Eq. (7.6) and the defuzzification method of Eq. (7.9) is *invertible* if $\forall \mathbf{x}$ and $\forall y$, a unique u exists such that $y = f(\mathbf{x}, u)$. In terms of the parameters of the model, the monotonicity is translated into the following conditions,

$$\text{card}(\text{core}(U_j)) = 1, \quad \forall j = 1, \ldots, M, \text{ and} \tag{7.12a}$$

$$\text{core}(U_1) < \ldots < \text{core}(U_M) \longrightarrow c_{i1} < c_{i2} < \ldots < c_{iM}, \quad \text{or}$$

$$\text{core}(U_1) < \ldots < \text{core}(U_M) \longrightarrow c_{i1} > c_{i2} > \ldots > c_{iM}, \tag{7.12b}$$

with $i = 1, \ldots, N$. Here, card$(\cdot)$ denotes the cardinality of a set.

Example 7.1 Figure 7.6 presents an example where both the above conditions are violated. Fuzzy set U_3 does not meet the condition card(core(U_3)) $= 1$; and for core(U_1) $<$ core(U_2) $\rightarrow c_{i1} < c_{i2}$, while for core($U_3$) $<$ core(U_4) $\rightarrow c_{i3} > c_{i4}$.

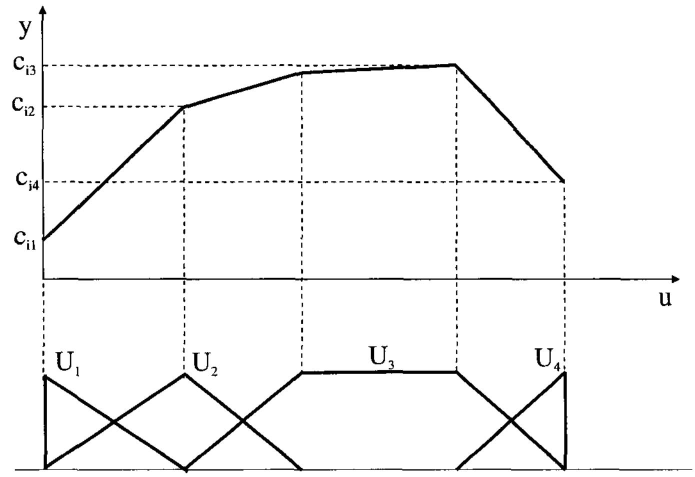


Fig. 7.6 Example where both conditions for the singleton model invertibility fail.

The inverse of the singleton fuzzy model can be formulated in the following theorem.

Theorem 7.1 **Inversion of the singleton fuzzy model.** *Let the process be represented by the singleton fuzzy model of Eq. (7.6) with the weighted-mean defuzzification method from Eq. (7.9). Further, let the antecedent membership functions* *form a partition, i.e., let* $\sum_{i=1}^{N} \mu_{A_i}(\mathbf{x}) = 1, \forall \mathbf{x}$, *and* $\sum_{j=1}^{M} \mu_{U_j}(u) = 1, \forall u$. *At a certain time* τ *the system is at the state* $\mathbf{x}(\tau)$, *and the inverse of the singleton model is given by the fuzzy rules*

$$\textbf{If } r(\tau+1) \textit{ is } C_j(\tau) \textbf{ then } u(\tau) \textit{ is } U_j, \quad j = 1, \ldots, M, \tag{7.13}$$

where C_j *are fuzzy sets that form a partition as in Fig. 7.7.*

The cores c_j *of the fuzzy sets* C_j *are given by*

$$c_j = \sum_{i=1}^{N} \mu_{A_i}(\mathbf{x}(\tau)) c_{ij}, \quad j = 1, \ldots, M. \tag{7.14}$$

The inference and defuzzification of the rules in Eq. (7.13) is accomplished by the fuzzy-mean method, i.e.

$$u(\tau) = \sum_{j=1}^{M} \mu_{C_j}(r(\tau+1)) \cdot \mathrm{core}(U_j) \,. \tag{7.15}$$

The open loop connecting the control action resulting from the inversion and the singleton fuzzy model gives an identity mapping (perfect control),

$$y(\tau+1) = f_{\mathbf{x}}(u(\tau)) = f_{\mathbf{x}}(f_{\mathbf{x}}^{-1}(r(\tau+1)) = r(\tau+1), \tag{7.16}$$

when $u(\tau)$ *exists, and thus* $r(\tau+1) = f(\mathbf{x}(\tau), u(\tau))$.

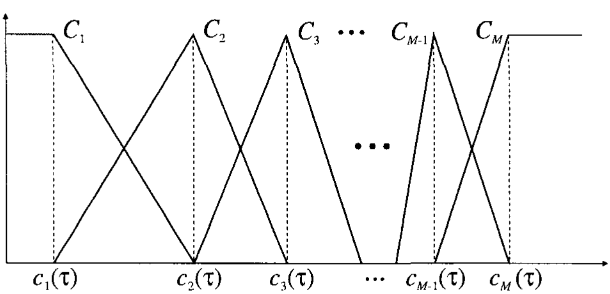


Fig. 7.7 Partition of fuzzy sets C_j built using the cores c_j.

Proof. As the product t-norm is used both for the **and** connective in the rule antecedent and for the Mamdani inference, the rule of Eq. (7.6) can be rewritten as

$$\textbf{If } \mathbf{x}(\tau) \text{ is } A_i \textbf{ then } (\textbf{If } u(\tau) \text{ is } U_j \textbf{ then } \hat{y}(\tau+1) \text{ is } c_{ij}) \,. \tag{7.17}$$

The inversion is made for a given state $\mathbf{x}(\tau)$. The degree of fulfillment for this state given in the proposition, '$\mathbf{x}(\tau)$ is A_i', is denoted by $\mu_{A_i}(\mathbf{x}(\tau))$. Then the N consequents of the rules containing a particular U_j (columns in Eq. (7.7)) can be aggregated. This aggregated consequents $c_j(\tau)$ are given by Eq. (7.14). As a result, the following set of M rules is obtained.

$$\textbf{If } u(\tau) \text{ is } U_j \textbf{ then } \hat{y}(\tau+1) \text{ is } c_j(\tau), \quad j = 1, \ldots, M. \tag{7.18}$$

As the Mamdani inference is performed by using the product t-norm, which has the commutative property, the antecedent and the consequent can be exchanged, inverting each of above rules, resulting in

$$\textbf{If } \hat{y}(\tau+1) \text{ is } c_j(\tau) \textbf{ then } u(\tau) \text{ is } U_j, \quad j = 1, \ldots, M. \tag{7.19}$$

As the consequents $c_j(\tau)$ are singletons, an interpolation method must be applied to obtain $u(\tau)$. This interpolation is accomplished by the fuzzy sets C_j defined as in Fig. 7.7. The serial connection of the inverse and the model gives an identity mapping if the desired reference $r(\tau+1)$ is in the range of the reached states from the actual state $\mathbf{x}(\tau)$, *i.e.*, $c_1 \leq r(\tau+1) \leq c_M$. Figure 7.8 shows that the inversion of the reference $r(\tau+1)$ is given by one and only one point for the described conditions. Moreover, the direct model reaches $y(\tau+1) = r(\tau+1)$ for the same state $\mathbf{x}(\tau)$, if it receives $u(\tau)$ as input.

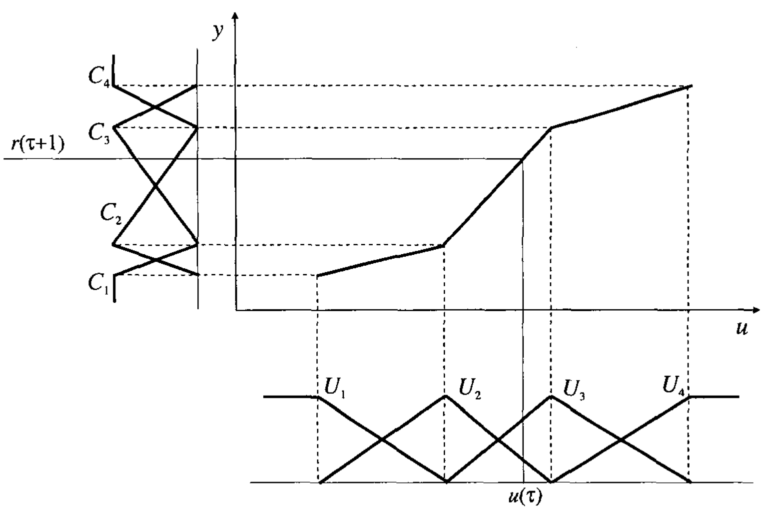


Fig. 7.8 Unique mapping between $u(\tau)$ and $r(\tau+1)$ for the singleton fuzzy model considered.

If the desired output cannot be reached from the current state in one time step, *i.e.*, $r(\tau+1) < c_1$ or $r(\tau+1) > c_M$, the control action is still the mapping with the minimal error. In the case that $r(\tau+1) > c_M$, $\mu_{C_M}(r(\tau+1)) = 1$, and the control input is $u(\tau) = \text{core}(U_M)$. The degree of fulfillment for the control action is given by $\mu_{U_M}(u(\tau)) = 1$, which yields the model output $y(\tau+1) = c_M$. As $c_M > c_j$, $1 \le j \le M-1$, the difference $|r(\tau+1) - y(\tau+1)|$ is the minimum possible. A similar situation occurs for $r < c_1$, where the control action $u(\tau) = \text{core}(U_1)$ yields the output $y(\tau+1) = c_1$. This is the best control action, since $c_1 < c_j$, $2 \le j \le M$. □

A simple example of the inversion of a singleton model is presented in Example 7.2. For fuzzy models with input delays $\hat{y}(\tau+1) = f(\mathbf{x}(\tau), u(\tau-d))$, the inversion cannot be applied directly since, in that case, the control law of Eq. (7.13) would compute a control action $u(\tau-d)$ which is d steps delayed. To generate the appropriate control action $u(\tau)$, the inverse model must be applied to a state $\mathbf{x}(\tau+d)$, d samples ahead, *i.e.*, $u(\tau) = f^{-1}(r(\tau+1+d), \mathbf{x}(\tau+d))$, with

$$\mathbf{x}(\tau+d) = [\hat{y}(\tau+d), \ldots, y(\tau-p+1+d), u(\tau-1), \ldots, u(\tau-m+1)]. \quad (7.20)$$

The unknown values $\hat{y}(\tau+1), \ldots, \hat{y}(\tau+d)$, are predicted by using the fuzzy model, *i.e.*, $\hat{y}(\tau+j) = f(\mathbf{x}(\tau+j-1), u(\tau+j-1-d))$, for $j = 1, \ldots, d$. Note that for large time delays, an accurate plant model is required for the d-steps ahead prediction, because predictions of unknown values of the state $\mathbf{x}$ are used in the subsequent steps.

Example 7.2 Consider a model of the form $\hat{y}(\tau+1) = f(y(\tau), u(\tau), u(\tau-1))$ where two linguistic terms {LOW, HIGH} are used for $y(\tau)$, and three terms {SMALL, MEDIUM, LARGE} for $u(\tau)$ and $u(\tau-1)$. Therefore the model rule base consists of $2 \times 3 \times 3 = 18$ rules, in total.

If $y(\tau)$ is LOW **and** $u(\tau)$ is SMALL **and** $u(\tau-1)$ is SMALL **then** $\hat{y}(\tau+1)$ is c_{11}
If $y(\tau)$ is LOW **and** $u(\tau)$ is SMALL **and** $u(\tau-1)$ is MEDIUM **then** $\hat{y}(\tau+1)$ is c_{21}
⋮
If $y(\tau)$ is HIGH **and** $u(\tau)$ is LARGE **and** $u(\tau-1)$ is LARGE **then** $\hat{y}(\tau+1)$ is c_{63}

In this example, $\mathbf{x}(\tau) = [y(\tau), u(\tau-1)]$, $N = 6$ and $M = 3$. The rule base can be represented by Table 7.1. For a given state $\mathbf{x}(\tau) = [y(\tau), u(\tau-1)]$, the degree of fulfillment of the first antecedent proposition "$\mathbf{x}(\tau)$ is A", is calculated as $\mu_{A_i}(\mathbf{x}(\tau))$. Using Eq. (7.14), the consequents $c_j(\tau)$ are given by

$$c_j(\tau) = \frac{\sum_{i=1}^{6} \mu_{A_i}(\mathbf{x}(\tau))\, c_{ij}}{\sum_{i=1}^{6} \mu_{A_i}(\mathbf{x}(\tau))}, \quad j = 1, 2, 3, \quad (7.21)$$

Table 7.1 Example rule base for a model with three antecedents and 18 rules.

	$u(\tau)$		
$\mathbf{x}(\tau)$	SMALL	MEDIUM	LARGE
A_1(LOW & SMALL)	c_{11}	c_{12}	c_{13}
A_2(LOW & MEDIUM)	c_{21}	c_{22}	c_{23}
A_3(LOW & LARGE)	c_{31}	c_{32}	c_{33}
A_4(HIGH & SMALL)	c_{41}	c_{42}	c_{43}
A_5(HIGH & MEDIUM)	c_{51}	c_{52}	c_{53}
A_6(HIGH & LARGE)	c_{61}	c_{62}	c_{63}

resulting in the following three rules.

$$\begin{aligned}
&\textbf{If } u(\tau) \text{ is } U_1 \textbf{ then } \hat{y}(\tau+1) \text{ is } c_1(\tau) \\
&\textbf{If } u(\tau) \text{ is } U_2 \textbf{ then } \hat{y}(\tau+1) \text{ is } c_2(\tau) \\
&\textbf{If } u(\tau) \text{ is } U_3 \textbf{ then } \hat{y}(\tau+1) \text{ is } c_3(\tau)
\end{aligned}$$

An example of membership functions $C_j(\tau)$, $j = 1, 2, 3$, of the fuzzy partition created by using the consequent singletons $c_1(\tau)$, $c_2(\tau)$, $c_3(\tau)$ is shown in Fig. 7.9. Assuming that the fuzzy rule base is monotonic, the rules can be inverted

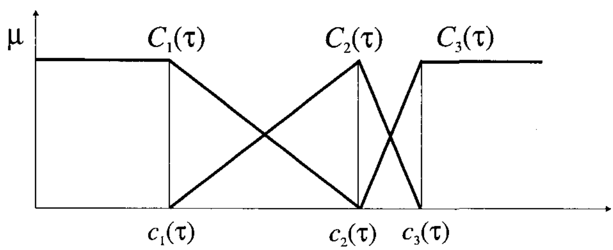


Fig. 7.9 Fuzzy partition created from $c_1(\tau)$, $c_2(\tau)$ and $c_3(\tau)$.

resulting in

$$\begin{aligned}
&\textbf{If } r(\tau+1) \text{ is } C_1(\tau) \textbf{ then } u(\tau) \text{ is } U_1 \\
&\textbf{If } r(\tau+1) \text{ is } C_2(\tau) \textbf{ then } u(\tau) \text{ is } U_2 \\
&\textbf{If } r(\tau+1) \text{ is } C_3(\tau) \textbf{ then } u(\tau) \text{ is } U_3
\end{aligned}$$

If the rule base is not invertible due to non-monotonicity, the inversion can still be performed for the P monotonous parts, with $P \geq 2$. For each of the P parts, a control action is found by inverting the singleton fuzzy model. The choice of one control action is made by using additional criteria, *e.g.*minimal control effort. This solution is in general different from the solution obtained by predictive control, where more than one-step-ahead predictions are used, and the control effort is included in the objective function, see Sec. 7.5. Therefore, in these situations it is preferable to use the predictive control scheme as in Sec. 7.5, which gives an optimal control solution directly. The inversion of non-monotonous systems is not considered further.

7.3 Inversion of an affine Takagi–Sugeno fuzzy model

This section presents the inversion of a TS fuzzy model that is affine with respect to a control action. This method was introduced by Sousa et al. (1997). Let a MIMO fuzzy system be given as in Eq. (5.4). The global control problem is reduced to control one of the outputs, say y_1, by manipulating one of the inputs, say, u_1. The remaining inputs are considered constant values or represent measurable disturbances, which were defined as non-controllable inputs in Sec. 7.1.1. Denote $y \triangleq y_1$ and $u \triangleq u_1$ for the ease of notation. Let the control action $u(\tau) \equiv u_1(\tau)$ not be considered in the state vector of Eq. (5.3), and thus the vector of states is now given by

$$\begin{aligned}\mathbf{x}(\tau) = [&y_1(\tau), \ldots, y_1(\tau - p_1 + 1), \ldots, y_p(\tau), \ldots, y_p(\tau - p_p + 1),\\ &u_1(\tau - 1), \ldots, u_1(\tau - m_1 + 1), u_2(\tau), \ldots, u_2(\tau - m_2 + 1), \ldots,\\ &u_m(\tau), \ldots, u_m(\tau - m_m + 1)]^T.\end{aligned} \tag{7.22}$$

This notation helps to describe the inversion as it will be seen in the following. The system under control is represented by a MISO model,

$$\hat{y}(\tau + 1) = f(\mathbf{x}(\tau), u(\tau)). \tag{7.23}$$

Note that only a MISO system is considered, because only one variable is under control. The parameters $m_1, \ldots, m_m$ and $p_1, \ldots, p_p$ are the orders of the inputs and outputs, as before in Eq. (5.3). The dimension of the state vector is now given by $n = \sum_{j=1}^{m} m_j + \sum_{j=1}^{p} p_j - 1$.

7.3.1 *TS fuzzy model*

The unknown function f in Eq. (7.23) is parameterized by the Takagi–Sugeno (TS) fuzzy model (Takagi and Sugeno 1985), which can approximate a large class of nonlinear systems, see Sec. 5.2.3. In this type of models, the rule consequents are crisp functions of the antecedent variables. The most common TS fuzzy model utilizes an affine linear function as the consequent function.

$$
\begin{aligned}
R^k\colon\ & \textbf{If } y_1(\tau) \text{ is } A_1^k \textbf{ and } \ldots y_1(\tau - p_1 + 1) \text{ is } A_{p_1}^k \textbf{ and} \\
& y_2(\tau) \text{ is } A_{(p_1+1)}^k \textbf{ and } \ldots y_2(\tau - p_2 + 1) \text{ is } A_{(p_1+p_2)}^k \textbf{ and } \ldots \\
& y_p(\tau) \text{ is } A_{(p_1+\ldots+p_{p-1}+1)}^k \textbf{ and } \ldots y_p(\tau - p_p + 1) \text{ is } A_{(p_1+\ldots+p_p)}^k \textbf{ and} \\
& u_1(\tau) \text{ is } B_1^k \textbf{ and } \ldots u(\tau - m_1 + 1) \text{ is } B_{m_1}^k \textbf{ and } \ldots \\
& u_m(\tau) \text{ is } B_{(m_1+\ldots+m_{m-1}+1)}^k \textbf{ and } \ldots \\
& u_m(\tau - m_m + 1) \text{ is } B_{(m_1+\ldots+m_m)}^k \textbf{ then} \\
& \hat{y}^k(\tau+1) = \sum_{j=1}^{p_1} a_{1j}^k y_1(\tau - j + 1) + \ldots + \sum_{j=1}^{p_p} a_{pj}^k y_p(\tau - j + 1) + \\
& \qquad \sum_{j=1}^{m_1} b_{1j}^k u_1(\tau - j + 1) + \ldots + \sum_{j=1}^{m_m} b_{mj}^k u_m(\tau - j + 1) + c^k, \\
& k = 1, \ldots, K,
\end{aligned} \tag{7.24}
$$

where A_j^k, B_j^k are fuzzy sets, $a_{1j}^k, \ldots, a_{pj}^k$, $b_{1j}^k, \ldots, b_{mj}^k$ and c^k, are crisp consequent parameters, and K denotes the number of rules in the rule base. The consequents can be written in a more compact form,

$$
\begin{aligned}
\hat{y}^k(\tau+1) = & \sum_{\ell=1}^{p} \left(\sum_{j=1}^{p_\ell} a_{\ell j}^k y_\ell(\tau - j + 1) \right) + \\
& \sum_{\ell=1}^{m} \left(\sum_{j=1}^{m_\ell} b_{\ell j}^k u_\ell(\tau - j + 1) \right) + c^k.
\end{aligned} \tag{7.25}
$$

The fuzzy rule base given in Eq. (7.24) can be expressed in a compact way by using the state vector $\mathbf{x}(\tau)$ and the control action $u(\tau)$ as

$$
R^k\colon\ \textbf{If } [\mathbf{x}(\tau), u(\tau)] \text{ is } A^k \textbf{ then } \hat{y}^k(\tau+1) = (\mathbf{a}^k)^T \mathbf{x}(\tau) + b^k u(\tau) + c^k, \tag{7.26}
$$

where $k = 1, \ldots, K$, and A^k is the antecedent multidimensional fuzzy set, defined by its membership function

$$
\mu_{A^k}(\mathbf{x}(\tau), u(\tau))\colon \mathbb{R}^{n+1} \to [0, 1], \tag{7.27}
$$

resulting from the conjunctive aggregation of all the A_j^k, for $j = 1, 2, \ldots, \sum_{\ell=1}^{p} p_\ell$, and B_j^k, for $j = 1, 2, \ldots, \sum_{\ell=1}^{m} m_\ell$. Thus, the antecedent fuzzy set A^k is defined as

$$
A^k = A_1^k \circledast \ldots A_{p_1}^k \circledast \ldots A_{(p_1+\ldots+p_p)}^k \circledast B_1^k \circledast \ldots B_{m_1}^k \circledast \ldots B_{(m_1+\ldots+m_m)}^k, \tag{7.28}
$$

where ⊛ denotes a t-norm operator. The consequent parameters of the kth rule, $\mathbf{a}^k \in \mathbb{R}^n$, and $b^k, c^k \in \mathbb{R}$ are related to the consequents in Eq. (7.24) by

$$\begin{aligned} \mathbf{a}^k &= [a_{11}^k, a_{12}^k, \ldots, a_{p\,p_p}^k, b_{12}^k, \ldots, b_{m\,m_m}^k]^T \\ b^k &= b_{11}^k, \\ c^k &= c^k. \end{aligned} \tag{7.29}$$

The output of the model $\hat{y}(\tau+1)$ is computed by

$$\hat{y}(\tau+1) = \sum_{k=1}^{K} \hat{\beta}^k(\tau)[(\mathbf{a}^k)^T \mathbf{x}(\tau) + b^k u(\tau) + c^k], \tag{7.30}$$

where $\hat{\beta}^k(\tau)$ is the normalized degree of fulfillment of the kth rule's antecedent, and it is given by

$$\hat{\beta}^k(\tau) = \frac{\mu_{A^k}([\mathbf{x}(\tau), u(\tau)])}{\sum_{j=1}^{K} \mu_{A^j}([\mathbf{x}(\tau), u(\tau)])}. \tag{7.31}$$

7.3.2 *Inversion of the TS fuzzy model*

In order to invert the fuzzy model, *i.e.*, to compute $u(\tau)$ based on the current state $\mathbf{x}(\tau)$ and on the desired future output $r(\tau+1)$, the general fuzzy model of Eq. (7.26) can be described by the simplified affine fuzzy model of the form

$$\hat{y}(\tau+1) = f_1(\mathbf{x}(\tau)) + f_2(\mathbf{x}(\tau))u(\tau). \tag{7.32}$$

If the term in $u(\tau)$ is not considered in the input, *i.e.*, the input of the fuzzy model is just given by the state vector $\mathbf{x}(\tau)$, this model is parameterized by the Takagi–Sugeno (TS) structure as

$$R^k\text{: } \textbf{If } \mathbf{x}(\tau) \text{ is } A_{[P]}^k \textbf{ then } \hat{y}^k(\tau+1) = (\mathbf{a}^k)^T \mathbf{x}(\tau) + b^k u(\tau) + c^k \tag{7.33}$$

with $k = 1, \ldots, K$. The fuzzy set $A_{[P]}^k$ represented by $\mu_{A_{[P]}^k}(\mathbf{x})\colon \mathbb{R}^n \to [0,1]$ is the projection of the antecedent fuzzy set $A^k \in \mathbb{R}^{n+1}$ onto the space of the state vector $\mathbf{x}(\tau) \in X \subset \mathbb{R}^n$. This projection can be obtained by using the Gustafson–Kessel algorithm, presented in Algorithm 5.1. Thus, the projected membership functions are given by

$$\mu_{A_{[P]}^k}(\mathbf{x}(\tau)) = \frac{1}{\sum_{j=1}^{K} (d_{k\tau}/d_{j\tau})^{2/(\alpha-1)}}, \tag{7.34}$$

with

$$d_{k\tau} = |\mathbf{F}_k^x|^{1/n} (\mathbf{x}(\tau) - \mathbf{v}_k^x)^T (\mathbf{F}_k^x)^{-1} (\mathbf{x}(\tau) - \mathbf{v}_k^x). \tag{7.35}$$

Here, $\mathbf{v}_k^x = [v_{1k}, \ldots, v_{nk}]^T$ denotes the projection of the cluster center k onto X, and each $\mathbf{F}_k^x = [f_{ij}]$, $1 \leq i, j \leq n$ is the submatrix of the covariance matrix $\mathbf{F}_k$. Since the antecedent partition of Eq. (7.33) is different from the one in Eq. (7.26), the optimal consequent parameters $\mathbf{a}^k$, b^k and c^k must be re-estimated utilizing the least squares algorithm, as presented in Sec. 5.4.2, by using Eq. (5.31) to Eq. (5.34), where the matrices $\mathbf{\Gamma}^k$ contain the membership degrees given by Eq. (7.34) and Eq. (7.35). The normalized degrees of fulfillment $\hat{\beta}_{[P]}^k$ for the affine TS fuzzy model are defined as

$$\hat{\beta}_{[P]}^k(\tau) = \frac{\mu_{A_{[P]}^k}(\mathbf{x}(\tau))}{\sum_{j=1}^K \mu_{A_{[P]}^j}(\mathbf{x}(\tau))} . \tag{7.36}$$

The predicted output of the model $\hat{y}(\tau+1)$ is recalculated by

$$\hat{y}(\tau+1) = \sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau)[(\mathbf{a}^k)^T\mathbf{x}(\tau) + b^k u(\tau) + c^k]. \tag{7.37}$$

As the antecedent of Eq. (7.33) does not include the input term $u(\tau)$, the model output $\hat{y}(\tau+1)$ is affine in the input $u(\tau)$. Thus, Eq. (7.37) can be easily divided into two terms

$$\hat{y}(\tau+1) = \sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau)\left[(\mathbf{a}^k)^T\mathbf{x}(\tau) + c^k\right] + \sum_{k=1}^K \beta_{[P]}^k(\tau)\, b^k\, u(\tau). \tag{7.38}$$

This expression can be translated in the nonlinear affine form given in Eq. (7.32), with

$$\begin{aligned} f_1(\mathbf{x}(\tau)) &= \sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau)\left[(\mathbf{a}^k)^T\mathbf{x}(\tau) + c^k\right] \\ f_2(\mathbf{x}(\tau)) &= \sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau) b^k . \end{aligned} \tag{7.39}$$

If the goal is that the model at time step $\tau+1$ equals the reference output, *i.e.*, $\hat{y}(\tau+1) = r(\tau+1)$, the corresponding control input $u(\tau)$ is computed by a simple algebraic manipulation on Eq. (7.32),

$$u(\tau) = \frac{r(\tau+1) - f_1(\mathbf{x}(\tau))}{f_2(\mathbf{x}(\tau))} . \tag{7.40}$$

In terms of Eq. (7.38) one obtains

$$u(\tau) = \frac{r(\tau+1) - \sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau)[(\mathbf{a}^k)^T\mathbf{x}(\tau) + c^k]}{\sum_{k=1}^K \hat{\beta}_{[P]}^k(\tau)\, b^k} . \tag{7.41}$$

Thus, similarly to a singleton fuzzy model, the TS fuzzy model affine on the control input, as in Eq. (7.33), can be *exactly* inverted, provided that the function $f_2(\mathbf{x}(\tau))$ is different from zero. The procedure to be followed for systems with input delays is identical to the one presented for the singleton fuzzy model in Sec. 7.2, where for d samples of delay, the inverse model must be applied d-samples ahead.

7.4 On-line adaptation of feedforward fuzzy models

Many industrial processes are characterized by frequent changes in the operating conditions, such as the ones caused by varying quality of the raw materials, varying process throughput and changing product mix. In order to assure the desired product quality, the process control system must be able to cope with frequent changes in the process parameters and structure. One possible approach is to adapt the controller parameters based on a specified performance measure. In fuzzy control literature, several adaptive control structures have been presented, such as the classical self-organizing linguistic controller (Procyk and Mamdani n.d.), a neuro-fuzzy controller with temporal backpropagation learning (Jang 1992) or a self-learning fuzzy controller based on reinforcement learning (Berenji and Khedkar 1992). The common feature of these approaches is that the controller is adapted directly without identifying the plant model.

A different approach consists of adapting the fuzzy model, using the exact inversion to derive the control input (Sousa et al. 1995). The advantage of the proposed scheme is that, apart from the controller design, the process model can be used for other purposes, such as monitoring, fault detection and prediction, when comparing different control scenarios. An extension to an adaptive fuzzy model predictive control scheme is then possible. Since the control actions are derived by inverting the fuzzy model on-line, the controller can be adapted automatically, if the invertibility conditions are fulfilled.

Adaptation of fuzzy models can be distinguished in adapting the antecedent membership functions or the consequents. In many practical situations, the initial antecedent partition derived through off-line identification remains valid. Therefore, only the adaptation of consequent parameters will be considered. In the absence of a reasonably accurate initial model, the antecedent membership functions can be adapted by using a nonlinear optimization technique. Various approaches have been suggested in literature, such as error backpropagation, nonlinear programming or genetic algorithms (Jang 1992, Klawonn et al. 1994). Moreover, the identification of fuzzy models using product-space fuzzy clustering, as presented in Sec. 5.4, can be performed using new data from the system.

To cope with model–plant mismatch and disturbances, on-line adaptation of the fuzzy model is done in the IMC scheme shown in Fig. 7.10, where the consequents of the fuzzy model are adapted. The adaptation of singleton fuzzy models

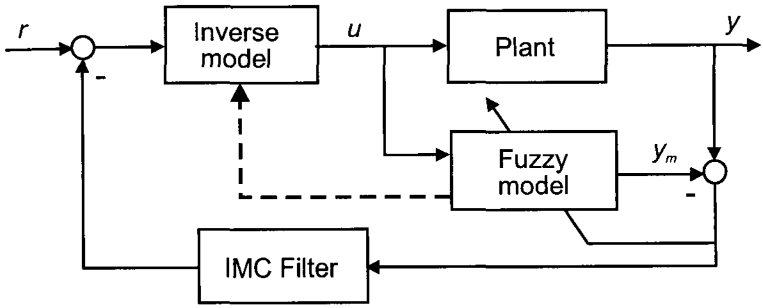


Fig. 7.10 Adaptive internal model control scheme.

is done as follows. The predicted output for the singleton model, as in Eq. (7.3), is given by Eq. (7.9), which is linear in the consequent parameters. These features allow for a straightforward application of standard recursive least-squares algorithms for estimating the consequent parameters from data, see, *e.g.*, Ljung (1987). Although the TS fuzzy models given by Eq. (7.24) are also linear in the consequent parameters, the number of parameters to be tuned is very high ($K \times (n+1)$), and the adaptation of all the parameters to stable values is difficult to obtain. Therefore, only on-line adaptation of singleton fuzzy models is considered. The consequent parameters can be indexed by the rule number, and arranged in a column vector denoted $\mathbf{c}(\tau) = [c^1(\tau), c^2(\tau), \ldots, c^K(\tau)]^T$, where K is the number of rules. Similarly, the normalized degrees of fulfillment of the rule antecedents are arranged in a column vector $\hat{\beta}(\tau) = [\hat{\beta}^1(\tau), \hat{\beta}^2(\tau), \ldots, \hat{\beta}^K(\tau)]$. These normalized degrees of fulfillment are computed as

$$\hat{\beta}^k(\tau) = \frac{\beta^k(\tau)}{\sum_{j=1}^{K} \beta^j(\tau)}. \tag{7.42}$$

The consequent vector $\mathbf{c}(\tau)$ is updated recursively by using the standard least squares equation

$$\mathbf{c}(\tau) = \mathbf{c}(\tau-1) + \frac{\mathbf{R}(\tau-1)\hat{\beta}(\tau)}{\lambda + \hat{\beta}^T(\tau)\mathbf{R}(\tau-1)\hat{\beta}(\tau)} \left[y(\tau) - \hat{\beta}^T(\tau)\mathbf{c}(\tau-1) \right], \tag{7.43}$$

where λ is a constant (forgetting factor) and $\mathbf{R}(\tau)$ is a covariance matrix updated with

$$\mathbf{R}(\tau) = \frac{1}{\lambda}\left[\mathbf{R}(\tau-1) - \frac{\mathbf{R}(\tau-1)\hat{\beta}(\tau)\hat{\beta}^T(\tau)\mathbf{R}(\tau-1)}{\lambda + \hat{\beta}^T(\tau)\mathbf{R}(\tau-1)\hat{\beta}(\tau)}\right] \tag{7.44}$$

The forgetting factor λ influences the tracking capabilities of the adaptation algorithm. The smaller the λ, the faster the consequent parameters adapt, not only to the process changes but also to disturbances and noise. Therefore, the choice of λ is problem-dependent. The initial covariance is usually set to $\mathbf{R}(0) = \sigma \cdot \mathbf{I}$, where $\mathbf{I}$ is a $K \times K$ identity matrix, and σ is a large positive constant. Another possibility is to calculate the initial covariance $\mathbf{R}(0)$ from part of the identification data (Ljung 1987).

The presented model-based adaptation has several advantages over the more conventional model-free adaptive schemes.

- Adaptation is based on a standard linear parameter estimation algorithm with well-understood numerical properties.
- The model, the adaptation law and the controller are easily implemented using vector and matrix operations, allowing for an efficient in-line implementation even with high sampling rates. Most of the other adaptive control schemes have higher computational and memory requirements (self-organizing fuzzy control) and slower convergence (reinforcement and backpropagation based controllers).
- Once a process model is available, it can be used for multiple purposes, such as monitoring, fault diagnosis or prediction. Extensions of the proposed scheme to adaptive model predictive control are possible.

The presented scheme allows for *local* adaptation (learning) of the controller, as opposed to *parameter tracking* used in linear adaptive control. A drawback of linear methods is that a balance between the tracking speed and insensitivity to noise is difficult to achieve and the linear controller has no 'memory', *i.e.*, for a nonlinear system it must continuously re-adapt the parameters as the process state changes.

7.5 Predictive control using the inversion of a fuzzy model

In the absence of any constraints and disturbances, and when the model and the system are identical (there is no model–plant mismatch), the inversion can be performed at each sampling instant, giving the optimal control action. However, one of the most serious problems of inverse control is the presence of constraints in some of the variables of the system. A system presents, at least, absolute and rate constraints on the control actions $\mathbf{u}(\tau)$ due to physical or safety reasons. Constraints in state variables are also often found. Model predictive control, presented in Appendix A, is a general control method which can deal with constraints of the system, allowing us to find the optimal sequence of control actions $\mathbf{u}(\tau), \ldots, \mathbf{u}(\tau + H_p - 1)$ over the prediction horizon H_p, for a given objective function, using a (nonlinear) model of the process. Even with the traditional cost function consisting of the sum squared error between a desired reference and the predicted output (see Eq. (A.2) for more details), the optimization problem remains non-convex if a nonlinear model of the system is used. Therefore, as the inversion algorithms presented are computationally fast, it is advantageous to utilize controllers based on inverse plant models combined with a predictive control scheme.

For a system with constraints, if none of them are violated, one-step-ahead prediction is equivalent to inverse model control and also guarantees optimal performance for first-order systems. On the other hand, when the constraints are active, inverse model control results in sub-optimal or even unfeasible performance. This observation leads to the idea of combining the predictive control strategy with inverse control described in Sec. 7.1 (Sousa et al. 1997). Such a control scheme can circumvent the non-convex optimization problems, allowing the use of predictive control in real-time for systems with relatively small sampling times. When a recursive application of the inverse model control law over the entire prediction horizon results in a violation of a constraint at any step, predictive control is used, since it will result in better performance. On the other hand, if no constraints are violated, the first control action computed by the inverse model is applied to the process. Algorithm 7.1 summarizes the described control strategy.

Algorithm 7.1 Combination of predictive and inverse-model control.

Step 1: **Apply inverse-model control.** Compute the control actions $u(\tau), \ldots,$, $u(\tau + H_p - 1)$ over the prediction horizon and the predicted process outputs using the inverse fuzzy model to calculate the control output, and the fuzzy model to calculate the predicted outputs.

Step 2: **Check constraints.** If some of the constraints are violated at any of the prediction steps, go to Step 3; otherwise apply the control action $u(\tau)$ computed in Step 1 to the process.

Step 3: **Use predictive control.** By using a given objective function, a sub-optimal solution for the control actions is found. Apply $u(\tau)$ to the process.

The optimization performed in Step 3 is usually non-convex. This can be a serious drawback if an iterative optimization is used. Methods such as sequential quadratic programming (Gill et al. 1981) can exhibit high computational costs and converge frequently to local minima, hampering the application of the combined control scheme described in Algorithm 7.1. Another possibility for the optimization problem is to transform the problem into a discrete space of control alternatives. Techniques from operational research and decision making, such as dynamic programming, genetic algorithms or the branch-and-bound method can be utilized. The application of branch-and-bound and genetic algorithms in model predictive control is described in Chapter 10. The control scheme proposed in Algorithm 7.1 avoids the oscillations that usually occur, due to the discretization of the control space, see Sec. 10.1. With constant or slowly varying references, the constraints are typically not violated and the inversion can be applied, yielding a continuous (interpolated) control action. An example presented in the next section illustrates the control scheme proposed above, as well as the direct application of inverse fuzzy control.

7.6 Pressure control of a fermentation tank

This section describes an example regarding the application of the control methods that are discussed in this chapter. Inverse model control and the combination of inverse control with predictive control are applied to highly nonlinear pressure dynamics in a laboratory fermenter presented in Fig. 7.11 *. The volume of the fermenter tank is 40 l, and at normal working conditions it is filled with 25 l of water. At the bottom of the tank, air is fed into the water at a specified flow rate, and kept constant by a local mass flow controller. The air pressure above the water level is controlled by an outlet valve at the top of the tank. With a constant input flow rate, the system has a single input, the valve position, and a single output, the air pressure. Because of the underlying physical mechanisms, and because of the nonlinear characteristic of the control valve, the process has a nonlinear steady-state characteristic, as well as a nonlinear dynamic behavior. In the control experiments presented here, the second process input, the air flow rate, is kept constant. Under the conditions specified above, the smallest time constant of the process is about 45 s, which allows for a sample time of $T_s = 5$ s. The following nonlinear differential equation is used as the simulation model which describes the pressure dynamics

$$\frac{dP}{dt} = \frac{1000\, R_q T_v}{22,4\, V_h} \cdot \left[\Phi_g - (\pi r_H^2)\sqrt{\frac{2P_0}{\rho_0 K_f}\ln(\frac{P}{P_0})} \right] . \qquad (7.45)$$

The symbols represent the following quantities.

* This application was made using data collected from the fermenter at the Kluyver Laboratory for Biotechnology, Delft University of Technology.

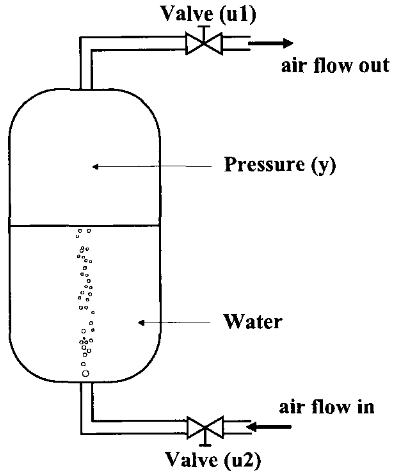


Fig. 7.11 Laboratory fermenter.

R_q : the gas constant (8.134 J $\mathrm{mol}^{-1}\mathrm{K}^{-1}$),
T_v : temperature (305 K),
V_h : gas volume (0.015 m^3),
Φ_g : gas flow-rate (3.75×10^{-4} $\mathrm{m}^3\mathrm{s}^{-1}$),
r_H : radius of the outlet pipe (0.0178 m),
P_0 : reference pressure (1.013×10^5 N m^{-2}),
ρ_0 : outside air density (1.2 Kg m^{-3}),
P : pressure in the tank (N m^{-2}),
K_f : valve friction factor (J mol^{-1}).

The valve friction factor K_f is a nonlinear function of the valve position u and the flow rate Φ_g. The maximum changes in the valve position are $\Delta u(\tau) = -\Delta u(\tau) = 10\%$ of the total range per sample, and the level constraints are $u_{\min} = 0\%$ and $u_{\max} = 90\%$ of the valve position. More detailed descriptions of the process can be found in van Can et al. (1995).

7.6.1 *Fuzzy modeling*

The inversion of singleton models presented in Sec. 7.2 and the inversion of a TS fuzzy model affine on the control action $u(\tau)$ explained in Sec. 7.3, as well as the predictive control scheme based on the inversion of a fuzzy model, described in Sec. 7.5, are applied to the fermenter. In order to apply these control schemes, a singleton fuzzy model and an affine TS fuzzy model are developed for the pressure system.

7.6.1.1 *Singleton fuzzy model*

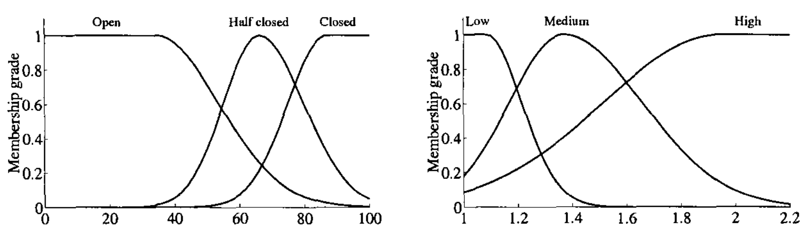
First, a fuzzy model of the Takagi–Sugeno type (Takagi and Sugeno 1985) is constructed from the process input–output measurements by means of product-space fuzzy clustering presented in Sec. 5.4. The model consists of three rules with linear consequents, including the bias terms, to capture the different operating regimes. The current valve position is denoted $u(\tau)$, the current pressure $y(\tau)$, and the pressure at the next sampling instant $\hat{y}(\tau+1)$. The identified model is as follows.

1. **If** $y(\tau)$ is LOW **and** $u(\tau)$ is OPEN
   then $\hat{y}(\tau+1) = 0.67y(\tau) + 0.0007u(\tau) + 0.35$
2. **If** $y(\tau)$ is MEDIUM **and** $u(\tau)$ is HALF CLOSED
   then $\hat{y}(\tau+1) = 0.80y(\tau) + 0.0028u(\tau) + 0.07$
3. **If** $y(\tau)$ is HIGH **and** $u(\tau)$ is CLOSED
   then $\hat{y}(\tau+1) = 0.90y(\tau) + 0.0071u(\tau) - 0.39\,.$

This rule base represents a nonlinear first-order regression model

$$\hat{y}(\tau+1) = f(y(\tau), u(\tau))\,. \tag{7.46}$$

Figure 7.12a shows the membership functions for 'OPEN', 'HALF CLOSED' and 'CLOSED' for the valve position, and Fig. 7.12b the membership functions found for the pressure, which are 'LOW', 'MEDIUM' and 'HIGH'.



(a) valve position (% closed) (b) pressure (10^5 Pa)

Fig. 7.12 Membership functions for the TS fuzzy model premise variables.

This model gives certain insight into the nonlinear dynamics of the system, as it is represented as a set of local linear ARX models. The validity regions for these models are defined by the antecedent membership functions. It can be observed, for instance, that for low values of the pressure, the system has both lower gain and slower dynamics than for high pressure, which agrees well with the prior knowledge about the system.

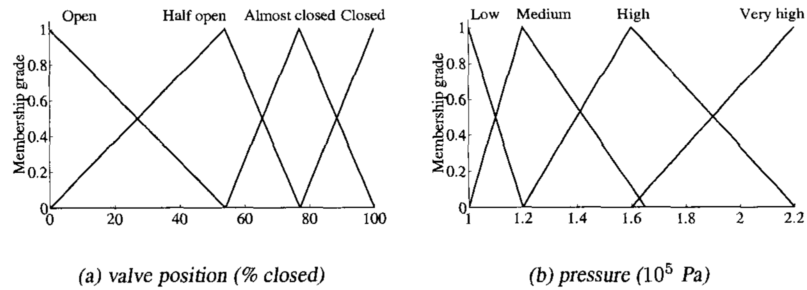


Fig. 7.13 Membership functions for the inputs of the singleton fuzzy model.

The singleton fuzzy model is derived from the TS fuzzy model by using the method described in (Babuška et al. 1998). The membership functions obtained for the inputs with this method are presented in Fig. 7.13.

This singleton model consists of 16 rules containing all possible combinations of the antecedents for the valve position and for the pressure. The singleton consequents are estimated by using the least-squares method. The rules obtained are presented in Table 7.2. In this table, the first rule, *e.g.*, reads '**If** $y(\tau)$ is LOW **and** $u(\tau)$ is OPEN, **then** $\hat{y}(\tau+1) = 1.06$'. The model is validated using the nonlinear output error, where only the real input of the system is used, *i.e.*, $\hat{y}(\tau+1) = f(u(\tau), \hat{y}(\tau))$. Figure 7.14 presents the validation made on a different data set from the one used for identification.

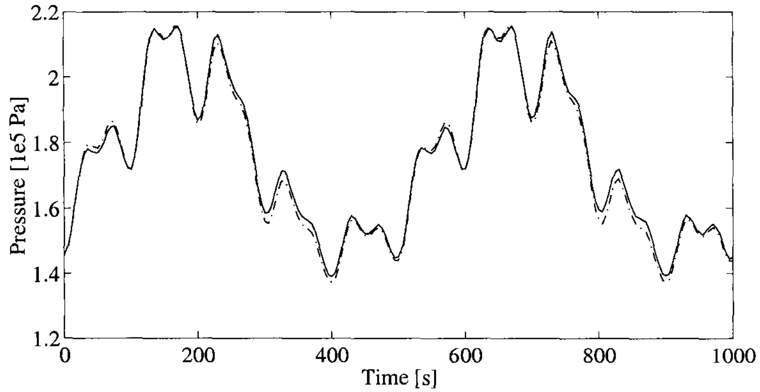


Fig. 7.14 Validation of the singleton model (solid line – process, dash-dotted line – model).

Table 7.2 Singleton fuzzy model described by the individual rules.

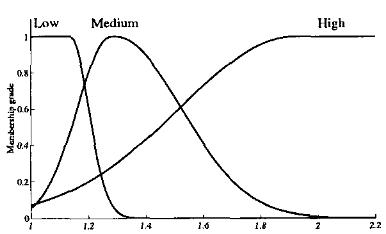
R#	$y(\tau)$	$u(\tau)$	$\hat{y}(\tau+1)$
1	Low	Open	1.06
2	Medium	Open	1.13
3	High	Open	1.46
4	Very high	Open	1.84
5	Low	Half open	1.05
6	Medium	Half open	1.19
7	High	Half open	1.51
8	Very high	Half open	2.03
9	Low	Almost closed	1.07
10	Medium	Almost closed	1.23
11	High	Almost closed	1.63
12	Very high	Almost closed	2.12
13	Low	Closed	1.11
14	Medium	Closed	1.33
15	High	Closed	1.79
16	Very high	Closed	2.34

7.6.1.2 *Affine Takagi–Sugeno model*

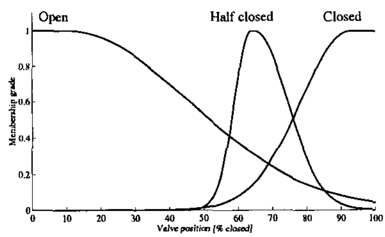
First, a second-order TS fuzzy model in $u(\tau)$ is developed for the pressure system. This model is a straightforward extension of the model as in Eq. (7.46), and the process dynamics is represented by $\hat{y}(\tau+1) = f(y(\tau), u(\tau), u(\tau-1))$. The additional term $u(\tau-1)$ in the model is important to derive the affine TS fuzzy model, as will be explained below. Figure 7.15a shows the membership functions fitted to the projection of the fuzzy partition onto the pressure antecedent variable. Figure 7.15b and Fig. 7.15c present membership functions built in the same way for the valve position and the one-step delayed valve position, respectively. By not considering the term on $u(\tau)$ in the input, the TS fuzzy model remains affine in the input $u(\tau)$, as shown in Sec. 7.3. The consequent parameters are re-estimated by using the least-squares technique, and the following three rules are obtained.

1. **If** $y(\tau)$ is Low **and** $u(\tau-1)$ is Open **then**
   $\hat{y}(\tau+1) = 0.76y(\tau) + 2.4 \cdot 10^{-3}u(\tau) + 2.7 \cdot 10^{-4}u(\tau-1) + 0.15$
2. **If** $y(\tau)$ is Medium **and** $u(\tau-1)$ is Half closed **then**
   $\hat{y}(\tau+1) = 0.75y(\tau) + 2.4 \cdot 10^{-4}u(\tau) + 3.1 \cdot 10^{-4}u(\tau-1) + 0.27$
3. **If** $y(\tau)$ is High **and** $u(\tau-1)$ is Closed **then**
   $\hat{y}(\tau+1) = 0.93y(\tau) + 8.2 \cdot 10^{-3}u(\tau) + 9.2 \cdot 10^{-4}u(\tau-1) - 0.45$

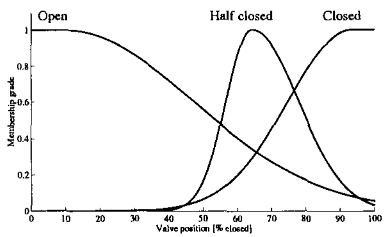
Figure 7.15d shows the validation of the model by simulation from the inputs



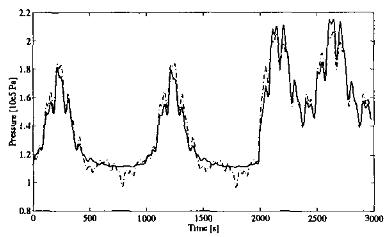
(a) Membership functions for pressure.



(b) Membership functions for $u(\tau)$.



(c) Membership functions for $u(\tau - 1)$.



(d) Model validation (solid - process, dash-dotted - model).

Fig. 7.15 Membership functions and model validation for the TS model.

only, as it was done for the singleton fuzzy model. This model is slightly inferior to the singleton model presented in Fig. 7.14. Therefore, it is expected that the control based on this model also presents inferior performance. The next sections will apply both models in controlling the process.

7.6.2 *Predictive control based on the singleton fuzzy model*

The control algorithm based on inversion of the singleton fuzzy model, the predictive control scheme and the adaptive fuzzy control scheme are tested in simulations of the pressure system. The first-principles model of the process given by the nonlinear differential equation Eq. (7.45) is used for simulation purposes. This simulation can be called realistic, since the model is highly nonlinear (as is the process itself), including the rate and level constraints on $u(\tau)$. There is a significant model–plant mismatch (in fact the fuzzy model approximates the real process better than the analytical model), and the process output is corrupted with sensor noise in the same range as in reality. The reference signal contains several steps of different amplitudes in different operating regions, in order to verify the con-

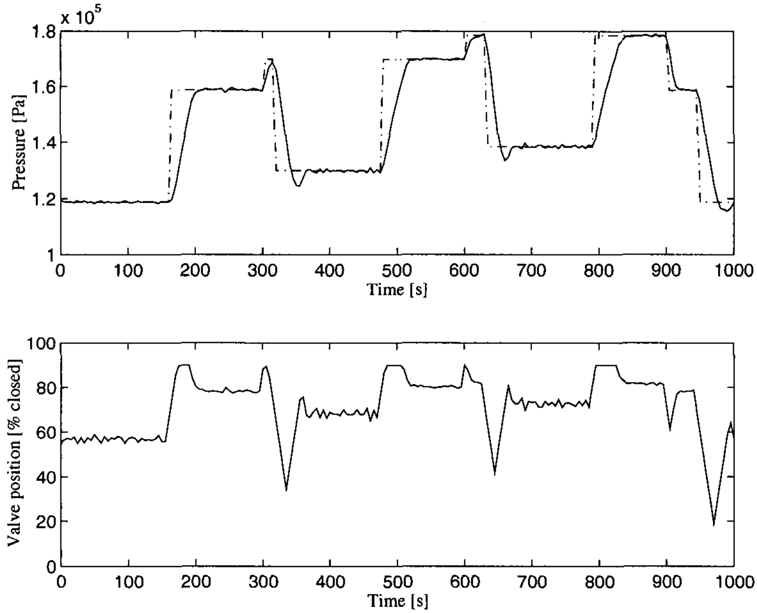


Fig. 7.16 Fuzzy controller based on inverted fuzzy model.

troller's capability to cope with the process nonlinearity. The predictive control is included in an IMC scheme in order to cope with model–plant mismatches and reduce the effect of noise disturbances. Figure 7.16 shows the simulation results with the controller based on the inverse model. The inverse model-based control cannot cope with the rate constraints. The inversion generates larger changes in the control actions than allowed, and the constraint imposed by the rate limiter results in undesired overshoots. Figure 7.17 shows the results obtained using the combination of predictive and inverse model control. The prediction and the control horizons of 3 steps (15 s) were used. The branch-and-bound optimization algorithm is used in the predictive control scheme. The application of B&B to predictive control is presented in Sec. 10.1. A discretization of the possible control actions is necessary to apply this technique. Therefore, the change of the control input is discretized in three levels: $\Delta u(\tau) \in \{-10, 0, 10\}$. Note that by using the combined control scheme the overshoots are eliminated. The sum squared error between the pressure output and the desired references is decreased by 69%, when compared to inverse control, due to the predictive means of control.

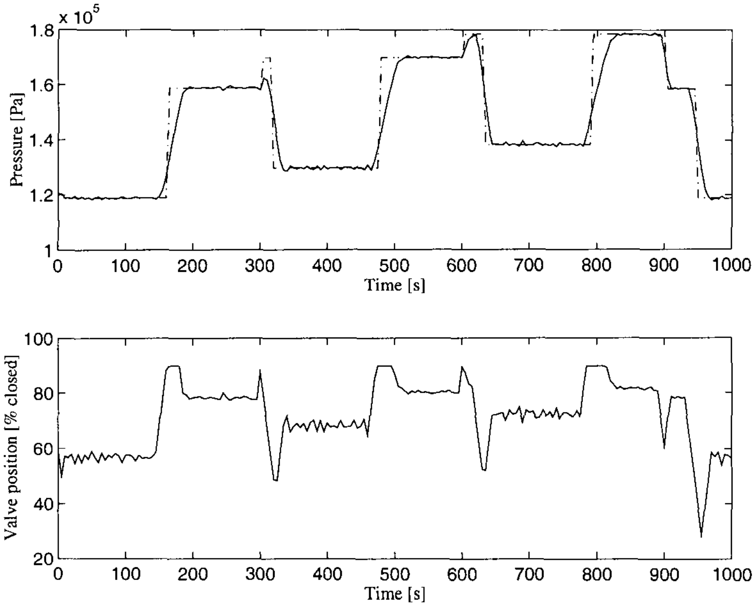


Fig. 7.17 Combination of predictive and inverse model control.

7.6.3 *Adaptive control*

The on-line adaptation mechanism presented in Sec. 7.4 is tested using an external disturbance. The flow-rate is increased from 3.75×10^{-4} to 5×10^{-4} [m^3s^{-1}] at time $t = 400$ s. Figure 7.18 compares the system outputs with controller adaptation (solid line) and without controller adaptation (dotted line). After a short period of adaptation (about 30 s) the adaptive controller again follows the reference (dashed-dotted line), while the fixed controller exhibits a constant offset. This figure shows that the adaptation of the plant model can cope with changes in the plant parameters. Let the consequent parameters of the rules presented in Table 7.2 be denoted by $c^k(\tau)$, for the kth rule, $k = 1, \ldots, 16$. The evolution of these consequent parameters is shown in Fig. 7.19, and it illustrates the local nature of the model. In fact, some of the rule consequents have been adapted immediately after the disturbance at time $t = 400$ s and some others later, as the system dynamics evolving through the input-state space activates the corresponding rules. As shown in Fig. 7.19b, some parameters are adjusted only after a sudden step-like change of the reference, which occurs at time $t = 600$ s. The forgetting factor is set to $\lambda = 0.98$ and the covariance matrix is initialized at $\mathbf{R} = 100 \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The covariance matrix $\mathbf{R}$ is automatically reset each 100

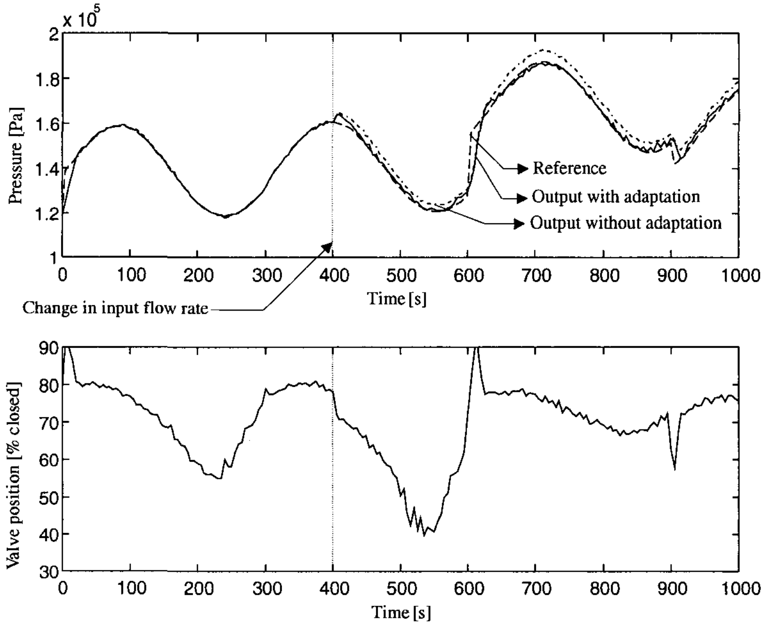
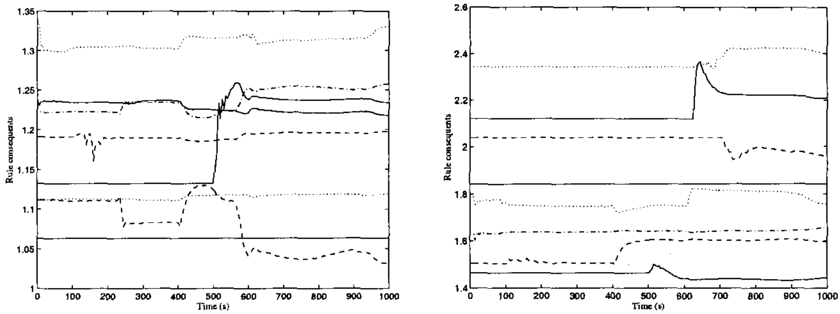


Fig. 7.18 Adaptive control of the pressure system.

samples to guarantee permanent adaptation of the fuzzy model.

7.6.4 *Predictive control based on the affine TS fuzzy model*

The control algorithm based on the inversion of an affine TS fuzzy model and the predictive control scheme based on this type of models are also tested in simulations of the pressure system. The simulation conditions are exactly the same as the ones used for the singleton fuzzy model presented in Sec. 7.6.2. The nonlinear internal control scheme is again used to cope with model–plant mismatch and noise disturbances. Figure 7.20 shows the simulation results with the controller based on the inverse model. The behavior of the system is generally good, but an overshoot is found for low pressure values due to a bigger model–plant mismatch at this region (see the validation of the model presented in Fig. 7.15d). The predictive control scheme based on inverted affine TS models uses the branch-and-bound algorithm for optimization with $\Delta u(\tau) \in \{-10, 0, 10\}$, as it is done for the singleton fuzzy model. The results of this control scheme are shown in Fig. 7.21. The control horizon and the prediction horizon are chosen as $H_c = 2$ and $H_p = 4$, respectively. Due to the relatively large model–plant mismatch referred to before, it is not possible to eliminate the overshoot with the predictive control scheme as



(a) Consequent parameters $c^1(\tau)$ to $c^8(\tau)$ (b) Consequent parameters $c^9(\tau)$ to $c^{16}(\tau)$

Fig. 7.19 Evolution of the fuzzy model consequents $c^k(\tau)$ for the run shown in Fig. 7.18.

it was when the inversion of the singleton model was combined with a predictive control scheme (see Fig. 7.17). However, the sum squared error between the pressure output and the desired references is decreased again, this time by 68%, which is a very similar value to the one obtained using the singleton fuzzy model. The results obtained with the TS fuzzy model confirm the importance of having an accurate model to be used in the control scheme.

7.7 Fuzzy compensation of steady-state errors

One disadvantage of the IMC scheme is that the linear feedback used can deteriorate the closed loop dynamics in the presence of highly nonlinear systems. This section describes another method called *fuzzy compensation*, introduced in (Sousa, Babuška, Bruijn and Verbruggen 1996), that can be used to compensate for steady-state errors, based on the information contained in the model of the system. A fuzzy set for the steady-state error is defined for this purpose, and it determines the degree of activation of the fuzzy compensator. The introduction of this fuzzy set allows for the change of the compensation action, from an active to an inactive state, in a smooth way. In addition, fuzzy compensation also depends on the current system state. Taking the local derivative of the model with respect to the control action, it is possible to achieve compensation with only one parameter to be tuned (similar to the integral gain in a PID controller). Thus, fuzzy compensation makes explicit use of a nonlinear model of the process. This section describes the method in detail, and an application example is presented afterwards. For the sake of simplicity, the method is presented for nonlinear discrete-time

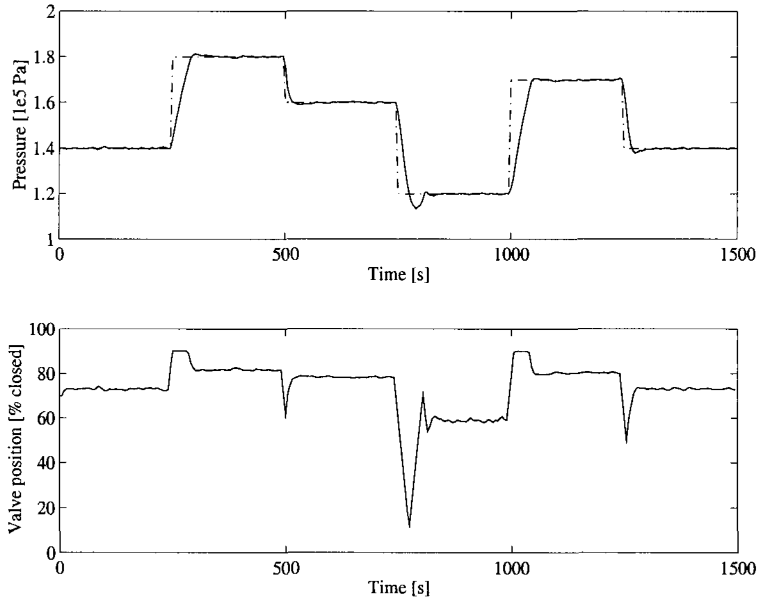


Fig. 7.20 Fuzzy controller based on inverse fuzzy model.

SISO systems, but it can be extended for MIMO systems.

7.7.1 *Derivation of fuzzy compensation*

In this section it is convenient to delay the model by one step for simplicity of notation. The discrete-time SISO regression model of the system under control is then given by

$$y(\tau) = f(\mathbf{x}(\tau - 1)), \tag{7.47}$$

where $\mathbf{x}(\tau - 1) = [y(\tau - 1), \ldots, y(\tau - p), u(\tau - 1), \ldots, u(\tau - m)]$ is the state containing the lagged model outputs and inputs given by $y(\tau - 1), \ldots, y(\tau - p)$ and $u(\tau - 1), \ldots, u(\tau - m)$, respectively.

Fuzzy compensation uses a correction action called $u_c(\tau)$, which is added to the action derived from a (model-based) controller, $u_m(\tau)$, as shown in Fig. 7.22. The total control signal applied to the process is thus given by,

$$u(\tau) = u_m(\tau) + u_c(\tau). \tag{7.48}$$

The controller in Fig. 7.22 can be any controller that can control the system, such as a predictive controller. Taking into account the noise and a (small) offset

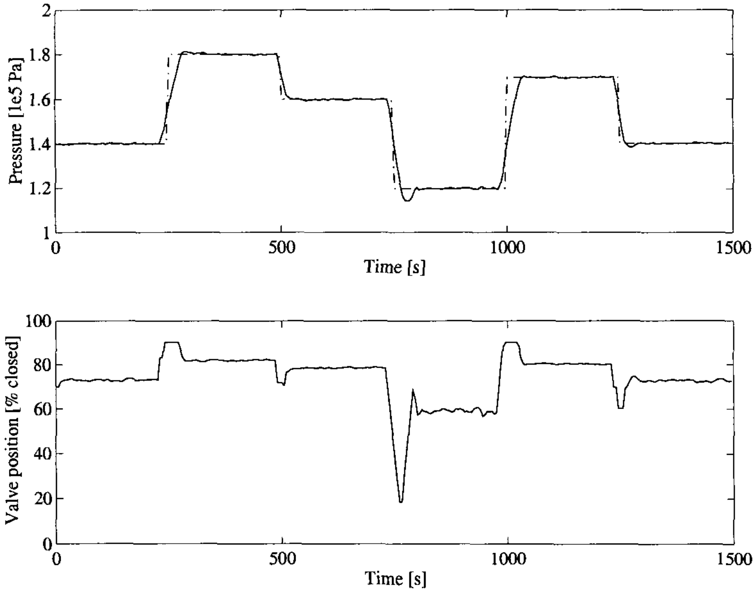


Fig. 7.21 Combination of predictive and inverse model control.

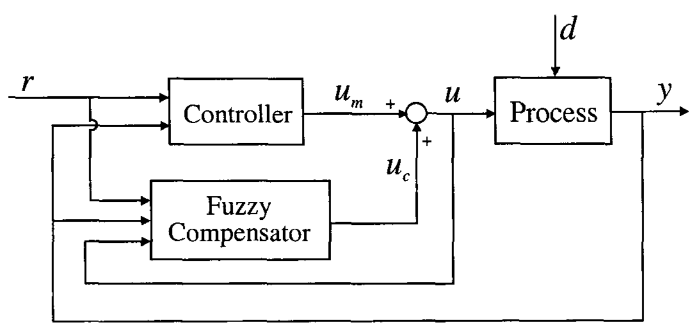


Fig. 7.22 Fuzzy model-based compensation scheme.

error, a fuzzy set SS defines the region where the compensation is active, see Fig. 7.23. The error is defined as $e(\tau) = r(\tau) - y(\tau)$, and the membership function $\mu_{SS}(e(\tau))$ is designed to allow for steady-state error compensation whenever the support of $\mu_{SS}(e(\tau))$ is not zero. The value of b that determines the width of the core SS should be an upper limit of the absolute value of the possible steady-state errors. Fuzzy compensation is fully active in the interval $[-b, b]$. The support of $\mu_{SS}(e(\tau))$ should be chosen such that it allows for a smooth transition from enabled to disabled compensation. This smoothness of SS induces smoothness on the fuzzy compensation action $u_c(\tau)$, and avoids abrupt changes in the control action $u(\tau)$.

The compensation action $u_c(\tau)$ at time τ is given by

$$u_c(\tau) = \mu_{SS}(e(\tau)) \left(\sum_{j=0}^{\tau-1} u_c(j) + K_c \, e(\tau) \, f_u^{-1} \right) , \tag{7.49}$$

where $\mu_{SS}(e(\tau))$ is the error membership degree at time τ, K_c is a constant and

$$f_u = \left[\frac{\partial f}{\partial u(\tau - 1)} \right]_{\mathbf{x}(\tau-1)} \tag{7.50}$$

is the partial derivative of the function f in Eq. (7.47) with respect to the control action $u(\tau - 1)$, for the present state of the system $\mathbf{x}(\tau - 1)$. Comparing Eq. (7.49) with a classical integral action, there are two new terms: $\mu_{SS}(e(\tau))$, whose effect is already described, and the term in Eq. (7.50), which gives the sensitivity of the model for a variation in the control action. In linear systems, this term is constant and is incorporated in K_c, but for highly nonlinear systems, the compensation can be largely improved by taking this factor into account. As the partial derivative in Eq. (7.50) increases, the system becomes more sensitive to changes in the control actions, and a smaller compensation action is demanded. The contrary is also valid. Therefore, the inverse of Eq. (7.50) must be considered in the compensation action Eq. (7.49). For linear systems, the term in Eq. (7.50) is constant and the parameter K_c multiplied by the term f_u^{-1} can be seen as the integral gain in a PID controller. The parameter K_c must be properly tuned. Its value should be chosen such that the steady-state error decreases as fast as possible without oscillations in the response of the system. These oscillations can occur if the fuzzy compensation action is too large, resulting in a new error $e(\tau + 1)$ of opposite sign from the previous $e(\tau)$.

When the model of the system f is available, the partial derivative in Eq. (7.50) can be easily computed, providing that the system is differentiable. However, some black-box modeling techniques derive global models which are a collection of piece-wise linear models. For these type of models, the derivative is not defined at the transients of the linear parts of the model. However, it is possible to define a pseudo-derivative for these points given by the mean value of the left and the right derivatives. These two derivatives exist because they are derived from linear function approximations of the nonlinear system. Thus, the derivative at these

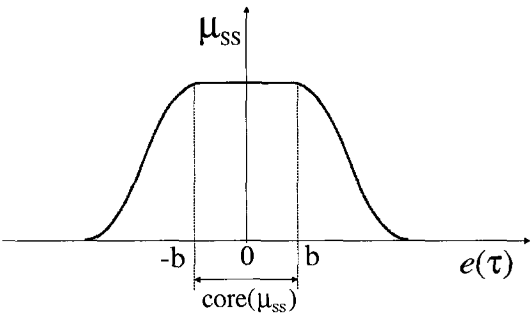


Fig. 7.23 Definition of the fuzzy boundary SS where fuzzy compensation is active.

points can be computed as,

$$\left[\frac{\partial f}{\partial u(\tau-1)}\right]_{\mathbf{x}(\tau-1)} = \frac{\left[\frac{\partial f}{\partial u(\tau-1)}\right]_{(\mathbf{x}(\tau-1))^{+}} + \left[\frac{\partial f}{\partial u(\tau-1)}\right]_{(\mathbf{x}(\tau-1))^{-}}}{2}.$$

This approximation does not deteriorate the control performance significantly, as it can be seen from the simple example of a nonlinear system with dead-zone presented in the next section.

7.7.2 *Application to a system with dead-zone*

As a test case, fuzzy compensation is applied to a nonlinear system with a dead-zone. Assume that the system can be approximated by a first-order discrete time dynamic model $\hat{y}(\tau+1) = f(y(\tau), u(\tau))$. The model of the system is illustrated in Fig. 7.24. Two nonlinearities are present: a dead-zone for the control actions u between -0.4 and 0.2, and saturation levels. The disturbance d in Fig. 7.24 is given by the sum of normal white noise with a random small constant value changing at each step in the reference, in order to simulate different model–plant mismatches.

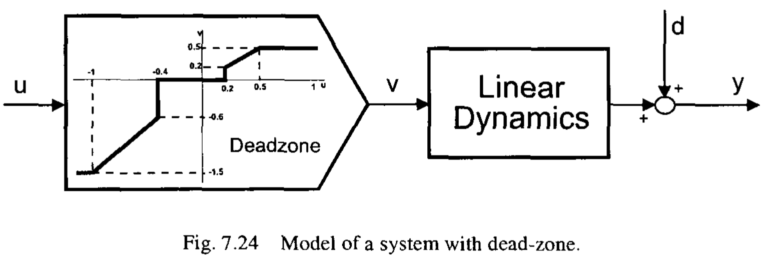


Fig. 7.24 Model of a system with dead-zone.

Table 7.3 Fuzzy model with singleton consequents.

				$u(\tau)$			
$y(\tau)$	-1	-0.401	-0.399	0.199	0.201	0.5	1
-7	-7.01	-5.60	-5.40	-5.10	1.80	2.39	2.90
3	-6.20	-5.59	-5.10	0.90	2.40	2.60	3.00

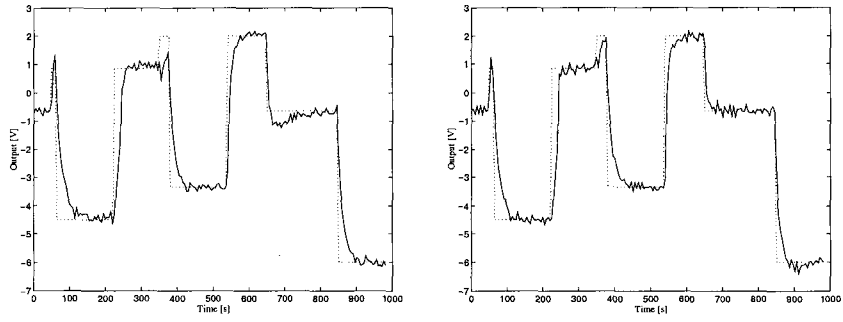
The process is simulated to generate input–output data and a fuzzy model with singleton consequents, as described in Sec. 5.2.2, is derived for the system. The model of the process is described by the lookup table presented in Table 7.3. The values for the output $y(\tau)$ and the control input $u(\tau)$ represent the cores of fuzzy partitions using triangular membership functions. The values in the table are the fuzzy singleton consequents for the predicted model output $\hat{y}(\tau + 1)$. The range of the output of the system is $y(\tau) \in [-7, 3]$, and the range of the control actions is $u(\tau) \in [-1, 1]$. Only two values are needed for the process output $y(\tau)$ because the system is linear with respect to this variable, and thus, the linear interpolation between the points in the table completely describes the system. The cores of the input $u(\tau)$ coincide with the nonlinearities of the process. The simulated system uses a simple inverted model control technique, as presented in Sec. 7.2. To accomplish the inversion of the fuzzy model, the points, where the dead-zone nonlinearity occurs, are slightly changed. Hence, the control action -0.4 is divided in -0.399 and -0.401, and $u(\tau) = 0.2$ is divided in 0.199 and 0.201.

In order to eliminate steady-state errors, standard nonlinear IMC and fuzzy compensation, using the control scheme shown in Fig. 7.22, are added to the controller based on the fuzzy model. The parameter K_c is chosen equal to 5 and the membership function SS is given by

$$\mu_{SS}(e(\tau)) = \begin{cases} 0 & \text{when } |e(\tau)| > 1, \\ 1 & \text{when } |e(\tau)| < 0.5, \\ 2\,e(\tau) + 2 & \text{when } -1 < e(\tau) < -0.5, \\ -2\,e(\tau) + 2 & \text{when } 0.5 < e(\tau) < 1\,. \end{cases}$$

The value of b is then 0.5 to define an interval $[-b, b]$ which contains the possible regions where the system presents steady-state errors. The interval $[-1, 1]$ is chosen for the support of SS, allowing for a smooth transition in the compensation control actions $u_c(\tau)$ as desired. The results obtained are shown in Fig. 7.25.

The advantage of using fuzzy compensation is clearly seen at the region from 350 s to 400 s, for instance, where the response is faster than the one resulting from the application of the IMC scheme. The sum square error and the sum of the absolute error between the reference and the output of the system improve



(a) Nonlinear IMC

(b) Fuzzy model-based compensation

Fig. 7.25 Compensation of steady-state error.

both by about 10%. However, comparing the figures, it seems that sometimes the fuzzy compensation scheme is less robust with respect to noise. This assumption is confirmed when the parameter K_c is slightly increased. For this situation the system presents small oscillations in some regions, which means that this parameter should be carefully chosen, and proves the lack of robustness of the actual system in certain cases.

7.8 Summary and concluding remarks

Methods of deriving nonlinear controllers based on the inversion of fuzzy models have been presented. The methods benefit from the convenient mathematical structure of certain types of rule-base fuzzy models in order to invert them. Section 7.2 presented the inversion of singleton fuzzy models and Sec. 7.3 described the inversion of Takagi–Sugeno fuzzy models affine in the control action $u(\tau)$. Both inversions are *exact* in analytical terms and computationally very fast, allowing for their use in systems with small sampling times, and for applications in real-time control. Note that the inversion of singleton fuzzy models can only be performed for SISO systems, which can constitute a significant drawback. As the TS fuzzy model must be constrained to be affine on $u(\tau)$ in order to be invertible, the resultant model accuracy is usually reduced.

The inverted fuzzy models obtained can be used in an open-loop configuration. If an ideal model of the process is available and both model and controller (inverse model) are input–output stable, the control is *perfect*, and input–output stable. This 'ideal' control configuration cannot be directly applied in practice because the model is never a perfect mapping of the system, resulting in model–plant mismatches. Moreover, disturbances are usually present in the system, and some variables of the process (more often the control actions) can be subject to level and/or rate constraints. Model–plant mismatches and disturbances are reduced by using the nonlinear internal model control scheme. In the presence of significant model–plant mismatches due to permanent or temporary changes in the operating conditions, the model can be adapted on-line in order to cope with these phenomena. An adaptation algorithm based on recursive least-squares is presented in Sec. 7.4, where the singleton fuzzy model is adapted. The adaptation is performed such that the invertibility of the model remains valid, and the scheme can be used for control purposes.

Level and rate constraints on the input variables of the model can be coped with by utilizing the inverse model in a predictive control scheme, as presented in Sec. 7.5. The resultant optimization problem is usually non-convex, and algorithms to reduce the computational time by using discrete optimization techniques are proposed in Chapter 10. Compared to conventional fuzzy logic control, the controllers developed by using fuzzy models demand much less tuning effort, although some experimentation and iterative tuning may be required in the modeling phase. However, once a fuzzy model of the process is available, it can be directly used in the control scheme. The application example presented in Sec. 7.6 demonstrates the control performance and the computational aspects of the described algorithms. It is shown that the predictive control scheme can prevent overshoots and reduce the sum squared error between the output of the system and the desired reference.

One disadvantage of the IMC scheme is that it uses a linear filter for the model/plant error, which can slow down the response of the system, and introduce undesired overshoots. A different solution to dealing with model–plant mismatch is to use fuzzy compensation, which deals with steady-state errors resulting from model–plant mismatches and disturbances. However, fuzzy compensation can introduce undesired oscillations, although the problems concerning the speed of the response and overshoots are usually overcome. Experiments have shown that this control scheme is, in general, less robust then IMC, since the gain parameter requires careful tuning, due to the sensitivity of the method to this parameter.

This page is intentionally left blank

Chapter 8

Performance Criteria

The specification of performance criteria for designing control systems is considered in this chapter. We assume, without loss of generality, that a general feedback control system is studied, as was discussed in Chapter 1 (see Fig. 1.1). In the design of control systems, design specifications usually represent a translation of the main goal in control design, *i.e.*, the outputs $\mathbf{y}$ should be as close as possible to respective pre-specified references $\mathbf{r}$, suppressing the influence of the disturbances. This goal must be accomplished despite the fact that $\mathbf{u}$ or its change are limited due to some physical constraints present in the system. Several examples of constraints in control actions can be given, *e.g.*the flow rate has its maximum value when a valve is fully open, or the opening of a valve should be kept small to save energy.

Design specifications fulfilling the design goals and objectives for the controller design must be specified. For nonlinear systems, three main objectives are required.

(1) Stability of the overall system.
(2) Performance regarding the accuracy and the speed of the system's response.
(3) Robustness to disturbances and dynamics that are not modeled.

The design specifications are usually combined in an optimal control problem, where several design criteria can be aggregated using different approaches. Design specifications are discussed in Sec. 8.1, where systematic approaches for designing linear control systems are presented, and general procedures for deriving controllers in the presence of nonlinear systems are discussed.

Particular attention is devoted to performance specifications, because they are directly related to the general control objectives. Classical performance specifications are presented in Sec. 8.2. Performance specifications are formalized in performance criteria. These performance criteria are expressed by the size of certain signals of interest. There are different ways of defining the size of a signal, given by different norms or semi-norms for signals. An overview of the classical performance criteria using norms and semi-norms of signals and/or systems for defining the performance criteria is given in Sec. 8.3.

Classical design specifications are specified by using performance criteria, which are based on norms or semi-norms. Sometimes, however, it is preferable to define informal design goals such as 'the step response from the reference signal to the output should not overshoot too much' or 'the sensor noise should not cause u to be too large', which may better describe the control goals. These types of control goals can be formally translated to performance criteria using fuzzy logic theory. Fuzzy performance criteria are presented in Sec. 8.4 before the concluding remarks in Sec. 8.5.

8.1 Design specifications

In general, the *design goals* also called *design objectives* for controller design are expressed by *design specifications*. These can have different forms, and are usually related to the architectures or configurations of the respective control systems. As an example, consider, for instance, an air-conditioning system, where the global design goal can be stated as obtaining and maintaining 'human comfort'. This goal must be translated in terms of temperature, humidity, ventilation and noise. Stating, for instance, that 'the temperature should be around 20 °C' is already a design specification, because the control goal is specified for a certain variable. The simplest example is to consider just one goal, such as the minimization of the error between a given reference and the output(s) of the system. In this case, a single cost or objective function is optimized. Often, however, several goals are simultaneously considered and a multicriteria optimization approach must be applied, where the controller must perform well mutually on all these goals. Several criteria can be combined in a single cost function as in the *optimal control* paradigm.

A clear distinction between *design specification* and *design criterion* is usually not utilized in control design, and especially for linear time-invariant systems both terms are used interchangeably. In this chapter the term *design specifications* is reserved for the imprecise design goals and objectives required by the control designer for the variables under control, and the term *design criterion* is used for the formal or mathematical description of the design specifications. The design specification stated as 'the overshoot must be small', for instance, is translated to the precise design criterion: 'the overshoot ϕ_{OS} must be smaller than 5%' (see the definition of overshoot in Eq. (8.7)). This section presents design specifications and criteria. The possible combination of design criteria for the design of controllers is discussed. First, control design specifications for linear systems are presented. Second, a generalization for nonlinear systems is discussed.

8.1.1 *Design specifications for linear systems*

For linear systems, the design specifications can be translated to design criteria in a systematic way. The design criteria can be specified either in time or in frequency domain. Quantitative specifications of the closed loop system are established, and a controller meeting these specifications can be designed. The first problem posed is the *feasibility* problem, *i.e.*, determining whether all design specifications can be simultaneously satisfied.

A design specification is translated to a design criterion J_j, which is dependent on different variables of the system, such as the control actions $\mathbf{u}$, the outputs of the system $\mathbf{y}$, the states $\mathbf{x}$ and the disturbances $\mathbf{d}$. When the design criteria are defined for different variables, *all* of them must be satisfied, *i.e.*, the problem must be feasible. This approach is sometimes called *multicriteria optimization* (Boyd and Barret 1991). In this approach, a tradeoff between the separate parts of the criteria is made, in order to find the possible solutions. Note however, that there is no ordering or priority among the design criteria in this approach. Therefore, several solutions can be obtained.

Another approach that is more often utilized is *optimal control*. In this approach all the design criteria J_j must be translated to functions of only one variable, usually the control actions $\mathbf{u}$. Moreover, an ordering of several criteria must be given, and a unique solution of the optimal problem is obtained. Let $\mathbf{v} \in V$ be a general variable under optimization. Each design criterion is thus translated to a function of this variable represented by $J_j(\mathbf{v})$. The combination of all the design criteria is given by the cost (objective) function

$$J(\mathbf{v}) = f(J_1(\mathbf{v}), \ldots, J_n(\mathbf{v})) , \tag{8.1}$$

where n is the number of criteria defined. In the following paragraph, the two most widely used methods of combining criteria are presented after a discussion of a formal definition of optimal control.

8.1.1.1 *Optimal control problem*

The general form of an optimization problem, usually known as nonlinear constrained optimization problem is defined as

$$\begin{aligned} &\min_{\mathbf{v} \in V} \; J(\mathbf{v}) \\ &\text{subject to} \;\; g_i(\mathbf{v}) \leq 0, \quad i = 1, \ldots, l; \end{aligned} \tag{8.2}$$

where the objective function $J(\mathbf{v})$ is defined as before, the constraint functions $g_i(\mathbf{v})$ are real-valued scalar functions, $\mathbf{v} \in V$, and l is the number of constraints. The constraints in a system can be present for the control actions $\mathbf{u}$, state variables $\mathbf{x}$, outputs of the system $\mathbf{y}$, or changes in these variables. Note that all the constraints must be expressed in the constraint functions g_i in the optimal control formulation, depending on the chosen variable under optimization $\mathbf{v}$.

As an example, let the variable under optimization be the control actions, *i.e.*, $\mathbf{v} \triangleq \mathbf{u}$. Let the design criterion be given simply by the error between the desired reference and the predicted outputs using the model of the system: $J(\mathbf{u}) = \mathbf{r} - \mathbf{y}$. Consider a regulation problem, where the reference is constant. Thus, in this case it is sufficient to have a function relating $\mathbf{y}$ to $\mathbf{u}$,

$$\mathbf{y} = \mathbf{f}(\mathbf{u}), \tag{8.3}$$

in order to solve the optimization problem. As this function is actually a part of the model of the system, this problem is quite trivial. Unfortunately, this formulation is not always so simple.

The constraints considered in $g_i(\mathbf{v})$ are usually known as 'hard' constraints, contrary to the 'soft' constraints. Each criterion J_j has an optimal value, if only that specific criterion is considered. Therefore, a trade-off between the several design criteria for a suitable design of a control system is 'searched'. Thus, the specification of $J(\mathbf{v})$ determines the trade-off between the several criteria. This is generally done interactively, often by repeatedly adjusting the weights in a *weighted-sum* or *weighted-max* objective and evaluating the resulting optimal design. These two methods to combine design criteria are presented in the next paragraphs.

8.1.1.2 *Weighted-Sum Objective*

A common method of combining the individual goals translated in a design criterion J_j is to add all of them, after they have been multiplied by the non-negative weights λ_j

$$J(\mathbf{v}) = \lambda_1 J_1(\mathbf{v}) + \ldots + \lambda_n J_n(\mathbf{v}). \tag{8.4}$$

The weights assign relative values among the functionals J_j. The objective function as in Eq. (8.4) is called a weighted-sum objective. A typical example of the application of the weighted-sum objective is in model-based predictive control, where the sum-squared error added to a term minimizing the control effort is often used as the cost function (see Appendix A).

8.1.1.3 *Weighted-Max Objective*

Another approach, called *minimax design*, is to form the objective function as the maximum of the weighted functions,

$$J(\mathbf{v}) = \max\{\lambda_1 J_1(\mathbf{v}), \ldots, \lambda_n J_n(\mathbf{v})\}, \tag{8.5}$$

where λ_j are again non-negative weights. The weights are meant to express the designer's preference among the criteria, just as in the weighted-sum objective.

The combination of the several criteria for both methods of constructing the objective function is usually chosen in such a way that they lead to closed-loop convex constraints. If $J(\mathbf{v})$ is a convex function and the constraints are convex, the optimization is a convex programming problem (Gill et al. 1981), which is known to have efficient numerical solutions. Therefore, only convex problems are usually considered in the classical approach, even if the system under optimization is nonlinear. As a final remark, note that even for linear systems it can be quite complex to define the required design specifications. Moreover, the translation of them to design criteria is usually difficult or sometimes even impossible. When this stage is possible to achieve, *i.e.*, the design criteria are all defined, it is still necessary to choose a method to combine them, and to choose the respective weights for the different criteria.

8.1.2 *Design specifications for nonlinear systems*

The procedure for designing linear systems described in the previous section can be applied to nonlinear systems only in the time domain. In general, the response of a nonlinear system to a specific input signal does not reflect its response to a different input signal. Therefore, a description in the frequency domain is not adequate for this type of systems.

In general, it is possible to look for some qualitative design specifications in the operating region of interest. For any type of system (linear or nonlinear) the design specifications can be divided into three main groups (Slotine and Li 1991).

(1) *Stability*, for closed loop system under control, both in local and global sense.
(2) *Performance*, which is described by the accuracy and speed of the time responses for some typical references, such as the step response. For this particular response, the three most used specifications are,
   - rise time,
   - overshoot, and
   - settling time.
(3) *Robustness* to disturbances, measurement noise and model–plant mismatch,

where the system must still be able to satisfy the desired specifications when these effects are present.

Some remarks should be made at this point. Note that stability for nonlinear systems is usually defined in a way that does not cope with persistent disturbances (Slotine and Li 1991). The reason for this is that the stability of nonlinear systems is defined with respect to initial conditions, and only temporary disturbances can be translated to initial conditions. Therefore, robustness is used to cope with persistent disturbances. The three most important design specifications, *i.e.*, robustness, performance and stability, may conflict to some extent, and a trade-off between them is usually required to obtain a good control system.

This book does not explicitly address stability and robustness specifications, *i.e.*, no design specifications are specified concerning these features by themselves. These issues are, however, implicitly considered in some control structures, such as internal model control, presented in Sec. B.1. Hence, only *performance specifications* are explicitly treated in this book. It should be stressed, however, that although stability and robustness are not considered, they can be implicitly present in some performance specifications. This is maybe one of the reasons why rule-based FLC, in the Mamdani's sense (Mamdani 1974), are widely applied in industry and performing so well. Note finally that performance specifications defined for nonlinear systems can be translated to performance criteria and combined into an optimal control problem using the weighted-sum or the weighted-max objective, as is usually done for linear systems. The next section presents classical, *i.e.*, non-fuzzy, performance specifications for linear and nonlinear systems.

8.2 Classical performance specifications

One of the most important steps in the design of a control system is the choice of the performance specifications, which influences the type of controller to be used. Performance specifications, like design specifications, can also be contradictory by themselves. Hence, when performance specifications are translated to performance criteria, a trade-off between the different criteria must also be made, in order to find a suitable controller. Usually, the performance specifications are divided into the following groups:

(1) input/output (I/O) specifications, related to the effect of the control actions $\mathbf{u}$ on the system's outputs $\mathbf{y}$,
(2) regulation specifications, measuring the effect of the disturbances $\mathbf{d}$ and $\mathbf{d}_m$ on $\mathbf{y}$, and

(3) actuator effort of the control actions $\mathbf{u}$.

Sometimes, the combined effect of disturbances and control actions on the output is also considered. The following sections describe each of these three groups of performance specifications in more detail.

8.2.1 *I/O specifications*

It is usual to express specifications on the system outputs $\mathbf{y}$ in terms of a given input response. Some of the most used specifications for linear systems are made in terms of the step response of the process $\mathbf{P}$. Step responses give a good indication of the performance of the controlled variable to command inputs that are constant for long periods of time and occasionally change quickly to a new value (new set-point). Let $h(\tau)$ denote the unit step response of the SISO mapping describing a process $\mathbf{P}$. A SISO system is considered for the sake of simplicity, but the next definitions are also valid for MIMO systems. Note that for nonlinear systems, different steps of the system present different behaviors. Thus, several working points of the system must be considered and the specifications described in the following must be done for all these working points. In linear systems this procedure is simplified and only the unit step response needs to be considered. The performance specifications defined in the following must thus be applied for the several working points when a nonlinear system is considered. Note that the following specifications are defined for discrete or discretized systems.

A common specification for step responses is to assure *asymptotic tracking*, *i.e.*, a zero steady-state error for the system, which can be translated as

$$\phi_{\text{at}}(\mathbf{P}, A_P) \triangleq \lim_{\tau \to \infty} A_P h(\tau) = A_P \,, \tag{8.6}$$

where $A_P \in \mathbb{R}$ is the amplitude of the step.

Both the *overshoot* and the *undershoot* are defined as functions of $\mathbf{P}$. The overshoot is defined as

$$\phi_{\text{os}}(\mathbf{P}, A) \triangleq \sup_{\tau \geq 0} A_P(h(\tau) - 1) \,, \tag{8.7}$$

and the undershoot as,

$$\phi_{\text{us}}(\mathbf{P}, A_P) \triangleq \sup_{\tau \geq 0}(-A_P h(\tau)) \,. \tag{8.8}$$

The *rise time* and *settling time* can be defined in different ways. In general, the rise time is defined as

$$\phi_{\text{rise}}(\mathbf{P}) \triangleq \inf\{\tau^* \mid A_P h(\tau) > \alpha_P A_P, \;\; \tau \geq \tau^*\}, \tag{8.9}$$

where a common value for the parameter is $\alpha_P = 0.8$. The settling time is given by

$$\phi_{\text{set}}(\mathbf{P}) \triangleq \inf\{\tau^* \mid |A_P h(\tau) - A_P| < \epsilon, \ \tau \geq \tau^*\}, \tag{8.10}$$

where the parameter ϵ is usually set to 0.05 or 0.02. Figure 8.1 presents an example of a step with amplitude $A_P = 1$, where the overshoot, the undershoot, the rise time with $\alpha_P = 0.8$, and the settling time with $\epsilon = 0.05$ are illustrated. Other specifications normally used for the step response of linear systems are the

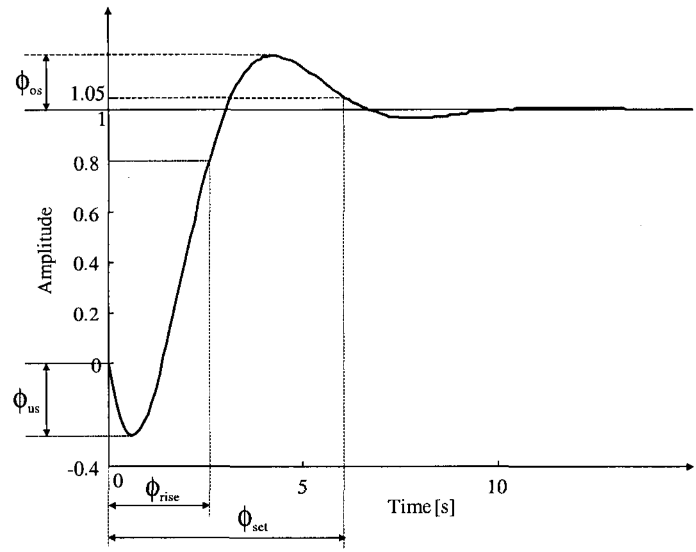


Fig. 8.1 Example of several I/O specifications.

general step response envelope specification, the *general response-time functional* or the *step response interaction*. The readers interested in these specifications are referred to Boyd and Barret (1991).

Step response specifications are suitable for systems where the references to be followed are constant for long periods and change abruptly to new values after those periods. However, typical command signals can be more diverse, changing frequently in a way that is not completely predictable. For these systems, the goal is to have some system variables that follow or track a (continuously) changing reference. Usually, the outputs $\mathbf{y}$ should track the respective references $\mathbf{r}$ with small errors, ideally zero. The errors are thus defined as the difference between the references to be followed and the outputs of the system under control as

$$\mathbf{e}(\tau) = \mathbf{r}(\tau) - \mathbf{y}(\tau) \,. \tag{8.11}$$

Some norms of these error signals, as their root-mean square values, the average-absolute norm or the ∞-norm (peak), are commonly used as performance criteria for control purposes. The definitions of these performance criteria are given in Sec. 8.3.

8.2.2 *Regulation specifications*

This type of specifications considers the effect of the disturbances $\mathbf{d}$ and $\mathbf{d}_m$ in the outputs of the system, assuming that the control signals $\mathbf{u}$ are equal to zero or constant. This formulation is useful for linear systems, where the effects of different inputs can be studied separately and summed up afterwards, due to the superposition principle. Ideally, the effect of the disturbances on the output should be as small as possible.

For linear systems, some typical performance specifications are usually considered. The simplest case is to consider the disturbances constant, and to require that the disturbances should be asymptotically rejected, *i.e.*, the effect of the disturbances should converge to zero. When the disturbances can be described by a stochastic process, it is usual to require that the root-mean square (see the definition in Sec. 8.3.1) of the obtained outputs is smaller than a certain constant value. Another common regulation specification in the frequency domain is the classical minimum regulation bandwidth, which is defined as the largest frequency below which the disturbance is largely damped. A detailed description of regulation specifications for linear systems can be found in Boyd and Barret (1991).

For nonlinear systems, the effects of the disturbances cannot be studied separately from the control inputs, because the superposition principle is not valid for these type of systems. Therefore, the specifications dealing with disturbances are in the group of robustness specifications.

8.2.3 *Actuator effort*

The size of the actuator signals is usually limited. Performance specifications must define the proper limits in the control signals or in their variations. The limitations of the actuators can have different reasons, such as the following.

- *Actuator heating.* Excessive heating of an actuator can be caused by large or fluctuating actuator signals, damaging or causing wear to the system. Such constraints can be expressed in terms of a root-mean square norm of $\mathbf{u}$, possibly with weights.
- *Saturation.* The limits of actuator signals should not be exceeded, because the actuators may be damaged. These specifications can be expressed in terms of

criteria defined as a scaled or weighted ∞-norm of $\mathbf{u}$.

- *Power or resource use.* Large and high frequent actuator signals are usually associated with excessive power consumption or resource use. A scaled average-absolute semi-norm of $\mathbf{u}$ is often used to express the criteria fulfilling these specifications.
- *Mechanical or other wear.* Frequent rapid changes in the actuator signal may cause undesirable stresses or excessive wear. These constraints may be expressed in terms of slew rate or acceleration norms of $\mathbf{u}$.

A brief survey of the different performance specifications defined for a given system has been presented in this section. Performance criteria are the translation of performance specifications to a formal description. This translation can be made in classical or fuzzy terms. The next section describes classical performance criteria, while fuzzy performance criteria are presented in Sec. 8.4.

8.3 Classical performance criteria

Usually, the control goals can be expressed in terms of the size of certain signals of interest. For example, tracking error signals, given by the difference between the references $\mathbf{r}$ and the system's outputs $\mathbf{y}$ must be 'small', while actuator signals $\mathbf{u}$ should, normally, not be 'too large'. The criterion describing the performance of the tracking system can be measured, *e.g.*, by the size of the error signal. The size of a signal can be precisely defined using *norms* presented in the next section, which generalize the concept of the Euclidean length of a vector (Boyd and Barret 1991).

8.3.1 *Norms and semi-norms of signals*

Different norms for signals are described in this section. First, the concept of *norm* is defined as follows. Let $v(t)$ denote a time signal in a vector space V. A norm of v, represented by $||v||$ maps the space V to $\mathbb{R}$ and has the following four properties.

(1) $||v|| \geq 0$ (Nonnegativity),
(2) $||v|| = 0 \iff v = 0,$ (Positive definiteness),
(3) $||\alpha v|| = |\alpha| ||v||, \ \forall \alpha \in \mathbb{R}$ (Homogeneity),
(4) $||v_1 + v_2|| \leq ||v_1|| + ||v_2||$ (Triangle inequality).

for any $v_1, v_2 \in V$. If all the properties except the positive definiteness hold, then a *semi-norm* is defined. Several norms of signals are presented in the next paragraphs in both the time and frequency domains, and the physical meaning of each one is described. Note that the signals of interest in a system are usually obtained in a discrete or discretized way. Hence, discrete-to-continuous transformation of these signals using, *e.g.*, a zero-order-hold or a first-order-hold must be applied, so that a certain norm or semi-norm of the signals can be computed.

The most common norms are the 1-norm, the 2-norm and the ∞-norm. These norms can be derived as special cases of a p-norm defined as

$$||v||_p \triangleq \left(\int_0^\infty |v(t)|^p dt \right)^{1/p}, \quad p \geq 1. \tag{8.12}$$

8.3.1.1 *1-norm*

This norm is the integral of the absolute value of a signal $v(t)$,

$$||v||_1 \triangleq \int_0^\infty |v(t)| dt, \tag{8.13}$$

and can be seen as a measure of the total fuel or resource consumption.

8.3.1.2 *2-norm*

The 2-norm of a signal gives the square root of the total energy, and is given by

$$||v||_2 \triangleq \left(\int_0^\infty |v(t)|^2 dt \right)^{1/2}. \tag{8.14}$$

If the system under control is linear, the 2-norm can be computed in the frequency domain using Parseval's theorem, see *e.g.*, Zhou et al. (1996). Note that the 1-norm and the 2-norm are appropriate for transient signals, which decay to zero as time progresses. The same happens for the integral of time multiplied by the absolute error (ITAE) norm defined below. The rest of the norms defined in this section are used for measuring the size of persistent signals.

8.3.1.3 *∞-norm*

One simple interpretation of 'the signal v is small' is that it is small at all times, or equivalently, its maximum or *peak* absolute value is small. The ∞-norm of v is thus the least upper bound (supreme) of the absolute value of a signal, given by

$$||v||_\infty \triangleq \sup_{t \geq 0} |v(t)|. \tag{8.15}$$

The ∞-norm of a signal depends entirely on the extreme or large values the signal takes on. As the ∞-norm depends on occasionally large values of the signal, it is a worst case norm.

8.3.1.4 *ITAE norm*

Sometimes it is useful to introduce a time dependent weight in the norm, given a certain function of time $w(t)$. The most simple example is the integral of time multiplied by the absolute error (ITAE) norm, where $w(t) = t$. The ITAE-norm is defined as

$$||v||_{\text{ITAE}} \triangleq \int_0^\infty t\,|v(t)|dt\,. \tag{8.16}$$

This norm is given by the 1-norm of v weighted by the time. This weight emphasizes the importance of the signal v as time evolves, and de-emphasizes the signal at the beginning of the response. Thus, for this norm the steady-state behavior of the signal is more important than the transient behavior.

8.3.1.5 *Root-Mean-Square*

For signals with finite steady-state power (non-transient signals) it is useful to define a measure that reflects its average size, which is given by the *root-mean-square* (RMS) *value*, defined by

$$||v||_{\text{rms}} \triangleq \left(\lim_{t^* \to \infty} \frac{1}{t^*} \int_0^{t^*} v(t)^2 dt \right)^{1/2} , \tag{8.17}$$

provided that the limit exists. This semi-norm is a classical notion of the size of a signal, and it is widely used in many areas of engineering. Signals with small RMS norms can still present occasional large peaks, if they are not too frequent and do not contain too much energy. The $||v||_{\text{rms}}$ is thus an average measure of a signal. Hence, a signal with small RMS value can still be very large for some time period.

8.3.1.6 *Average-Absolute Value*

The *average-absolute value* is a measure that puts even less emphasis on large values of a signal than the RMS norm, and it is defined by

$$||v||_{\text{aa}} \triangleq \lim_{t^* \to \infty} \frac{1}{t^*} \int_0^{t^*} |v(t)|dt\,, \tag{8.18}$$

supposing that the limit in Eq. (8.18) exists. The $||v||_{\mathrm{aa}}$ semi-norm is useful to measure the average resource used (like fuel), when the resource consumption is proportional to $|v(t)|$.

The comparison of the three (semi-)norms: ∞-norm, $||v||_{\mathrm{rms}}$ and $||v||_{\mathrm{aa}}$, shows that they simply put different emphasis on large and small signal values. The ∞-norm puts all its emphasis on large values, the RMS semi-norm puts less emphasis on signal amplitudes, and the average-absolute semi-norm puts uniform emphasis on all signal amplitudes.

Other (semi)-norms of signals can be defined, but the seven presented are probably the most commonly utilized to measure different characteristics of a signal. The notion of norm of a signal can also be extended to the norm of a system, as explained below.

8.3.2 *Norms of systems*

Let $\mathbf{H}$ be a mapping from a given input u to an output y as in Fig. 8.2. The input can be, *e.g.*, a control action or a disturbance. Note that $\mathbf{H}$ can be a subsystem of the total considered system $\mathbf{P}$.

Fig. 8.2 Input–output mapping of a subsystem.

The notion of norm can be used for the mapping $\mathbf{H}$ as an extension of the *induced norms* usually defined for linear time-invariant (LTI) systems (Zhou et al. 1996). Thus, the *induced p-norm* of a mapping $\mathbf{H}$ is defined as

$$||\mathbf{H}||_{ip} \triangleq \max_{||u||_p<\infty} \frac{||y||_p}{||u||_p}, \tag{8.19}$$

using the definition of p-norm as in Eq. (8.12). A norm for a particular mapping is used when an input signal, *e.g.*a step, sinusoidal or impulse signal, is applied to the system, and the output y is measured. Note that if the system is nonlinear, dividing the system in subsystems does not simplify the analysis because the superposition principle is not valid. Hence, in general it is not possible to measure the effect of a particular control action or disturbance, without taking into account the influence of the remaining inputs (control actions or disturbances). Moreover, different input signals of the same type, *e.g.*different steps or impulses, have in general different responses. Thus, the norms of nonlinear systems have little practical value, except for systems described by first principles, or systems for which some prior knowledge about its behavior is readily available.

A completely different situation occurs when the system under control is linear. The norms defined in Sec. 8.3.1 can be applied, and the discussion presented for each norm can be extended for systems. For linear systems, when many input signals are applied with a probability distribution, the average size of the response can be measured using the expectation with respect to the distribution of the input signals. Another possibility for measuring the size of a linear system when several inputs are considered, is to take the worst case or the largest norm of the response of $\mathbf{H}$ to the given input signals.

Classical performance criteria pretend to translate the performance specifications in formal terms such that the behavior of the system is as close as possible to the desired behavior. Performance specifications are sometimes contradictory, and a compromise between them is then necessary. This is also the case in decision making problems, in which a compromise between competing criteria and requirements should also be obtained. For this type of problems it is useful to fuzzify the criteria and requirements, usually leading to better decisions. Considering the difficulty in translating the performance specifications to performance criteria in order to fulfill the designer requirements, even in the presence of linear systems, it seems reasonable to describe the performance specifications using fuzzy performance criteria.

The definition of fuzzy performance criteria can be easier than classical performance criteria due to the inherent fuzziness present in the performance specifications and the flexibility introduced in the definition of the performance criteria (Sousa et al. 1999). Moreover, the combination of different criteria can be made using different and more general operators than the ones presented in Sec. 8.1.1, by using decision functions to combine the several criteria.

8.4 Fuzzy performance criteria

Classical performance criteria can only be defined using norms or semi-norms. However, sometimes informal design goals better translate the designer's intentions than the precise classical design criteria. In a non-classical approach, the design specifications can be seen as (fuzzy) objectives and (fuzzy) constraints. Some examples are 'the error between the reference and the output should be very small' or 'the control signal should not change too much'. The linguistic terms used in the design specifications such as 'small' and 'not change too much' can be defined by using fuzzy sets. With the introduction of fuzzy sets for defining the goals of a control system, it is possible to use criteria that do not constitute a norm or semi-norm, which generalizes the concept of design goals as used in the classical control theory. This type of goals can, however, be easily combined with norms in a fuzzy decision making environment.

Fuzzy performance criteria must be aggregated in order to find the optimal control actions for a given control system. An approach that transparently translates the objectives and constraints derived from the control design goals of a given system to performance criteria is fuzzy multicriteria decision making. The concept of multicriteria decision making in a fuzzy environment is originally defined as a confluence of decision goals and constraints (Bellman and Zadeh 1970). Both the goals and the constraints are represented by membership functions, which facilitates their aggregation. As goals and constraints are both represented by membership functions in a similar manner, they are usually called fuzzy *decision criteria* in the fuzzy decision making environment.

The use of fuzzy performance criteria in control is presented in more detail in Chapter 9. Each performance criterion is described by a membership function. As an example, let each error in Eq. (8.11) be defined by a triangular membership function, as depicted in Fig. 8.3. Note that this definition is for a certain time step τ, and a confluence of the different errors for all the responses should be made. This procedure is, however, time consuming, and normally not necessary, because

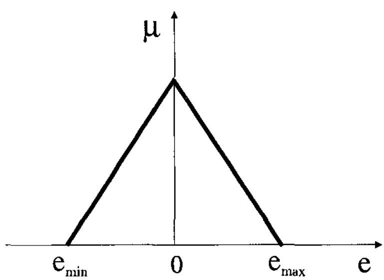


Fig. 8.3 Example of a performance criteria defined for an error, as defined in Eq. (8.11).

it is enough to consider the number of steps necessary to guarantee a good measure of the defined error. For step responses, for instance, this number of steps is equal to the settling time, as defined in Eq. (8.10). In fact, the I/O specifications defined in Sec. 8.2.1 can almost be used straightforwardly as fuzzy performance criteria. Regulation effort and actuator effort are usually seen as fuzzy constraints, and fuzzy sets can be defined for these specifications. Classical performance criteria cannot be directly used as fuzzy performance criteria, but the physical meaning behind their definition is of great interest. In fact, instead of using the norms and semi-norms defined in Sec. 8.3 directly, they can be used as an indication on how to aggregate the different fuzzy sets defining the fuzzy criteria, and also to define the fuzzy sets translating the different criteria. A system that is required to avoid peak errors, for instance, should aggregate the errors over the time horizon penalizing each large error, similar to the peak or ∞-norm as in Eq. (8.15). On the other hand, if only the average of errors is important, and an eventual peak is allowed, the aggregation of criteria must make a sort of average, using a procedure similar to the 1-norm or the 2-norm, see Eq. (8.13) and Eq. (8.14).

The formulation of the control problem as a confluence of (fuzzy) goals and (fuzzy) constraints, can be seen as a generalization of the cost function usually utilized in model-based predictive control. The application of fuzzy performance criteria in MBPC is presented in Sec. 9.2, showing the advantages of generalizing the objective function, usually at the cost of increasing computational time, to derive the optimal control actions.

8.5 Summary and concluding remarks

In classical control theory, design specifications are rigorously defined for linear systems, resulting directly in performance criteria. These specifications are given by several cost functions. The best control actions can be determined by solving an optimal control problem, where design specifications are combined using, *e.g.*the weighted-sum or the weighted max type of functions. Design specifications cope with stability, performance and robustness of the system under control. The characteristics of the model of the system to assure stability and robustness are usually too strict, restricting the type of system for which they can be applied. Therefore, this book does not consider these characteristics explicitly in the design of the developed control systems. However, performance specifications can be defined in such a way that stability and robustness are implicitly considered. This feature is considered when performance criteria are defined for a controlled system.

Classical performance specifications are usually divided into I/O specifications, regulation specifications and actuator effort. The I/O specifications are defined by some measures on a transient response. For a step response, for instance, overshoot, rise time or settling time can be considered. Regulation specifications are defined for disturbances $\mathbf{d}$ and $\mathbf{d}_m$, in order to diminish their effects on the system. Control signals are usually limited by rate or level constraints, which must be considered in the control design.

Performance specifications are defined by performance criteria. in the classical approach, they are usually built by using different norms or semi-norms of the signals of interest. Several norms, such as 1-norm, 2-norm, or ∞-norm are used to measure the size of different signals in the transient regime. The norms most widely used for non transient signals are the root-mean-square and the average-absolute value. The norms of signals can be extended to the norms of systems. However, norms of systems are only normally used in the control design if the system under control is linear. Note that the design goals given by the performance specifications are usually contradictory, and a trade-off between them must be made in order to choose the desired performance criteria.

A different approach is to use fuzzy sets to define the imprecise control design goals. Control objectives defined as fuzzy goals and fuzzy constraints can be combined in the fuzzy decision making environment. This approach translates the objectives and constraints derived from the control design goals of a given system in a transparent way. As in the classical approach, the decision goals and the constraints are defined on relevant system variables.

The formulation of the control problem as a confluence of (fuzzy) goals and (fuzzy) constraints can be seen as a generalization of the cost function usually used in model-based predictive control. Various classical and fuzzy criteria can be used in MBPC, as shown in the next chapter.

This page is intentionally left blank

Chapter 9

Model-Based Control with Fuzzy Decision Functions

Human operators can control complex, nonlinear and partially unknown systems across a wide range of operating conditions, while the conventional linear control techniques often fail or can only be applied locally. Fuzzy logic control, as defined in Sec. 4.2, is one of the most popular techniques for translating human knowledge to control, and has been successfully applied to a large number of consumer products and industrial processes (Terano et al. 1994, Yen et al. 1995, Jamshidi et al. 1997). However, most of these applications of fuzzy control use a *descriptive* approach introduced in the seventies by Mamdani (Mamdani 1974). The operator's knowledge is verbalized as a collection of If–Then control rules that are directly translated into a control algorithm.

Besides direct fuzzy control, in which the control law is explicitly described by If–Then rules, human expertise can be used to define the design specifications. These specifications can be translated to performance criteria by using fuzzy sets, by defining the (fuzzy) goals and the (fuzzy) constraints for the system under control. This procedure is a special approach to fuzzy model-based control, following closely the classical model predictive control design approach, but making use of the fuzzy set theory on a higher level than usually made in FLC, where the fuzzy rules to control the system are given directly from expert knowledge. When fuzzy sets are used to represent the performance criteria, the appropriate control actions are obtained by means of a multistage fuzzy decision making algorithm, as introduced by Bellman and Zadeh (1970). A mixture of conventional fuzzy control with fuzzy representations of design specifications is one the first applications in fuzzy predictive control: automatic train operation using a linguistic description of the system (Yasunobu and Miyamoto 1985). A different approach called fuzzy multiobjective optimal control is presented by Jia and Zhang (1993), but it is quite complex and difficult to implement in real-time. More recently, satisficing decisions have also been used in a similar setting to design controllers (Goodrich et al. 1998). A good survey on model-based approach to fuzzy control and decision making is presented by Kacprzyk (1997). However, the last reference considers only open-loop control applications. In fact, the approach reported is generally computationally intensive, which hampers its application in real-time control. The generalization of fuzzy predictive control as model predictive control with fuzzy decision functions, using the fuzzy decision making approach to select proper control actions, has been introduced by Sousa and Kaymak (2001). This method can be applied to real-time problems with relatively small sampling times. This chapter begins by describing the relation of fuzzy decision making to model-based predictive control in Sec. 9.1. Fuzzy model-based predictive control is explained in Sec. 9.2. Fuzzy goals and fuzzy constraints in a control setting are presented, and an approach to solving the optimization problem for fuzzy criteria defined in different sets is proposed. Note that FDM applied to control considers multistage fuzzy decision making. Section 9.3 presents possible types of fuzzy objective functions for predictive control, briefly discussing the operators to aggregate fuzzy criteria. Two illustrative examples are presented in Sec. 9.4, where the main features of fuzzy decision functions applied to MBPC are shown. Section 9.5 discusses the selection of a decision function from a description of an expert's control strategy. Simulation of a gantry container crane is used as an example, and the fuzzy set theory is used to translate a human operator's control strategy into a mathematical objective function that can be used for optimization. This section also illustrates the use of weighted aggregation operators for aggregation in fuzzy decision making. Concluding remarks are given in Sec. 9.6.

9.1 Fuzzy decision making in predictive control

Although distinct, it is common to present multistage decision making and FDM in control as synonymous. In fact, the control problem is more general, and multistage decision making can also be applied to other fields. This chapter considers multistage decision making applied to control, similar to the approach taken before by several authors (Bellman and Zadeh 1970, Kacprzyk 1997). The relation between a decision making problem and a control problem has been discussed in Sec. 4.1. Recall that when multistage decision making is translated to the control environment, the set of alternatives constitutes the different *control actions* and the *system* under control is a relationship between inputs and outputs. The mapping relating the inputs to the outputs of the system under control is referred to as the *model*. Moreover, fuzzy constraints are defined for several variables present in the system, which can be 'hard' or 'soft' constraints, and the decision criteria (fuzzy goals and constraints) are the translation of the control performance criteria defined in Chapter 8 to the decision making setting.

The systems and models considered for FDM applications follow the definitions presented in Sec. 5.1, Eq. (5.2). However, in the general case, the systems can be time-variant, and not only time-invariant as in Eq. (5.2). Moreover, the system considered in Eq. (5.2) is a deterministic one, while in general the system can be of other types, such as stochastic systems.

An important issue in FDM applied to control is the termination time, which is a generalization of the prediction horizon H_p defined for predictive control in Appendix A. The termination time is assumed to be fixed and specified beforehand, as the prediction horizon in MBPC. However, other types of termination times are possible. A short summary of the different termination times, and possible solutions found in the literature for these problems is presented in the following.

- *Fixed and determined specification time* – a solution of this type of termination time for deterministic systems using dynamic programming is given in (Bellman and Zadeh 1970). Different techniques have been proposed to solve this problem, such as, *e.g.*, branch-and-bound (Kacprzyk 1978, Esogbue et al. 1992), a genetic algorithm (Kacprzyk 1995), or a neural network (Francelin and Gomide 1993). For stochastic systems, two different formulations are usually employed. In Bellman and Zadeh (1970) the optimal control actions are found by maximizing the probability of satisfying fuzzy goals and fuzzy constraints. A different approach is presented by Kacprzyk and Staniewsky (1980), where the optimal control actions are found by maximizing the expected value of the fuzzy decision. Finally, for fuzzy systems, solutions using dynamic programming (Baldwin and Pilsworth 1982), branch-and-bound, interpolative reasoning and a genetic algorithm (Kacprzyk 1997) are proposed.
- *Implicitly specified termination time* – in these systems the process terminates when the outputs reach some pre-specified value. An iterative solution for deterministic systems is introduced by Bellman and Zadeh (1970). A graph-theoretic analysis has also been used to tackle the same problem, but a simpler solution can be derived by using the branch-and-bound approach (Kacprzyk 1978).
- *Fuzzy termination time* – it is sometimes useful to consider a 'softer' definition of the termination time, by allowing its formulation as a fuzzy set, as it was first proposed by Fung and Fu (1977). For deterministic and stochastic systems, solutions using dynamic programming (Stein 1980) or branch-and-bound are possible. These solutions and an extension of the methods referred to for fuzzy systems is presented in Kacprzyk (1997).
- *Infinite termination time* – this type of termination time is used for processes that vary little over a very long time range. Optimal control is sought for this type of processes. The work of Howard (1971) introduces a policy iter-

ation technique, and solves infinite termination time problems using a finite sequence of iterations. A solution for deterministic, stochastic and fuzzy systems can be found in Kacprzyk et al. (1981).

Note that all the solutions proposed are obtained for open-loop control, which hampers the application of the proposed solutions so far for low and medium levels of control in real-time. Kacprzyk (1997) states the following.

> *"We consider (...)* open-loop control. *Unfortunately, not much is known about closed-loop (feedback) control in a fuzzy environment in the optimal control-type Bellman and Zadeh (1970) setting (...)."*

This book addresses this issue, and it studies multistage decision making (control) in a fuzzy environment by considering any type of model in closed-loop control. It assumes that the termination time is fixed and is specified beforehand. As the formulation is done in an MBPC environment, this termination time is the prediction horizon, which is shifted when time evolves. This condition is necessary to allow the application of multistage FDM to MBPC in real-time.

Some applications of fuzzy decision making to close-loop control can be found in the literature. This approach was first developed by Yasunobu and Miyamoto (1985). They included fuzzy control criteria such as safety, comfort, energy consumption and stopping accuracy in a linguistic fuzzy model derived from expert knowledge, and implemented it in a predictive control scheme. This control system has the disadvantages related to using linguistic models derived from expert rules, *i.e.*, it requires the trial-and-error tuning of the rules. However, the predictive fuzzy controller has been used on the Sendai city subway in Japan, and has shown better performance than the previously used controllers. Jia and Zhang (1993) presented a different approach for MIMO systems, but their approach is quite complex and computationally demanding, hampering its application in real-time. Others papers presented by the same and other authors (Chang et al. 1996) confirm these disadvantages. To our best knowledge, and excluding the references where the authors of this book are included, no other applications have been reported.

Only deterministic systems are considered in this book. In fact, most of the physical systems are still modeled using deterministic, time-invariant reasoning. Moreover, nonlinear systems are modeled by using nonlinear modeling techniques. In this book, fuzzy modeling is used to derive nonlinear models, see Sec. 5.2. However, other modeling techniques, such as standard nonlinear regression (Seber and Wild 1989) or neural networks (Hunt et al. 1992), can also be used to derive a model of the system. Fuzzy decision making applied in closed-loop control systems can be extended for time-variant, stochastic and/or fuzzy systems. The application of FDM in control to fuzzy systems is usually quite difficult, because the fuzziness tends to increase for a multistage FDM problem. An interesting fuzzy system using fuzzy arithmetic based interpolative reasoning (FAIR) is presented by Setnes, van Nauta Lemke and Kaymak (1998). For these systems, linguistic fuzzy rules of the Mamdani type with fuzzy numbers as consequents are used in an inference mechanism similar to the Takagi–Sugeno model. Further, both fuzzy and crisp inputs and outputs can be used, and chaining of rule bases is supported without increasing the fuzziness at each step. This type of fuzzy systems can model uncertain processes, for which no present-day modeling paradigm can be used, allowing the generalization of FDM in control for fuzzy systems.

9.2 Fuzzy model-based predictive control

Despite the one-to-one relation between the decision problems and the control problems, the way a control problem is formulated is often different from the way a decision problem is formulated. In decision problems, the alternatives are often evaluated explicitly in terms of the criteria, and the information is aggregated using relatively complicated aggregation operations that are suitable for the nature of the decision problem. When human beings are involved, the decision process may take several iterations, where additional information is asked in each iteration, refining the problem and its solution further and further. In control systems, however, the available time for computations is limited and often not sufficient for complicated calculations. Therefore, the control systems are designed as a static mapping, which maps its inputs directly to its outputs, often with simple computations in between. Much of control engineering is concerned with the specification and the determination of mappings that lead to control results which satisfy the control goals. The optimization which leads to the design of an (optimal) controller then occurs off-line. With the advent of modern computer technology, however, more complex algorithms and more complicated computations have come within the reach of control systems. Consequently, control algorithms that use the additional available computation power have been proposed. One of these methods is the model-based predictive control approach, where a mathematical model of the controlled system is used to predict the future response of the system, and then optimize the control actions accordingly. The optimization can be achieved with a decision making approach, and hence the relation to decision making is apparent in model-based predictive control. In this section, fuzzy decision making is combined with a model-based predictive control scheme to obtain a fuzzy model-based predictive controller.

In the following, Sec. 9.2.1 presents the definition of fuzzy goals and constraints in the control environment. The aggregation of the different criteria for control applications is presented in Sec. 9.2.2, where the set of possible alternatives is discretized in order to find the optimal control actions. The application of FDM to predictive control is presented in Sec. 9.2.3.

9.2.1 *Fuzzy goals and constraints in the control environment*

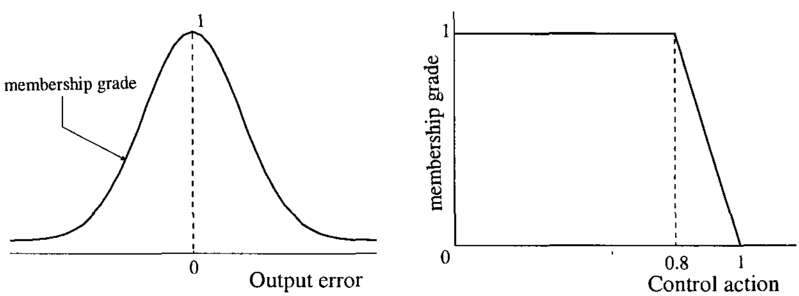
The fuzzy goals G and the fuzzy constraints C can be defined for the control, output or state spaces, or in any other convenient space. Note that usually fuzzy constraints are defined in the control space, and fuzzy goals in state space (Bellman and Zadeh 1970, Kacprzyk 1997). The approach is initially applied to systems with discrete states and a finite number of possible transitions between the states, and has subsequently been extended to systems with continuous states (Gluss 1973). The application of fuzzy decision making in control allows for the generalization of goals and constraints to different spaces. The confluence of fuzzy goals and fuzzy constraints in multistage decision making is similar to the one presented in Sec. 2.3 for fuzzy goals and constraints in different spaces.

A fuzzy set in the appropriate space characterizes both the fuzzy goals and the fuzzy constraints. The goals and constraints are defined on relevant system variables. For example, a common control goal is the minimization of the output errors. The satisfaction of this goal is represented by a membership function, which is defined on the space (universe of discourse) of the output errors. An example is the fuzzy goal 'small output error', defined for a SISO system and shown in Fig. 9.1a. Fuzzy constraints can be defined on the universe of discourse of the control variables $\mathbf{u}$. An example in a SISO system is the 'soft' constraint 'u should not be substantially larger than 0.8', whose degree of satisfaction can be represented by a membership function as shown in Fig. 9.1b. Note that the 'hard' constraints $0 \leq u \leq 1$ are also included in the given membership function.

In control problems, the constraints are usually defined on U, the universe of discourse for the control actions, whereas the goals are usually defined on X, the universe of discourse for the system states. The constraints are often related to physical quantities that cannot be violated, while the goals are often related to targeted or desired values of quantities. As such, it may be more advantageous to treat the goals and the constraints differently, contrary to the symmetric approach of Bellman and Zadeh (1970), discussed in Chapter 2. The qualitative distinction between goals and constraints can be captured in two ways.

(1) The definition of membership functions.
(2) The aggregation method.

A fuzzy goal can be defined in such a way that the membership grade is never zero, unless this is strictly necessary (which would imply that it is a 'hard' constraint). Therefore, the example in Fig. 9.1a uses a membership function of the exponential type, which never becomes zero even if the error is quite large. On the other hand, fuzzy constraints must include the 'hard' constraints, if they are present in the system. For instance, the constraint in Fig. 9.1b does not allow that the control action is outside the range [0, 1], which can be a very useful concept for many real systems. Suppose that the variable u in Fig. 9.1b is a valve opening, where 1 stands for completely open and 0 for completely closed. Hence, the definition of the fuzzy constraint, as is given in Fig. 9.1b, takes these physical limitations into account. In this case, the support of a *fuzzy constraint* represents the 'hard' constraints present in the system. A *fuzzy goal* should be defined so that the membership function can be very low, but never becomes zero, indicating the allowable but not desirable states of the system. This procedure distinguishes goals and constraints in the form of the defined membership functions, but clearly does not affect the confluence of criteria.



(a) Goal: 'small output error'.

(b) Constraint: 'u not substantially larger than 0.8'.

Fig. 9.1 Example of a fuzzy goal and a fuzzy constraint for FDM in control. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

The asymmetry between the goals and the constraints can also manifest itself in the operators selected for the confluence of the two types of criteria.

Assuming that one considers n_g goals $G_1, \ldots, G_{n_g}$ and n_c constraints C_1, $\ldots$, C_{n_c}, each fuzzy goal G and each fuzzy constraint C constitute a *decision criterion* ζ_ℓ, $\ell = 1, \ldots, m$, where $m = n_g + n_c$ is the total number of goals and constraints. Each criterion is defined in the space Φ_ℓ, $\ell = 1, \ldots, m$, which can be any of the various domains used in control.

The confluence of goals and constraints could be obtained by using the hierarchical aggregation scheme from Sec. 3.4. In order to solve the optimization problem in reasonably low time, it is defined in a discrete control space with a finite number of control alternatives. This limitation to digital control is however not too severe, and this methodology can still be applied to a large number of control problems. Therefore, the confluence of goals and constraints is defined for discrete alternatives, in the following. The resulting optimization problem is also addressed.

9.2.2 *Aggregation of criteria in the control environment*

Assume that a policy π is defined as a sequence of control actions for the entire prediction horizon H_p in MBPC,

$$\pi = \mathbf{u}(\tau), \ldots, \mathbf{u}(\tau + H_p - 1) \in \Omega\,, \tag{9.1}$$

where the control actions belong to the set of alternatives Ω. In the general case, all the criteria must be applied at each time step j, with $j = 1, \ldots, H_p$. Thus a criterion $\zeta_{j\ell}$ denotes that the criterion ℓ is considered at time step $\tau + j$, with $\ell = 1, \ldots, m$ and $j = 1, \ldots, H_p$. Further, let $\mu_{\zeta_{j\ell}}$ denote the membership value that represents the satisfaction of the decision criterion after applying the control actions $\mathbf{u}(\tau + j)$. The total number of decision criteria in the problem is thus given by $M = m \times H_p$. The confluence of goals and constraints can be done by aggregating the membership values $\mu_{\zeta_{j\ell}}$. The membership value μ_π for the control sequence π is obtained by using the aggregation operators $\circledast$, $\circledast_g$ and $\circledast_c$ to combine the decision criteria,

$$\begin{aligned}
\mu_\pi = {} & (\mu_{\zeta_{11}} \circledast_g \cdots \circledast_g \mu_{\zeta_{1n_g}}) \circledast (\mu_{\zeta_{1(n_g+1)}} \circledast_c \cdots \circledast_c \mu_{\zeta_{1m}}) \circledast \\
& (\mu_{\zeta_{21}} \circledast_g \cdots \circledast_g \mu_{\zeta_{2n_g}}) \circledast (\mu_{\zeta_{2(n_g+1)}} \circledast_c \cdots \circledast_c \mu_{\zeta_{2m}}) \circledast \\
& \cdots \\
& (\mu_{\zeta_{H_p1}} \circledast_g \cdots \circledast_g \mu_{\zeta_{H_pn_g}}) \circledast (\mu_{\zeta_{H_p(n_g+1)}} \circledast_c \cdots \circledast_c \mu_{\zeta_{H_pm}})\,.
\end{aligned} \tag{9.2}$$

In Eq. (9.2), $\circledast_g$ denotes an aggregation operator for combining the goals, $\circledast_c$ denotes an aggregation operator for combining the constraints, and $\circledast$ denotes an aggregation operator to combine the aggregated goals and constraints. In general, it is not necessary to use the same aggregation operator for all goals and for all constraints. However, using a single aggregation operator reduces complexity, making the confluence of criteria simpler.

Note that the aggregation operator to combine a goal and a constraint between time steps, *i.e.*, the last $\circledast$ in each row, is the same as the aggregation operator to combine goals and constraints within a time step, *i.e.*, the $\circledast$ operator in the middle of the rows. For the purpose of this book, the aggregation as in Eq. (9.2) is sufficiently general. However, some systems can demand different aggregation operators for combining goals and constraints at different time steps, or even between goals or between constraints at the same time step. For these cases, general aggregation operators $\circledast_l$, $l = 1, \ldots, M - 1$ can be used for each aggregation. This possibility is, however, not practical, because the degree of complexity becomes too high, and the effects of each different aggregation operator would be hardly predictable. Various types of aggregation operations can be used as decision functions for expressing different decision strategies by using the well-known properties of these operators (see Chapter 3). In fact, aggregation operators and membership functions translate a linguistic description of the control goals into a decision function. In this way, various forms of aggregation can be chosen, giving greater flexibility for expressing the control goals. A discussion on the influence of aggregation operators in FDM applied to control is given in Sec. 9.3.1.

In the following, the combination of criteria in different domains is done for a set of discrete alternatives, which corresponds to different policies π that can be applied to find the optimal control policy. The decision criteria in Eq. (9.2) should be satisfied as much as possible, which corresponds to the maximal value of the overall decision. Thus, the optimal sequence of control actions π^* is found by the maximization of μ_π

$$\pi^* = \arg \max_{\mathbf{u}(\tau),\ldots,\mathbf{u}(\tau+H_{p-1})} \mu_\pi . \tag{9.3}$$

Because the membership functions for the fuzzy criteria can have an arbitrary shape, and because of the nonlinearity of the decision function, the optimization problem of Eq. (9.3) is usually non-convex. To deal with the increasing complexity of the optimization problem, different methods can be utilized. One possibility is to consider only a few criteria in Eq. (9.2), removing the ones not considered from the equation. This approach, however, can result in sub-optimal control actions. A better method is to choose a proper optimization algorithm, or to formulate the problem in a way that leads to convex optimization. One set of conditions that leads to a convex optimization problem is discussed in Sec. 11.1. Elsewhere, several methods to deal with non-convex optimization problems have been used, such as sequential quadratic programming (Gill et al. 1981), the simplex method (Nelder and Mead 1965), genetic algorithms (Onnen et al. 1997) or branch-and-bound (Sousa et al. 1997, Sousa 2000). Problems concerning optimization are discussed in Chapter 10 and Chapter 11.

9.2.3 *Fuzzy criteria in model-based predictive control*

The definition of fuzzy goals and constraints must be given by a design engineer. Therefore, when FDM in control is considered, human knowledge is involved in specifying the control objectives and constraints, rather than the control protocol itself (Goodrich et al. 1999). Using a process model, a fuzzy decision making algorithm selects the control actions that best meet the specifications (see Fig. 9.2). Hence, a control strategy can be obtained that is able to push the process closer to the constraints, and that is able to force the process to a better performance based on the goals and the constraints set by the system operator together with the known conditions provided by the system's designers.

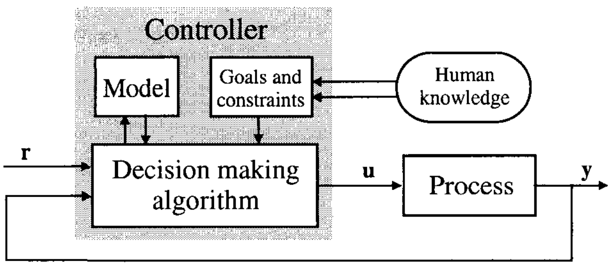


Fig. 9.2 Controller based on objective evaluation and fuzzy decision making. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

This approach is closely related to model-based predictive control, presented in Appendix A. The formulation of the control problem as a confluence of fuzzy goals and fuzzy constraints leads to a generalization of the objective function used in MBPC (Sousa and Kaymak 2001).

For practical reasons, it is desirable to have direct control over the influence of the individual components of the objective function on the controller performance. Thus, it is advantageous that the degree of compensation among the different goals and constraints can be specified by the designer. This additional freedom can be achieved by choosing a different representation of the objective function, given by the combination of fuzzy goals and fuzzy constraints, as in the FDM approach. In the MBPC environment, a policy π with the possible control actions $\mathbf{u}(\tau), \ldots, \mathbf{u}(\tau + H_p - 1)$ can be defined as in Eq. (9.1). The objective function using fuzzy criteria is defined in Eq. (9.2). The closed-loop control configuration is now discussed in more detail, in aspects concerning the criteria and the aggregation operator(s) that are used to combine them.

9.3 Fuzzy criteria for decision making in control

Fuzzy criteria play a main role in fuzzy decision making. When FDM is applied to control, the fuzzy goals and the fuzzy constraints must be a translation of the (fuzzy) performance criteria defined for the system. The definition of performance criteria in the time domain has shown to be quite powerful, especially for nonlinear systems (Slotine and Li 1991) and in the model predictive control framework (Camacho and Bordons 1995). This section investigates the use of fuzzy performance criteria in predictive control and compares the results to those obtained from conventional model predictive control. First, the aspects concerning the aggregation operator(s) combining the criteria are presented in Sec. 9.3.1. Next, control criteria and decision functions are discussed in Sec. 9.3.2, where classical objective functions and a proposed fuzzy objective function are presented. The proposed approach is demonstrated on two simulated systems:

- a non-minimum phase, unstable linear plant, and
- an air-conditioning system with nonlinear dynamics,

presented in Sec. 9.4.

9.3.1 *Aggregation operators for FDM in control*

This section presents a discussion on the possible use of different aggregation operators in fuzzy decision making, and the advantages and the disadvantages of their use in predictive control.

In general, the choice of the operator is application dependent. The first operator proposed to aggregate goals and constraints is the minimum operator (Bellman and Zadeh 1970). In this approach, the operators $\circledast$, $\circledast_g$ and $\circledast_c$ are all substituted by the minimum operator in Eq. (9.2), leading to

$$\mu_\pi = \min\left(\mu_{\zeta_{11}}, \mu_{\zeta_{12}}, \ldots, \mu_{\zeta_{H_pm}}\right) . \tag{9.4}$$

Although this operator is still largely used in FDM, it does not allow for any tradeoff or compensation between the criteria (Fung and Fu 1977), because it always chooses the smallest of the aggregated M values as the decision. For this reason, this operator is usually known as a safety-first or *pessimistic* operator. This disadvantage can be overcome by the use of another t-norm, which should still translate the aggregation as a simultaneous satisfaction of the fuzzy criteria, but allow for some interaction amongst the criteria. The most used aggregation operator after the minimum operator is possibly the product t-norm,

$$\mu_\pi = \mu_{\zeta_{11}} \cdot \mu_{\zeta_{12}} \cdot \ldots \cdot \mu_{\zeta_{H_pm}} . \tag{9.5}$$

This operator allows some interaction between the criteria, but keeps the characteristics of t-norms (*i.e.*, any low degree of membership for one criteria $\zeta_{j\ell}$ implies that the degree of membership μ_π is also low). When the number of criteria increases, μ_π tends to decrease. This fact is quite realistic because the larger the number of goals and constraints, the more difficult it is to satisfy them all. A similar conclusion can also be drawn for many other t-norms, because by definition they are always smaller than or equal to the min operator. The presented aggregation operators assume that the importance of different criteria is equal.

The attribution of different weights for different criteria can be made by using the weighted-sum in a similar way as it is usually done for classical criteria in predictive control, as will be presented in Eq. (9.9). Another possibility is to use the approach presented by Yager (1992), where each criterion has a different weight $w_{j\ell} \in [0,1]$, reflecting a different importance in the global criterion Eq. (9.2). Other weighted aggregation methods discussed in Chapter 3 can also be used.

A different approach can be followed by using *parametric* t-norms, which can generalize a large number of t-norms, and control the degree of compensation between the different goals and constraints. Usually, parametric t-norms depend only on one parameter, which makes them much easier to tune compared to the tuning of weight factors in weighted t-norms. However, they are not as general as the weighted approaches. For the examples presented in this section, parametric t-norms revealed good control performances. Several parametric t-norms can be considered, such as the ones introduced by Hamacher (1978) (see Eq. (3.7)) , Yager (1980) (see Eq. (3.6)) and Weber (1983). In the notation of this chapter, for instance, the Yager t-norm is given by

$$\mu_\pi = \max\left(0,\ 1 - \left\{\sum_{j=1}^{H_p}\sum_{\ell=1}^{m}(1-\mu_{\zeta_{j\ell}})^\gamma\right\}^{1/\gamma}\right), \quad \gamma > 0. \tag{9.6}$$

Equation (9.6) corresponds to a multioperand formulation of the Yager t-norm. This operator covers the entire range of t-norms, *i.e.*, it goes from the drastic intersection to the minimum operator.

The fuzzy decisions discussed so far are made under the assumption that 'G must be accomplished *and* C must be satisfied.' However, sometimes it is more appropriate to formulate the FDM problem as follows:

$$G \text{ must be accomplished } \textit{or } C \text{ must be satisfied.} \tag{9.7}$$

This type of situation is found when an agreement between different opinions must be achieved. In these cases, the fuzzy decision is given by the union of the fuzzy sets, generally described by an s-norm (t-conorm). When the maximum operator is used, no interaction between criteria is allowed, similarly to the minimum operator for the aggregation of criteria. In fact, an alternative is selected based only on the best criteria, regardless that all the others are poorly fulfilled. This feature is sometimes referred to as *full compensation*, as discussed in Chapter 3. In general, the t-conorms are not suitable for most control purposes, as they consider only the best satisfied criteria. Hence, they are not used further in this book for control purposes.

There is a large range of fuzzy operators between the t-norms and s-norms that can sometimes be suitable for the confluence of fuzzy criteria for control applications. Examples are the compensatory aggregation operator introduced by Zimmermann and Zysno (1980) (see Eq. (3.41)), or the generalized mean (Kaymak and van Nauta Lemke 1993). In the notation of this chapter, this last operator is given by

$$\mu_\pi = \left\{ \frac{1}{M} \sum_{j=1}^{H_p} \sum_{\ell=1}^{m} \mu_{\zeta_{j\ell}}^{\gamma} \right\}^{1/\gamma}, \qquad \gamma \in \mathbb{R}. \tag{9.8}$$

As discussed before, Eq. (9.8) reduces to the harmonic, geometric, arithmetic and quadratic mean when the parameter is $\gamma = -1$, $\gamma \to 0$, $\gamma = 1$ and $\gamma = 2$, respectively. Moreover, when $\gamma \to -\infty$ the generalized mean approaches the minimum operator, and when $\gamma \to +\infty$ it approaches the maximum operator.

When a large number of criteria is present and some tradeoff between the different criteria is allowed, Eq. (9.8) can have some advantages over aggregation operators described by t-norms. It should be emphasized, however, that the use of this operator may lead to the violation of 'hard' constraints, when they are defined as in Sec. 9.2.1. Therefore, when this operator is used, the optimal alternative found should be checked afterwards in order to assure that no hard constraints are violated. However, this procedure can cost precious optimization time. A solution is to use the generalized mean only for the general confluence operator $\circledast_g$ and a t-norm for the remaining $\circledast$ and $\circledast_c$. This choice ensures that the 'hard' constraints are not violated, but it can hamper the advantages of using the generalized mean.

Some general rules to choose the appropriate aggregation operators are presented in Yager and Filev (1994), Dubois and Prade (1980), and in Zimmermann (1996). Following Zimmermann (1996), the general guidelines deal with axiomatic strength, adaptability, numerical efficiency, degree of compensation, aggregating behavior and required scale level of membership functions. This book uses parameterized operators, and their choice is strongly recommended because they allow for different degrees of compensation between criteria. Moreover, the change of a single parameter results in the use of different operators, simplifying the tuning phase, always present in predictive control. Section 9.4.2 presents a simple, though illustrative example, showing the application of three different aggregation operators.

9.3.2 *Control criteria and decision functions*

When a control system is designed, performance criteria must be specified. In the time domain, these criteria are usually defined in terms of a desired steady-state error between the reference and the output, rise time, overshoot, settling time, and so on (see Sec. 8.3), representing the goals of the control system. In MBPC, these goals must be translated into an objective function. This function is maximized (or minimized) over the prediction horizon, given the desired control actions. The translation of the (fuzzy) goals into an objective function can be done in two different ways.

- The control goals are explicitly expressed in the objective function. This method usually leads to long term predictions of the behavior of the system, using a large prediction horizon H_p. From these predictions, quantities such as the overshoot or the rise time can be determined. In order to have accurate predictions, this method requires a highly accurate process model, which may not be available, and a lot of computation.
- Only short-term predictions (a few steps ahead) are used in the objective function. This method is usually applied in predictive control when the available model of the system is not very accurate, and cannot predict outputs for a large number of steps ahead. Despite this inaccuracy of the model, it still can lead to high performance control, provided that the overall control goals can be translated into the short-term goals, which are then represented in the objective function. This translation is, however, not unique, and it is application dependent. Therefore, tuning of some parameters in the objective function is usually required. This method is especially suitable for nonlinear systems, where a compromise between computational time to derive the control actions and accuracy of the predictions must be made, except for special cases, as when input–output feedback linearization is utilized (Henson and Seborg 1990). When using fuzzy criteria, the task of defining the goals becomes easier, as will be shown in this section.

9.3.2.1 *Classical objective functions*

Conventional MBPC mainly utilizes sum-quadratic functions, given by Eq. (A.2), as the objective function (Soeterboek 1992, Clarke and Mohtadi 1989). The main motivation for its use is that such an objective function has an analytical solution for linear systems without constraints. In the presence of crisp and convex constraints, the optimization problem remains convex for linear systems, and can still be solved in polynomial time. However, the presence of non-convex constraints and/or the presence of non-linearities in the system often leads to non-convex optimization problems. In these cases, the sum-quadratic objective function does not have any advantages over other more complex objective functions that can possibly better describe the (fuzzy) performance criteria for a broad class of control problems.

Let the overall control goals for the time domain be stated as achieving a fast system response while reducing the overshoot and the control effort. These goals are represented in the objective function Eq. (A.2). An extension of this objective function is used in this section by including the changes in the outputs, resulting in the following objective function for SISO systems,

$$J_c = \sum_{j=n_{1l}}^{n_{1u}} w_{1j}(\hat{\mathbf{e}}(\tau+j))^2 + \sum_{j=n_{2l}}^{n_{2u}} w_{2j}(\Delta\mathbf{u}(\tau+j-1))^2 + \sum_{j=n_{3l}}^{n_{3u}} w_{3j}(\Delta\hat{\mathbf{y}}(\tau+j))^2, \tag{9.9}$$

where $\hat{e}(\tau + j)$ denotes the predicted errors given by the difference between the reference r and the output of the system y, *i.e.*,

$$\hat{e}(\tau+j) = r(\tau+j) - \hat{y}(\tau+j). \tag{9.10}$$

The change of the predicted output $\Delta\hat{y}$ is defined as

$$\Delta\hat{y}(\tau+j) = \hat{y}(\tau+j) - \hat{y}(\tau+j-1), \tag{9.11}$$

and is equal to the change in the errors $\Delta\hat{e}(\tau + j)$, when the reference to be followed is constant. The change in the control actions is defined in a similar way as

$$\Delta u(\tau+j) = u(\tau+j) - u(\tau+j-1). \tag{9.12}$$

The parameters w_{1j}, w_{2j} and w_{3j} are weighting terms that are application dependent. The parameters n_{1l}, n_{1u}, n_{2l}, n_{2u}, n_{3l} and n_{3u} must be selected appropriately depending on the application, and they must satisfy $1 \leq n_{il} \leq n_{iu} \leq H_p$, $i \in \{1, 2, 3\}$. Usually, n_{1l}, n_{2l}, and n_{3l} are chosen equal to 1, n_{1u} and n_{3u} equal to H_p, and n_{2u} equal to H_c. Note that the weighting terms w_{1j}, w_{2j} and w_{3j} must account for the difference of magnitude between the different inputs and/or outputs of the system at various time instants. If this is not the case, and the weights are chosen all equal, for instance, the optimization automatically weighs different variables, which is not desirable, and it leads to poor control performance.

The objective function of Eq. (9.9) can be interpreted as follows. The term containing the predicted errors indicates that these should be minimized, while the term containing the change in the control actions indicates that the control effort should be reduced. Finally, the term containing the change in the outputs indicates that the system's output should not suffer sudden changes, and thus it helps to improve the smoothness of the response. For step references, the change of the output is also equal to the change in the respective output errors, except for the discontinuities in the reference signal. Hence, minimizing the output errors and the change of output errors can be regarded as forcing the system to the origin (steady-state solution) in the $e \times \Delta e$ phase space. The parameters containing the weights, w_{1j}, w_{2j} and w_{3j} can be changed so that the objective function is modified in order to lead to a desired system response. Notice that these parameters have two functions: they normalize the different outputs and inputs of the system, and they vary the importance of the three different terms in the objective function of Eq. (9.9) over the time steps.

9.3.2.2 *Fuzzy objective functions*

When fuzzy multicriteria decision making is applied to determine the objective function, additional flexibility is introduced. Each criterion $\zeta_{j\ell}$ is described by a fuzzy set, where $j = 1, \ldots, H_p$, stands for the time step $\tau + j$, and $\ell = 1, \ldots, m$ are the different criteria defined for the considered variables at the same time step. Fuzzy criteria can be described in different ways. The most straightforward and easy way is just to adapt the criteria defined for classical objective functions. A SISO system with a control action $u(\tau)$ and an output $y(\tau)$ is considered. Figure 9.3 shows examples of general membership functions that can be used for the error $\hat{e}(\tau + j) = r(\tau + j) - \hat{y}(\tau + j)$, for the change in the predicted output $\Delta\hat{y}(\tau + j)$, and for the change in the control action $\Delta u(\tau + j - 1)$, with $j = 1, \ldots, H_p$.

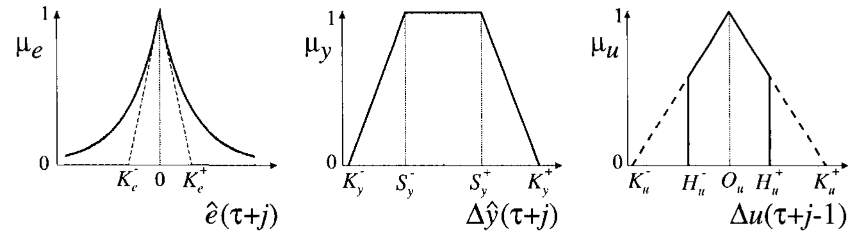


Fig. 9.3 Membership functions that represent the satisfaction of decision criteria for the error, change in output and change in the control action. Reproduced from (Sousa and Kaymak 2001),©2001 IEEE.

In this example, the minimization of the output error $\mu_e(\hat{e}(\tau + j))$ is represented by an exponential membership function, given by

$$\mu_e = \begin{cases} \exp\left(-\frac{|\hat{e}(\tau+j)|}{K_e^-}\right), & -\infty < \hat{e}(\tau+j) < 0\,; \\ \exp\left(-\frac{|\hat{e}(\tau+j)|}{K_e^+}\right), & 0 \leq \hat{e}(\tau+j) < \infty\,. \end{cases} \tag{9.13}$$

This well-known function has the nice property of being tangent to the triangular membership function defined using the parameters K_e^- and K_e^+, see Fig. 9.3. Another interesting feature of this exponential membership function is that it never reaches the value zero, and the membership value is still quite considerable, 0.37, for an error of K_e^- or K_e^+ magnitude. Therefore, this criterion is considered to be a fuzzy goal, as explained in Sec. 9.2.1. This definition of membership function allows for the comparison of the error parameters, K_e^- and K_e^+, to the parameters defined for other fuzzy criteria such as the change in output and the change in control actions.

The change in the output can be represented, for example, by a trapezoidal membership function $\mu_y(\Delta\hat{y}(\tau + j))$, as shown in Fig. 9.3. The system can vary with no limitations in the interval $[S_y^-, S_y^+]$. Outside this interval, physical limitations can be defined such that the change in the output cannot go below K_y^- or above K_y^+. This fuzzy constraint can be seen as a fuzzy goal if no physical limitations are present in the system, and it is not compulsory that the membership value is zero outside a given interval. Note that if this is the case, K_y^- and K_y^+ can play the same role as K_e^- and K_e^+ in the membership function defined for the error in Eq. (9.13). Thus, outside the interval $[S_y^-, S_y^+]$ exponential membership functions such as the one defined for the error $\hat{e}(\tau + j)$ can also be used.

The control effort $\mu_u(\Delta u(\tau + j - 1))$ is, in this case, represented by a triangular membership function around zero, which is considered a fuzzy constraint. The crisp rate constraints on Δu representing the maximum and the minimum allowed in the system are given by H_u^- and H_u^+, respectively. These constraints are related to physical limitations of the system. The membership degree should be zero outside the interval $[H_u^-, H_u^+]$. The parameters defining the range of the triangular membership function are K_u^- and K_u^+. Note that the membership function $\mu_u(\Delta u(\tau + j - 1))$ does not have to be symmetrical. Sometimes it is convenient to make $K_u^- = H_u^-$ and $K_u^+ = H_u^+$, but other systems may require bigger membership values for the points in the interval $[H_u^-, H_u^+]$, as in Fig. 9.3. Further, μ_u can be defined as a trapezoidal membership function in a similar way to the one defined for the change in output.

In principle, different criteria can be defined at each time instant $\tau + j$, $j = 1, \ldots, H_p$. This example has $m = 3$ decision criteria (for e, Δu and Δy), and the total number of criteria in a fuzzy MBPC problem is thus given by $3 \cdot H_p$.

Beyond the possibility of defining different criteria for different time steps, it is possible to skip some criteria at certain steps. An example of different criteria at different time steps can be the spread of the membership function defined for the error, which can be narrowed as the time approaches H_p, *i.e.*, it is more important to achieve the goal of small error close to the prediction horizon. This corresponds to a decreasing value of K_e in Fig. 9.3. Sometimes it is also advantageous to consider some criteria just at a particular time step. One example is the variation of the control action, which can be quite small for steady-states, but it should change quite significantly for different situations, as, *e.g.*, when a step response must be followed. The designer should thus carefully choose the criteria at each time step, regarding the desired performance criteria of the system under control. In general, all the parameters of the different membership functions are application dependent. However, it is possible to derive some tuning guidelines, as will be described in Sec. 9.4.

The membership values $\mu_{\zeta_{j\ell}}$ quantify how much the system satisfies the criteria given a particular control sequence, bringing various quantities into an unified domain. The use of the membership functions introduces additional flexibility for expressing the control goals, and it leads to increased transparency as it becomes possible to specify explicitly what type of system response is desired. For instance, it becomes easier to penalize errors that are larger than a specified threshold more severely. Note that there is no need to scale several parameters, such as w_{1j}, w_{2j} and w_{3j} in Eq. (9.9), when fuzzy objective functions are used, because the use of membership functions introduce directly the normalization required. This feature reduces the effort of designing a model-based predictive controller with fuzzy objective functions compared to the use of classical objective functions.

After the membership functions have been defined, they are combined by using a decision function, such as a parametric aggregation operator from the fuzzy sets theory (see Sec. 9.3.1). Usually, the additional parameter of the decision function influences the optimization results in a way that cannot be expressed by weight factors. In this way, the objective function can be tuned with a single parameter.

9.4 Application examples

This section presents two simulation examples showing the influence of the conventional and the fuzzy objective functions in predictive control. After the description of the systems, the choice of aggregation operators is discussed for one of the systems. Next, classical and fuzzy objective functions are applied to the systems, and a discussion about the results obtained is given.

9.4.1 *Description of the simulated systems*

The influence of conventional and fuzzy objective functions in predictive control has been studied by using two different systems.

(1) A simulated non-minimum phase, open-loop unstable linear system.
(2) A simplified nonlinear model of an air-conditioning system, which is derived from real data of a test cell by using fuzzy modeling techniques.

The following sections describe these systems in more detail. In order to concentrate on differences between the two control schemes, model–plant mismatch and the implementational aspects are not considered in this chapter. However, these aspects are considered for the real-time implementation of the air conditioning system presented in Chapter 12.

9.4.1.1 *Linear system*

A linear system has been selected for the first set of experiments in order to be able to compare the control results when classical and fuzzy criteria are applied. The selected system is described by the transfer function

$$G(s) = \frac{s-1}{s^3+s^2+s+2}. \tag{9.14}$$

This is a non-minimum phase system and it has two complex poles in the right-half plane (unstable in open-loop). The poles and zero placement are given in Fig. 9.4. The system, preceded by a zero-order-hold circuit, has been discretized with a sample time of 1 s.

9.4.1.2 *Air conditioning system*

A Heating, Ventilating and Air Conditioning (HVAC) system consists of a number of heat exchangers, pipes or dampers, which supply hot water, steam or chilled water to a heating or cooling unit responsible for the conditioning of a space. Figure 9.5a shows the HVAC system that is used in this study. Hot water at 65 °C is supplied to a coil which exchanges the heat between the hot water and the surrounding air. A valve controls the amount of hot water that flows through the coil. A fan is responsible for the ventilation and it supplies the hot air coming from the coil to the test–cell (room). The global control goal is to keep the room at a reference temperature while assuring sufficient ventilation. The fan can be set to three different velocities: low, medium, and high. A return damper controls the amount

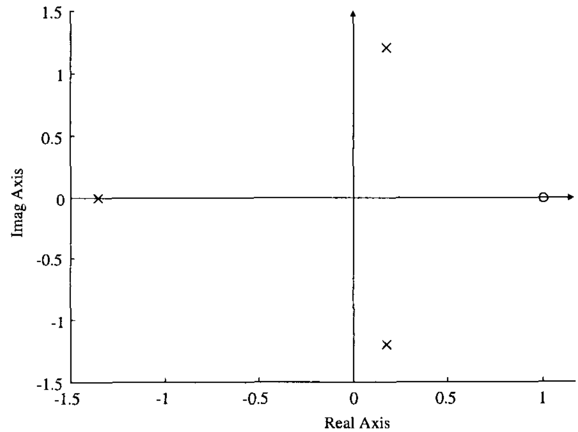
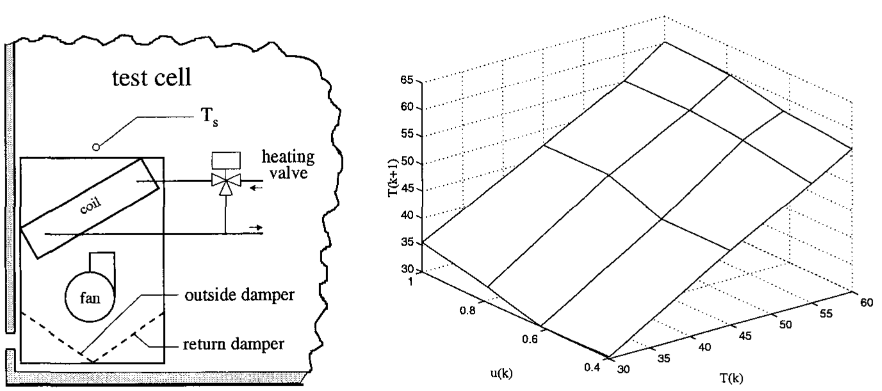


Fig. 9.4 Position of the poles and zeros of the linear system given by Eq. (9.14). Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

of recycled air from the room, while an outside damper controls the amount of fresh air coming from outdoors. The supply temperature T_s, which is measured after the coil, is controlled with the heating valve. A SISO model of the system is determined from input–output measurements made with a sampling period of 30 s. The temperature can be described as a nonlinear, first-order dynamic system $T_s(\tau + 1) = f(T_s(\tau), u(\tau))$, where $u(\tau) \in [0.4, 1]$ is the valve opening and $T_s(\tau) \in [30, 60]$ is the temperature in °C at time instant τ. A Mamdani fuzzy model with singleton consequents is obtained using the identification method described in Chapter 5. Figure 9.6 shows the triangular membership functions that are determined for the temperature $T_s(\tau)$ and the valve opening $u(\tau)$. Note that the universe of discourse for the valve opening is the interval $[0.4, 1]$ because the valve shows a dead-zone behavior between 0 and 0.4, when the system is considered to be a SISO system. The fuzzy rule base that describes the model is given in Table 9.1. An example of a rule for this singleton model, as, *e.g.*, the first rule, is '**If** $u(\tau)$ is *Small* **and** $T_s(\tau)$ is *Low* **then** $\hat{T}_s(\tau + 1) = 30.3$'. Figure 9.5b depicts the piece-wise linear mapping that is described by the fuzzy model. The fuzzy model is used to simulate the system and develop predictive controllers.

9.4.2 *Application of aggregation operators to the linear system*

In this section, several issues, such as interaction amongst criteria, the influence of the types of decision functions and their parameters, are studied using the simulated linear system given in Eq. (9.14). The membership functions for the fuzzy goals and the constraints are assumed to be given. Remember that this system



(a) Air conditioning system. *(b) Piecewise linear model of the air conditioning system.*

Fig. 9.5 Air conditioning system: schematic representation and derived model. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

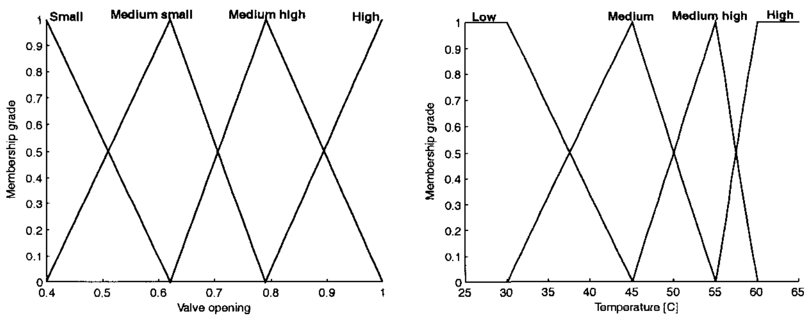
Table 9.1 Rule base for the fuzzy model of the HVAC process. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

	Temperature			
Valve Opening	*Low*	*Medium*	*Medium high*	*High*
Small	30.3	43.9	52.6	56.4
Medium small	30.0	43.8	54.2	57.6
Medium high	32.8	47.4	55.6	59.7
High	35.5	47.3	55.1	60.3

is non-minimum phase and has two complex poles in the right-half plane (unstable in open loop). Only the minimization of the predicted output error is used as an optimization criterion in order to keep the optimization problem transparent and to clearly understand the influence of the decision function on the solution. The optimization criterion is represented by a symmetric exponential membership function which is defined around zero output error as

$$\mu_e = \exp\left(-\frac{|\hat{e}(\tau+j)|}{30}\right),$$

with $\hat{e}(\tau+j) = r(\tau+j) - \hat{y}(\tau+j)$, where r is the reference and $\hat{y}$ is the predicted model output. This function is a particular case of Eq. (9.13) for $K_e^+ = -K_e^- = 30$. A crisp constraint $|\Delta u| = 0.5$ is imposed on the rate of the control action, and



(a) *Membership functions for $u(\tau)$.*

(b) *Membership functions for $\mathrm{T}_s(\tau)$.*

Fig. 9.6 Membership functions for the antecedent variables of the fuzzy HVAC model. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

it is represented by a membership function that is defined on Δu.

Step responses of the system have been studied. The controller is implemented in the incremental form and the optimization is performed in the discretized Δu space. This control space is divided into 11 discrete levels and an enumerative search scheme has been used to determine the best control action. The control horizon H_c is chosen as small as possible to keep the search space small. A value of 2 is found to be satisfactory. Similarly, the prediction horizon H_p is kept relatively small to a value of 6. The response of the controller is studied for three different aggregation operators, namely the minimum operator in Eq. (9.15), the generalized mean in Eq. (9.16), and the Yager t-norm in Eq. (9.17).

$$\mu_\pi = \min_{j=1,\ldots,H_p} (\mu_j(\hat{e}(\tau + j))) \tag{9.15}$$

$$\mu_\pi = \left\{ \frac{1}{H_p} \sum_{j=1}^{H_p} \mu_j(\hat{e}(\tau + j))^\gamma \right\}^{1/\gamma} \quad , \gamma \in \mathbb{R} \tag{9.16}$$

$$\mu_\pi = \max\left(0, 1 - \left\{ \sum_{j=1}^{H_p} (\bar{\mu}_j(\hat{e}(\tau + j))^\gamma \right\}^{1/\gamma} \right), \gamma > 0 \tag{9.17}$$

where $\bar{\mu}_j(\hat{e}(\tau + j)) = 1 - \mu_j(\hat{e}(\tau + j))$, *i.e.*, Zadeh's fuzzy complement. The responses with the parametric decision functions have been calculated for several values of the parameters.

Minimum operator. It is known that the minimum operator does not allow interaction amongst criteria. It optimizes the worst action in the control sequence and makes sure that it is as good as possible. However, because the system has non-minimum phase behavior, the minimum operator cannot be used for optimization because every control action except for zero will result in an (initial) increase of the error, decreasing the value of μ_π. Hence, the 'best' control action will be zero, and the controller will not select another control action. For this reason, a decision function that allows for interaction amongst criteria is required for this type of system.

Generalized mean. For the generalized averaging operator as in Eq. (9.16), when γ is chosen very large, the system tries to reach the reference value as soon as possible and shows an overshoot. The system slows down for small values of γ. A fast response without an overshoot is obtained for γ equal to 1 (arithmetic mean). Figure 9.7 shows the response of the controlled system for several values of the parameter γ.

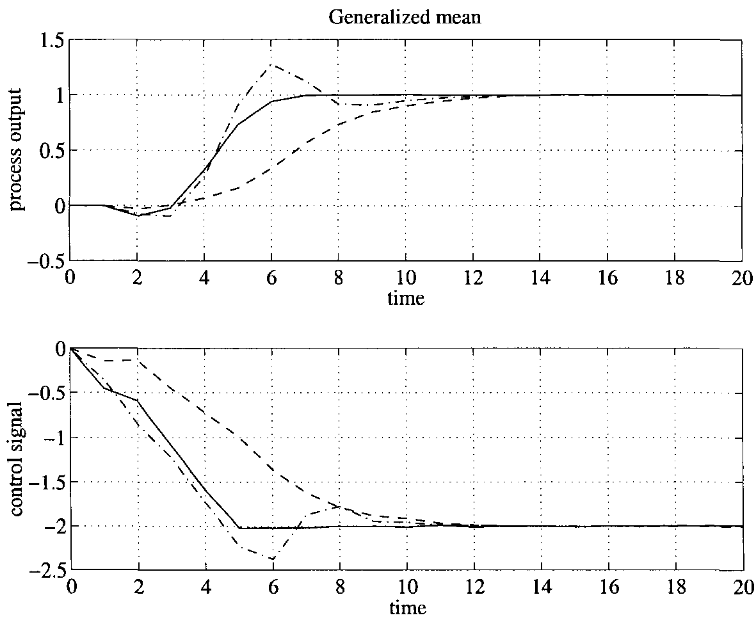


Fig. 9.7 Response of a fuzzy predictive controller using the generalized mean as the decision function. Dashed: $\gamma = -1$, solid: $\gamma = 1$, dash–dotted: $\gamma = 3$. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

Yager t-norm. Unlike the generalized mean, the system shows fast response for small values of the parameter γ, when the Yager t-norm is used. Being a t-norm, this operator tries to achieve a simultaneous satisfaction of all the criteria. The parameter γ should not be chosen very small as the simultaneous satisfaction

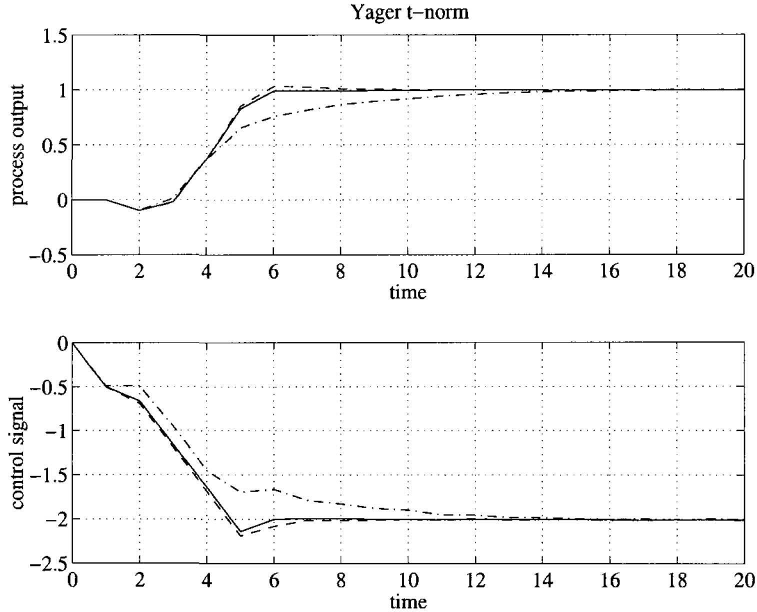


Fig. 9.8 Response of a fuzzy predictive controller using the Yager t-norm as the decision function. Dashed: $\gamma = 2$, solid: $\gamma = 2.8$, dash–dotted: $\gamma = 4$. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

of the criteria may then be unfeasible. When γ is around 2.8, the controlled system shows a very fast response without overshoot. The step response is even faster than the response that is obtained with the arithmetic mean as the decision operator. Figure 9.8 shows the step response for several values of γ.

The parameter γ can be interpreted as a speed indicator for the response. For the generalized mean operator, small values of γ favor small control actions and the system response slows down. Large values of γ favor a faster decrease of the error and thus larger control actions are favored. Thus, the system response can be tuned by using the parameter of the decision functions as an extra degree of freedom. Additional objectives such as the rising time and the overshoot could also be controlled with this single parameter.

9.4.3 *Fuzzy vs. conventional objective functions*

In this section, model-based predictive controllers are designed both for the linear system and for the HVAC system by using fuzzy objective functions as well as conventional objective functions. The performance of the controllers based on the two types of objective functions is compared.

Model predictive controllers have been designed for the systems described in Sec. 9.4.1 by using both the conventional objective function of Eq. (9.9) and a fuzzy objective function. A t-norm is used for the aggregation, since the decision goal is formulated as the simultaneous satisfaction of all the decision criteria. The aggregation operators $\circledast$, $\circledast_c$ and $\circledast_g$ in Eq. (9.2) are taken as the Yager t-norm, which combines the control criteria presented in Fig. 9.3. The overall aggregation is then given by

$$\begin{aligned}\mu_{\pi'} &= \sum_{j=n_{1l}}^{n_{1u}} (\bar{\mu}_e(\hat{e}(\tau+j)))^{\gamma} + \sum_{j=n_{2l}}^{n_{2u}} (\bar{\mu}_y(\Delta\hat{y}(\tau+j)))^{\gamma} \\ &\quad + \sum_{j=n_{3l}}^{n_{3u}} (\bar{\mu}_u(\Delta u(\tau+j-1)))^{\gamma} \\ \mu_{\pi} &= \max(0, 1-\mu_{\pi'}^{1/\gamma}), \qquad \gamma > 0, \end{aligned} \tag{9.18}$$

where the parameters n_{il} and n_{iu}, $i \in \{1,2,3\}$ are defined as in Eq. (9.9), and Zadeh's complement is defined as in Eq. (9.17). The parameter γ allows for the choice of different t-norms (see Sec. 9.3.1).

The response of the controllers is studied using simulations of the systems. Given Eq. (9.18) as an aggregation operator, the membership functions and the parameters of the objective functions have been chosen in such a way that they lead to fast response while avoiding excessive oscillations and overshoot within the working range of the controller. The prediction horizon is kept as small as possible, since in practice the model–plant mismatch hampers the use of long horizons.

In this study, the control space is discretized and the optimal control sequence is determined by an enumerative search. The control horizon is chosen equal to two in order to keep the computational load low. To further reduce the computational load, a two-step optimization approach is used, where a rough solution is found by using a coarse discretization of the control space, followed by the calculation of a finer solution around the rough solution. Other optimization techniques for non-convex problems, such as the branch-and-bound or genetic algorithms can also be used as discussed in Chapter 10.

9.4.3.1 *Linear system*

The predictive control scheme is applied to the linear system given by Eq. (9.14) without any constraints on the system. In this case, both the conventional criteria and the fuzzy criteria are able to control the system with a fast step response and no overshoot. However, when a rate constraint of $|\Delta u| \leq 0.5$ is imposed on the system, the influence of the fuzzy criteria on the control problem becomes more dominant. For these experiments, $H_c = 2$ and $H_p = 6$. It is required that the controller can bring the system to any level in the interval $[-3, 3]$. Using the output error and the change in the output with $n_{1l} = n_{3l} = 1$ and $n_{1u} = n_{3u} = H_p$ was found to be sufficient for controlling the system. The following parameters are used for the conventional objective function: $w_{1j} = 1$, $w_{2j} = 0$ and $w_{3j} = 5$, $j = 1, \ldots, H_p$. These values are chosen following the general guidelines presented in (Soeterboek 1992, Camacho and Bordons 1995), and by trial and error until a good response of the system is found. The parameter w_{3j} is a compromise between fast response (for smaller values) and small or no overshoot (for bigger values). For the fuzzy criteria, the following membership function parameters are found by tuning $K_e^+ = -K_e^- = 1$, $K_y^+ = -K_y^- = 1$, $S_y^+ = -S_y^- = 0.5$, and $\gamma = 2$ for the Yager t-norm. The way to tune the Yager parameter has been discussed in Sec. 9.3.1. The membership functions for the error and change in error are chosen to have equal magnitude by choosing $K_e^+ = K_y^+ = 1$, and by taking K_e^- and K_y^- symmetrical to K_e^+ and K_y^+, respectively. Note that the choice of these four parameters requires only the tuning of one of them, because they are all related. Finally, the parameters S_y^+ and S_y^- are chosen such that the system can move freely to a certain degree, and is penalized outside these limits. The criterion on the change of the control action is not considered because it does not introduce any improvement in the control performance of this system. The responses of the system for several steps using classical and fuzzy criteria are shown in Fig. 9.9 and Fig. 9.10, respectively. It is clear that the predictive controller with fuzzy criteria can improve the speed of the response considerably, while avoiding overshoots. The response of the controller with conventional criteria can be made faster by changing the values of w_{3j}, but this occurs at the expense of amplifying the oscillations due to the non-minimum phase behavior. Another solution can be found by extending the prediction horizon. However, a considerable increase of the prediction horizon is required, and this is in general undesirable. Hence, this system clearly benefits from the additional flexibility introduced by the fuzzy criteria. Moreover, the prediction horizon can be reduced when fuzzy objective functions are used, without deteriorating the control performance, when proper fuzzy objectives are designed.

9.4.3.2 *Air-conditioning system*

The air-conditioning system is simulated and a rate constraint of $|\Delta u| \leq 0.1$ is imposed on the system in these experiments. In this system H_c is chosen equal to 2, and H_p is chosen equal to 3. These horizons were revealed to be sufficient for controlling the system. It is required that the controller can bring the system to any level in the interval $[30\ ^\circ\mathrm{C}, 60\ ^\circ\mathrm{C}]$, which is the interval where the temperatures

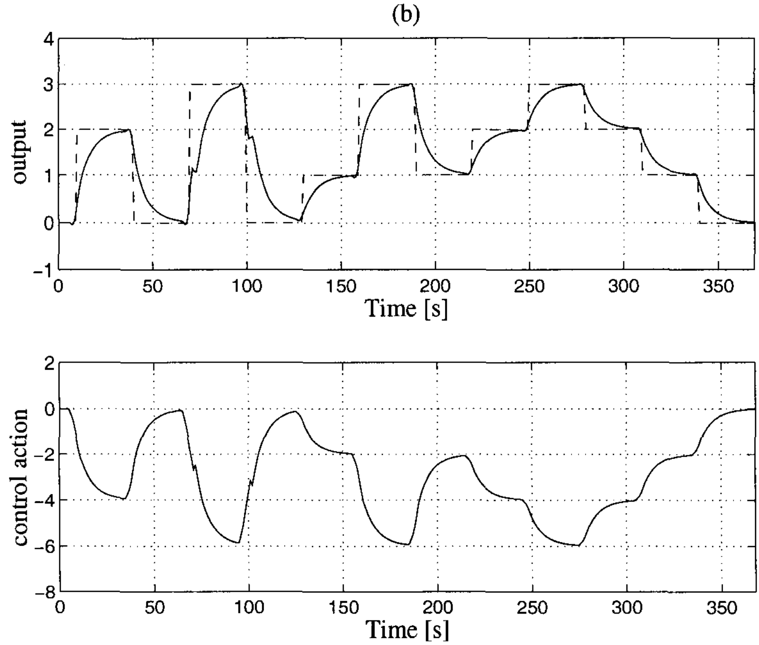


Fig. 9.9 Step responses for the linear system using the conventional objective function. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

usually range for this system. The output error with $n_{1l} = n_{1u} = 3$, the change in the control action with $n_{2l} = n_{2u} = 2$, and the change in the output with $n_{3l} = n_{3u} = 3$ are used to specify the objective function. The second change in the control action, chosen by $n_{2l} = n_{2u} = 2$, can be considered as a gradual transition between the control horizon and the prediction horizon. The first element in the control horizon is allowed to change freely within the crisp constraint on Δu, while the change is zero outside the control horizon. Including the second term in the objective function imposes a soft constraint on the change of the second control action, which reduces the oscillations of the control signal without slowing down the response of the system. The output error and the change in output are just considered for the final step $\tau + H_p$, because it requires less control effort in the system. Moreover, the use of the two first steps deteriorates the control performance due to the severe non-minimum phase behavior detected at some regions of the system's response.

The following parameters are used for the conventional objective function: $w_{13} = 1$, $w_{22} = 500$ and $w_{33} = 50$. The rest of the parameters are zero. The parameters w_{22} and w_{33} are chosen to make a trade-off between several criteria,

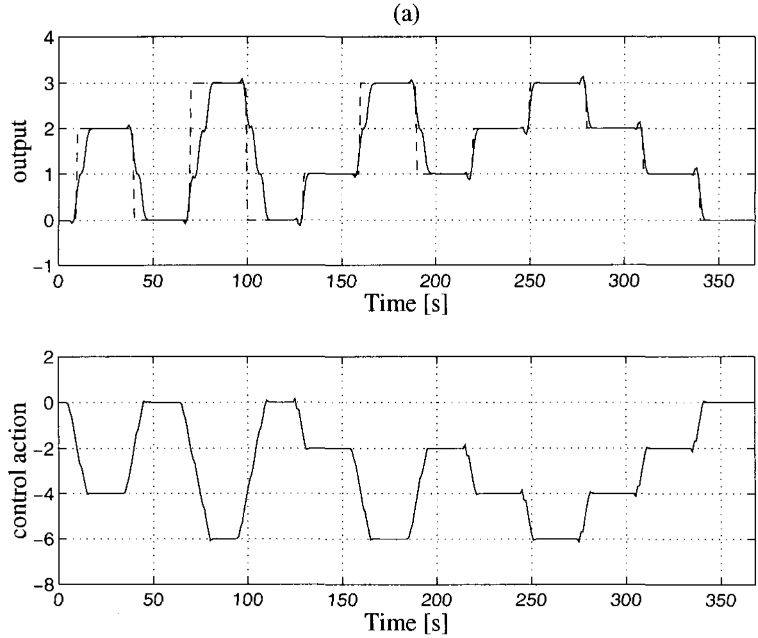


Fig. 9.10 Step responses for the linear system using the fuzzy objective function. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

and to scale different terms, namely the error, the change in control action and the change in the output. Note that the fuzzy objective function does not require this scaling due to the normalization introduced by the fuzzy sets. For the fuzzy criteria, the following membership function parameters are used: $K_e^+ = -K_e^- = 30$, $K_y^+ = -K_y^- = 3$, $S_y^+ = -S_y^- = 1$, $K_u^+ = -K_u^- = 0.6$, and $\gamma = 2$. Although nine parameters are present, only five must be tuned because the others are related to them. The parameter K_e^+ is chosen as the maximum error allowed for the system. K_y^+ is the maximum change allowed in the output. S_y^+ must be smaller than K_y^+, and this is the region where the temperature can change without being penalized. The parameter K_u^+ is chosen such that the valve can change almost freely (the total range is the interval $[0, 1]$), because the valve in the real system can change in this way, and the constraint in this valve is made for energy saving and stability reasons. Finally, the parameter for the Yager t-norm, $\gamma = 2$, allows for a good compromise between fast response and small overshoot, see Sec. 9.3.1. The responses of the air-conditioning system for several steps using classical and fuzzy criteria are shown in Fig. 9.11 and Fig. 9.12, respectively. The controller with the fuzzy criteria is more able to use the full range of control actions, and the

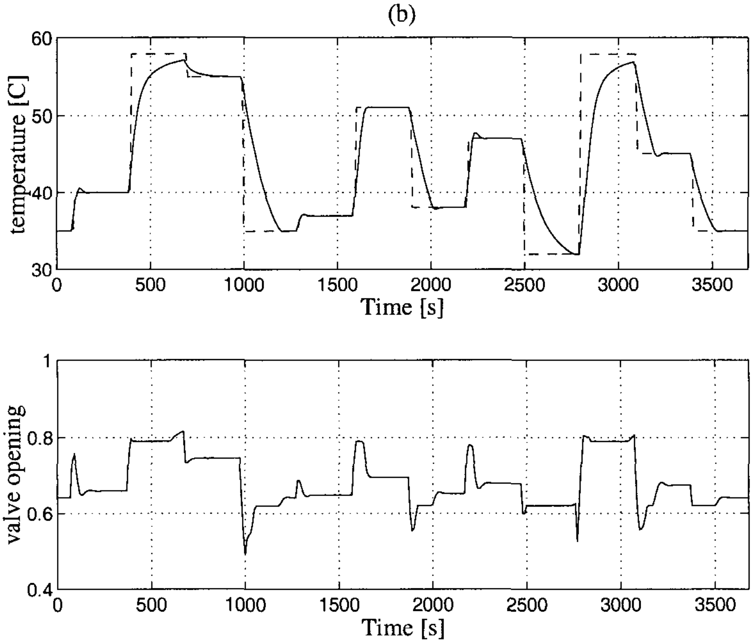


Fig. 9.11 Step responses for the air-conditioning system using the conventional objective function. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

response of this controller is in general faster, especially for references close to the limits of the range within which they can vary. Furthermore, some overshoots that are noticeable with the conventional criteria are reduced.

In summary, for the studied systems, the use of fuzzy criteria improves the response of the predictive controller when the parameters of the objective functions are tuned in order to obtain fast system response without overshoot. Despite the additional number of parameters, tuning the fuzzy criteria is not more tedious than tuning the conventional objective function because of a better understanding of the influence of the various parameters. The main disadvantage of the model predictive control with fuzzy criteria is that the optimization problem often becomes non-convex, which increases the computational load. Optimization issues for fuzzy model-based control are discussed in Chapter 10 and Chapter 11.

9.5 Design of decision functions from expert knowledge

One of the properties of fuzzy predictive control is that it provides a means to translate transparently the linguistic goals and preferences of the control engineer

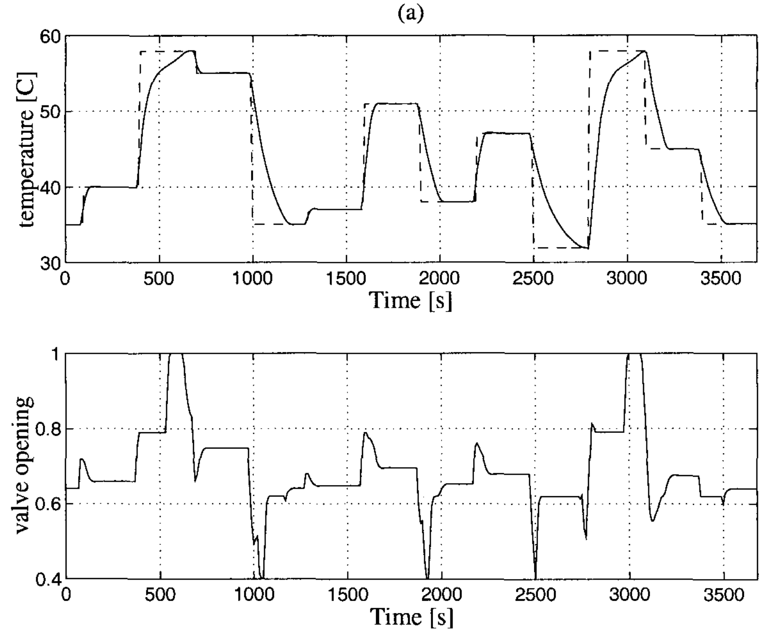


Fig. 9.12 Step responses for the air-conditioning system using the fuzzy objective function. Reproduced from (Sousa and Kaymak 2001), ©2001 IEEE.

into an objective function that can be used for optimization. Often, human experience and knowledge is available for controlling a process. Especially for systems which are difficult to control, the heuristic used by humans may provide good control response. A disadvantage of the model-based predictive control paradigm, however, is that the control strategy of a human cannot often be translated into a simple objective function that the predictive controller can use. Fuzzy predictive control provides a means for this translation by the use of many types of decision functions from fuzzy decision making. Humans use a variety of methods to achieve the control goals whereby the goals set for the controller as well as the controller parameters can be made time and state dependent. Therefore, the translation of a human's control strategy into a decision function involves dealing with time-varying goals and the adaptation of the parameter values. In this section, the design of a decision function that mimics the control strategy of an experienced crane driver is discussed for a simulated gantry crane, as first presented in (Kaymak and Sousa 1997).

9.5.1 *System description*

A container gantry crane consists of a bridge girder on portal legs from which a trolley system is suspended. The trolley can travel along the bridge girder that stretches over the container ship and part of the quay for loading and unloading the ship. A hoisting mechanism consisting of a spreader suspended from the trolley by means of hoisting cables is used for grabbing and hoisting the container. The control goal is to position the trolley at a desired horizontal location x while the swing θ of the load is damped so that the container can be positioned accurately (see Fig. 9.13). Moreover, the transport of the container must be done as fast as possible. Two motors, one for the trolley motion and one for hoisting, generate the torque T1 and T2 required to move the container and the load.

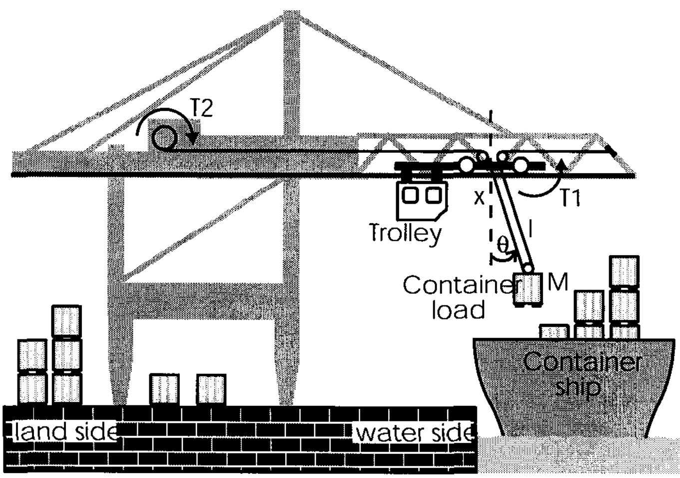


Fig. 9.13 Schematic representation of a container gantry crane.

To study a crane system, a simulation model is implemented using the Lagrangian of the system (Sakawa and Shindo 1982). The model is extended with the models of electric motors. The parameters of the models have been chosen from a real crane that is in use. Certain parasitic effects such as viscous friction have also been modeled. The trolley can reach a maximum velocity of 3.2 ms $^{-1}$ for a maximum load of 53 tons. The crane construction is assumed to be stiff, but the largest acceleration is bounded in order to avoid overloading the mechanical construction. This also limits the rate of change of the input voltage and current. The vertical motion of the load is not considered in this study, and it is assumed that the cable length is constant during the load transport.

9.5.2 *Expert control*

An experienced crane driver tries to reduce the load swing during transport as well as on arrival at the desired position. Since the trolley would not move if the swing is zero at all times, the crane driver accelerates first, making sure that the load swing is small at the end of the acceleration period. This means that the load lags behind initially and then catches up with the trolley. Ideally, the trolley must have reached its maximal speed at this time. Afterwards, the crane driver starts braking. The load swings in the travel direction and comes to rest with the trolley aligned on top of it. If there is a small error in the trolley position, it is corrected by 'creeping' the trolley to the desired location. However, this is avoided as creeping takes a lot of time. Figure 9.14 shows the trajectory for the load swing when the crane is controlled according to a crane driver's strategy.

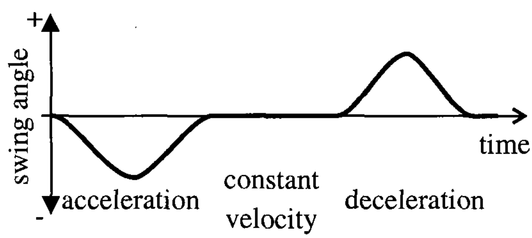


Fig. 9.14 Trajectory for the load swing when the crane is controlled according to a crane driver's strategy.

9.5.3 *Design of objective function*

In this section, an objective function is derived such that the resulting predictive controller mimics the control strategy of an experienced crane driver. The control strategy of a crane driver consists of two consecutive parts.

(1) Minimize load swing on achieving the maximal trolley speed.
(2) Minimize load swing on bringing the trolley to rest.

Minimizing the load swing can be described by 'small swing angle' and 'small swing angle speed', which are represented by triangular membership functions as shown in Fig. 9.15a and Fig. 9.15b. These goals do not change during the transport. There are two different goals concerning the trolley speed, each of which becomes dominant at different stages of the transport. Initially, the trolley speed must be large, which can be represented by a trapezoidal membership function (Fig. 9.15c). The trolley must stop at the desired position and thus the final goal becomes to attain a small trolley speed represented by a triangular membership function (Fig. 9.15d). Hence, at the beginning of the transport the former (large trolley speed) is the goal concerning the trolley speed and at a certain time (to be discussed later) the goal changes to the latter (small trolley speed). Note that it is not realistic to expect to satisfy all these goals at all sample instants within the prediction horizon, because the swing angle can only be reduced after it becomes large initially. For that reason, the three goals concerning the swing angle, swing angle speed and trolley speed need not be satisfied during the initial samples within the prediction horizon. Note also that the minimization of the transport time is implicit in the defined criteria. The optimization algorithm tries to maximize the velocity at all samples within the prediction horizon, which implicitly minimizes the transport time.

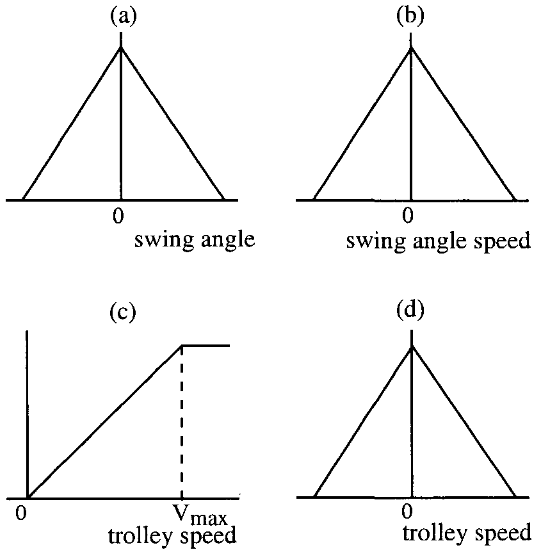


Fig. 9.15 Membership functions representing the goals for the predictive controller: (a) small swing angle; (b) small swing angle speed; (c) large trolley speed; (d) small trolley speed.

The criteria must be satisfied simultaneously in order to achieve the control goals. For that reason they are combined using a t-norm. Since the states trolley speed, swing angle and swing angle speed are closely related and influence each other, some interaction amongst the criteria is required. The minimum operator is not a suitable decision function, as it does not allow interaction amongst the criteria. Note that the objective function does not consider information concerning the trolley position. Because of that, the time at which the goal changes from large trolley speed to small trolley speed is important for the steady-state error in trolley position. Switching too soon means that the trolley starts to brake too early and the final position is not obtained, while switching too late implies that the trolley overshoots in position. The steady-state error can be reduced by switching to another control strategy (*e.g.*PI control) when the trolley comes to near stop. However, this results in creeping the trolley and should be avoided as much as possible. For that reason, a weight factor approach is followed in the following. In this approach, different criteria are weighted differently. The weight factors are adapted dynamically during the transport according to the distance of the trolley from the desired position. Initially, the weight factors for reducing swing angle are small, and they are increased in size gradually as the trolley is displaced. A fourth membership function regarding the position error of the trolley is added on approaching the desired position, and its weight is increased as the trolley approaches the desired position. These measures lead to improved steady-state error and the removal of oscillations during the transient phase. Hence, the weight factors implicitly determine the time period for the gradual passage from the first set of goals to the second set of goals.

9.5.4 *Simulation experiments*

This section shows the results of a simulation run obtained for the horizontal transport of the load when the crane is controlled by a fuzzy predictive controller. The criteria for the objective function and the corresponding membership functions have already been described in Sec. 9.5.3. Yager t-norm (Yager 1980) with a parameter value of $\gamma = 2.5$ has been used to combine the criteria. The prediction horizon is equal to 5, and the control horizon is equal to 2. The total decision is obtained using the aggregation function

$$\mu_{\pi} = \max\left(0, 1 - \left(\sum_{j=1}^{M} w_j(1-\mu_j)^{\gamma}\right)^{1/\gamma}\right), \quad \gamma > 0. \tag{9.19}$$

where M is the total number of decision criteria.

Figure 9.16 shows an example response of the system for a horizontal displacement of 60m with a sample rate of 0.75 s and a cable length of 30m from the suspension point to the container. The model–plant mismatch is not considered. No explicit reference trajectory is specified for the controller, which is an advantage of this method. The mass that is transported is 33 tons. As it can be seen from the trajectory for the swing angle, the controller approximates the control strategy of an experienced crane driver. This is achieved by implementing an adaptive objective function used by the optimization routine of the model-based predictive controller.

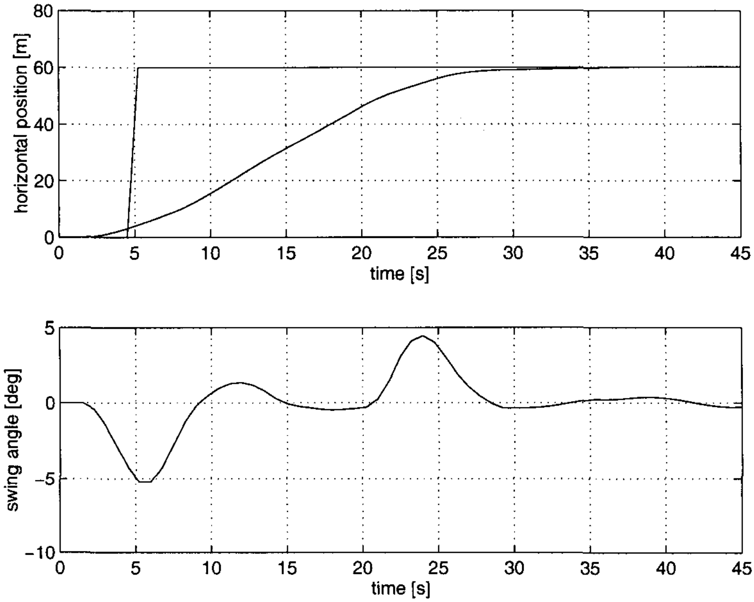


Fig. 9.16 Step response for a horizontal displacement of 60m.

9.6 Summary and concluding remarks

The application of fuzzy decision making to predictive control in closed-loop control systems is considered in this chapter. Fuzzy decision making applied to real-time control is based on multistage decision making. In this chapter multistage FDM is formulated for a control environment. A discussion on the types of systems, and the termination time defined for the multistage approach has been given in Sec. 9.1, presenting some references for the different solutions found in the past. Fuzzy goals and constraints defined for the control environment, and their respective aggregation are considered in Sec. 9.2. The use of fuzzy criteria in model-based predictive control is also addressed in this section.

FDM in control has two main design problems, which are the choice of the aggregation operators and the choice of fuzzy criteria. Fuzzy criteria for FDM in control is presented in Sec. 9.3. First, a discussion on the choice of the aggregation operators is presented in Sec. 9.3.1. The use of parameterized aggregation operators has some advantages, because these parameters can influence several criteria at the same time. The relation amongst various performance criteria, such as the rise time, the settling time or the overshoot, can then be adapted with only one parameter.

The possible types of fuzzy criteria used for FDM in model predictive control are discussed in Sec. 9.3. The choice of the prediction horizon is discussed, and the generalization of classical objective functions to fuzzy objective functions in MBPC is presented. Contrary to the symmetric formulation of fuzzy decision making, we make here a clear distinction between the representation of goals and the representation of constraints in control problems. Consequently, exponential curves can be used to represent fuzzy goals, while membership functions with bounded support are used to represent fuzzy constraints. The advantage of this approach is that the hard constraints of the control problem are guaranteed to be satisfied, provided that the aggregation of constraints is conjunctive. The choice of the prediction horizon is addressed, and the generalization of classical objective functions to fuzzy objective functions in MBPC is presented. This generalization brings additional flexibility to the definition of the objective functions.

Two examples are presented in Sec. 9.4. The examples show the improvements of the controller response by using fuzzy objective functions in MBPC. However, the optimization problem is non-convex with the known disadvantages. The computational time grows exponentially with the control horizon and the number of variables. These problems are discussed in Chapter 10 and Chapter 11.

Fuzzy aggregation operators introduce additional flexibility in designing an objective function to achieve control goals. The design of an objective function from expert knowledge has been demonstrated by using a simulated gantry crane as an example. This is achieved by using weighted fuzzy aggregation operators from Chapter 3, whose weights are adapted dynamically to account for the changing goals in the human expert's control strategy.

Chapter 10

Derivative-Free Optimization

Model-based control usually demands the optimization of an objective function. This is always the case in model predictive control. Sometimes, the optimization problem in MBPC has an analytical solution, provided that (see also Sec. A.3):

(1) a linear model of the system is used,
(2) the objective function is described by

$$J(\mathbf{u}) = \sum_{j=1}^{H_p} w_{1j}(\mathbf{r}(\tau + j) - \hat{\mathbf{y}}(\tau + j))^2 + w_{2j}(\Delta\mathbf{u}(\tau + j - 1))^2, \quad (10.1)$$

or a similar quadratic equation, and
(3) no constraints are active.

When some constraints are violated, a general analytical solution is not available. However, the optimization problem is still a quadratic problem, provided that the constraints are linear in the optimization variables. This problem is convex, and it can be solved by using quadratic programming with a guaranteed global optimum. However, in the most general, case both nonlinear models and constraints are present, and the optimization problem results in a non-convex problem.

Classical techniques used to solve non-convex optimization problems are the *sequential quadratic programming* method, see, *e.g.*, (Gill et al. 1981) and the *simplex method* introduced by Nelder and Mead (1965), which are both iterative optimization techniques. These iterative methods have generally high computational costs, and the solution may converge to local minima. When the solution space is discretized, alternative optimization methods for non-convex optimization problems, such as dynamic programming, branch-and-bound or genetic algorithms, can also be applied.

Amongst the discrete optimization techniques, dynamic programming (DP) is one of the most utilized. This technique is based on the *principle of optimality* introduced by Bellman (1957). The computation of the solution usually proceeds 'backwards', *i.e.*, from step $\tau + H_p$ to step $\tau + 1$. The principle of optimality is not changed when 'forward' dynamic programming, *i.e.*, computing from step $\tau + 1$ to step $\tau + H_p$, is utilized. In predictive control, on-line implementation does not allow heavy computational effort, which is usually the case when dynamic programming is utilized. Furthermore, DP requires the discretization of the control inputs, outputs and states, contrary to branch-and-bound and genetic algorithms, where only the control actions must be discretized if a continuous model of the system under control is available. This fact allows for more accuracy in the computation of the optimal control actions. Therefore, branch-and-bound and genetic algorithms are considered for discrete optimization problems in this book.

This chapter is divided into two main parts. The first part presents a branch-and-bound algorithm for solving non-convex optimization problems encountered in predictive control using nonlinear models. This is discussed in Sec. 10.1 for conventional objective functions. Section 10.2 proposes a branch-and-bound algorithm for predictive control with fuzzy decision functions. The performance of the branch-and-bound algorithm for fuzzy decision functions is illustrated with an example in Sec. 10.3. The second part considers a different optimization method, where a genetic algorithm is used in Sec. 10.4. Section 10.5 illustrates this method with an example, before the concluding remarks of the chapter in Sec. 10.6.

10.1 Branch-and-bound optimization for predictive control

A widely used technique to solve difficult (usually non-convex) optimization problems is the branch-and-bound method (Mitten 1970). Branch-and-bound algorithms (B&B) solve optimization problems by partitioning the solution space. In this method, the set of solutions is subsequently partitioned into increasingly refined parts (branching) over which lower and upper bounds for the optimal value of the objective function can be determined (bounding). Usually a minimization problem is considered, and the optimization is performed by finding a feasible solution with minimal value. Branch-and-bound methods have been applied to the solution of various constrained optimization problems, such as integer linear programming, nonlinear programming, the traveling salesman problem or the quadratic assignment problem (Horowitz and Sahni 1978). This book utilizes branch-and-bound to solve non-convex optimization problems in predictive control. Therefore, this section presents the branch-and-bound algorithm developed for classical predictive control (Sousa et al. 1997), and Sec. 10.2 presents a branch-and-bound algorithm for optimization in predictive control with fuzzy decision functions introduced by Sousa (2000).

A branch-and-bound algorithm can be characterized by the following three rules.

(1) *Branching rule* - defines how to divide a problem into sub-problems.
(2) *Bounding rule* - establishes lower and upper bounds in the optimal solution of a sub-problem. These bounds allow for the elimination of sub-problems that do not constitute an optimal solution.
(3) *Selection rule* - defines the next sub-problem to branch from.

Usually, these three basic rules are applied recursively in B&B methods. Some B&B algorithms apply search heuristics for the selection rule, and for guiding the memory storage of the already explored sub-problems, thereby improving the efficiency of the method significantly (Chen and Bushnell 1996).

10.1.1 *B&B in predictive control*

When the control actions are discretized, the branch-and-bound method can be applied to predictive control. The general MIMO model is given in Sec. 5.1 by Eq. (5.4). For the sake of simplicity, the B&B algorithm is presented here for SISO systems, but it can be generalized for MIMO systems (Roubos et al. 1999). Some remarks on the necessary procedures for this generalization are made during the description that follows. In order to make the description more clear, the state vector does not include the actual control action $u(\tau)$, and is given by

$$\mathbf{x}(\tau) = [y(\tau), \ldots, y(\tau - p_y + 1), \ldots, u(\tau - 1), \ldots, u(\tau - m_u + 1)]^T, \quad (10.2)$$

where m_u is the order of the input and p_y is the order of the output. With this state vector, the model of the system under control predicts the future outputs of the system $\hat{y}(\tau + 1), \ldots, \hat{y}(\tau + H_p)$, and is given by

$$\hat{y}(\tau + j) = f(\mathbf{x}(\tau + j - 1), u(\tau + j - 1)), \quad j = 1, \ldots, H_p\,, \quad (10.3)$$

where f is a function describing the system. The output values $\hat{y}(\tau + j)$, $j = 1, \ldots, H_p$, are calculated based on the state vector at time instant $\tau + j - 1$ and on the future control signal $u(\tau + j - 1)$, which are determined by optimizing a given objective function. Note that the state vector at time step $\tau + j - 1$ contains predicted and real input and output values. Let the possible inputs of the system be discretized in n_d possible control actions. Also, let the discretized control actions be denoted ω_i. Thus, at each step, the control actions $u(\tau + j - 1) \in \Omega$ are given by

$$\Omega = \{\omega_i | i = 1, 2, \ldots, n_d\}. \quad (10.4)$$

Note that this set can be seen as the set of possible alternatives in a multidimensional fuzzy decision making problem. Thus, the problem considered here is just a particular case of fuzzy decision making in control (which is presented in Sec. 9.1) and where the classical objective function of Eq. (10.1) is used. Branch-and-

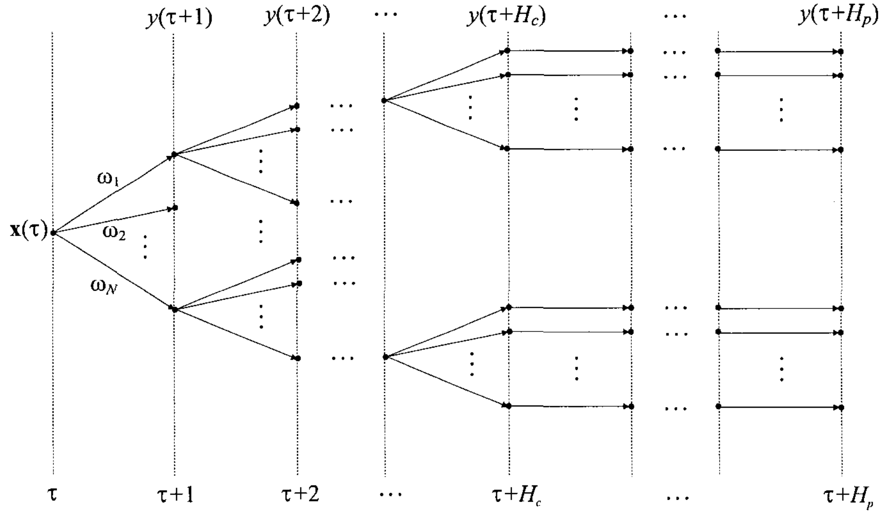


Fig. 10.1 Branch-and-bound optimization applied to predictive control.

bound methods can be visualized by a search tree, see Fig. 10.1. In this figure, the subsequent steps are depicted, and each point indicates a possible value for the predicted outputs $\hat{y}(\tau + j)$, $j = 1, \ldots, H_p$. In predictive control, the problem to be solved is normally represented by the objective function in Eq. (10.1)

$$J = \sum_{j=1}^{H_p} w_{1j}(\hat{e}(\tau + j))^2 + \sum_{j=1}^{H_c} w_{2j}(\Delta u(\tau + j - 1))^2 ,$$

where $\hat{e}(\tau+j)$ is the predicted output error given by $\hat{e}(\tau+j) = r(\tau+j)-\hat{y}(\tau+j)$. Other objective functions can also be considered, such as the one presented in Eq. (9.9). The optimization problem is successively decomposed by the branching rule into smaller sub-problems.

A sub-problem in the middle of the tree can be defined as follows. At time instant $\tau + j$ the cumulative cost of a certain path followed so far, leading to the state $\mathbf{x}(\tau + j)$ and output $\hat{y}(\tau + j)$, is given by

$$J^{(j)} = \sum_{l=1}^{j} \left[w_{1l} \left(r(\tau + l) - \hat{y}(\tau + l)\right)^2 + w_{2l}(\Delta u(\tau + l - 1))^2\right], \qquad (10.5)$$

where $j = 1, \ldots, H_p$, denotes the level corresponding to the time step $\tau + j$ (see Fig. 10.2). A particular branch i at level j is created if the cumulative cost $J^{(j)}(u)$ plus a *lower bound* on the cost from the level j to the terminal level H_p for the branch i, denoted J_{L_i}, is lower than an *upper bound* of the total cost, denoted J_U, *i.e.*,

$$J^{(j)} + J_{L_i} < J_U. \tag{10.6}$$

Let the total number of branches satisfying this rule at level j be given by N. Figure 10.2 illustrates the sub-problem at level j, where a particular state and output are considered. In order to increase the efficiency of the B&B method, it is required that the number N should be as low as possible, *i.e.*, $N \ll n_d$. Note that in the worst case $N = n_d$, and all the possible branches are generated for the different alternatives ω_j, $j = 1, \ldots, n_d$. The lower bound can be expressed as a

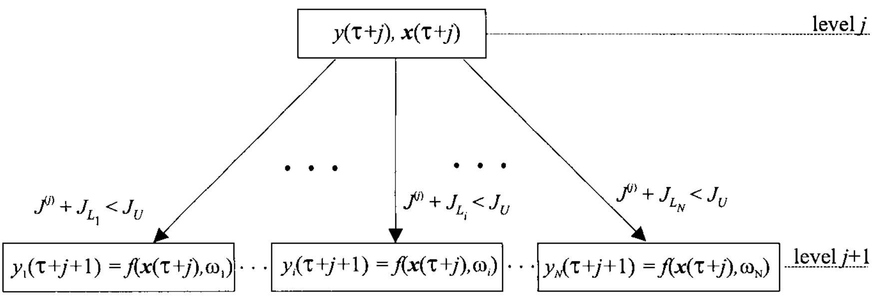


Fig. 10.2 One step of the branch-and-bound algorithm used in MBPC.

sum of two terms,

$$J_{L_i} = J_i^{(j)}(\omega_i) + J_L(j+2). \tag{10.7}$$

The first term, $J_i^{(j)}(\omega_i)$, is the cost associated with the transition $\hat{y}(\tau + j + 1) = f(\mathbf{x}(\tau+j), \omega_i)$, which is computed by evaluating the respective element in the cost function in Eq. (10.5). The second term is an estimated lower bound of the cost over the remaining steps $j+2, \ldots, H_p$, denoted $J_L(j+2)$, which is generally not known and must be estimated. Note that no branching takes place for $j > H_c - 1$ (beyond the control horizon), *i.e.*, the last control action $u(\tau + H_c - 1)$ is applied successively to the model, until H_p is reached.

In order to achieve $N \ll n_d$, the upper bound should be as low as possible (close to the optimal solution of the entire problem), and the lower bound as large as possible, so that the number N of new branches is decreased.

Note that until now only the branching rule and the bounding rule have been defined. The selection rule, which determines the way of searching for the best solution, must be selected from various possibilities. A particular *heuristic search* is proposed here, which applies depth first, breadth first and best-bound search in different stages of the optimization process (Ibaraki 1976). The concept of *heuristic search* provides a framework to compare different types of searches, *e.g.*depth first, breadth first, or best-bound search. The heuristic must govern the order in which the sub-problems are branched from, such that the branching is done from the sub-problem with the smallest heuristic value. The heuristic search applied in the branch-and-bound method for predictive control is described in the following.

First, an initial upper bound $J_U = J^{(H_p)}$ must be estimated. The cost $J^{(H_p)}$ is derived by branching n_d times (for all possible control actions) at each level j, starting at level 0. The smallest $J_i^{(j)}(\omega_i)$ is chosen at each level j. At this stage, the remaining nodes created at level H_c can already be eliminated because they do not constitute an optimal solution. With this initial upper bound, the algorithm goes back one level (to level $H_c - 1$), and chooses the second best branch found so far. This branch is expanded by applying the branch condition in Eq. (10.6). The lower bound of the cost over the remaining steps $J_L(H_c)$ must then be estimated. The cost associated with the transition to $\hat{y}(\tau + H_c)$ is estimated. The remaining cost $J_L(H_c + 1)$ is also estimated. If no better estimate is possible, it is set to zero: $J_L(H_c + 1) = 0$. The number of branches generated, N, is the one that fulfills the branch condition in Eq. (10.6). The new nodes are at level H_c, and must be compared to the best solution found so far (given by J_U). If a new optimal solution is found, J_U is replaced by this new $J^{(H_p)}$, and the best solution found so far is updated to this new value. As the upper bound is now smaller, the number of branches which are generated tends to be smaller, and the general optimization faster. After this branch is fully explored, the best of the remaining branches that still fulfill the branch conditions at level $H_c - 1$ is tested. This procedure is repeated until only one branch remains at level $H_c - 1$. Then, the algorithm goes back one level, to $H_c - 2$, and starts the branch-and-bounding procedure again as described for the level $H_c - 1$. The algorithm stops when there are no branches left to be explored, and then each level has only one node. This path is the optimal solution. This branch-and-bound method applied to predictive control is described in Algorithm 10.1.

Algorithm 10.1 Branch-and-bound algorithm applied to MBPC.

Choose the control and prediction horizons, H_c and H_p, respectively. Choose the number of discrete control actions ω_i, $i = 1, \ldots, n_d$.

Step 1: **Initialize algorithm.** At each level j (time $\tau + j$), starting from level 0, the smallest $J_i^{(j)}(\omega_i)$ is chosen, and branching is made for all possible discrete control actions n_d. The best cost at step H_p, $J^{(H_p)}$ is chosen as the initial upper bound:

$$J_U = J^{(H_p)}.$$

The remaining $n_d - 1$ nodes created at level H_c are eliminated because they do not constitute an optimal solution. The algorithm goes back one level to $H_c - 1$.

Step 2: **Estimate lower bound.** The algorithm is at level j. The branch i with the best cost function $J^{(j)}$ found so far, and that is not fully explored, is chosen. This means that if the best cost function at level j has already branches to level $j+1$, the second best value for $J^{(j)}$ must be chosen. The lower bound J_{L_i} is estimated by

$$J_{L_i} = J_i^{(j)}(\omega_i) + J_L(j+2),$$

with $J_i^{(j)}(\omega_i) = w_{1j}\left(r(\tau+j+1) - \hat{y}(\tau+j+1)\right)^2 + w_{2j}(\omega_i - u(\tau+j))^2$, and $\hat{y}(\tau+j+1) = f(\mathbf{x}(\tau+j), \omega_i)$. The lower bound on the cost over the remaining steps must be estimated. If no estimate is available, it is simply set to zero, *i.e.*, $J_L(j+2) = 0$.

Step 3: **Apply branch condition.** The branching condition

$$J^{(j)} + J_{L_i} < J_U\,,$$

is applied to the considered branch at level j for all $i, i = 1, \ldots, n_d$ discretized control actions ω_i. This procedure generates N branches. If no branch is generated, go to step 6.

If $j + 1 = H_c$,

Step 4: **Compute a new optimal solution.** Compute the outputs from H_c to H_p and the respective costs. Compare the optimal cost found so far, J_U, with the N new costs. If a new optimal solution is found, J_U is replaced by the new $J^{(H_p)}$. Update the best solution found so far. Eliminate the nodes with non-optimal solutions.

Else

Step 5: **Branch from the best generated node.** Choose the smallest cost $J^{(j)}(\omega_i)$ and go to the next level ($j \rightarrow j+1$). Go to Step 2.

Step 6: **Go up in the tree.** Go up in the levels of the tree until a non-totally-explored branch is found and afterwards go to Step 2. If all the branches are fully explored, the optimal solution is the optimal solution found so far, and the algorithm stops.

10.1.2 *Application of the B&B method to nonlinear control*

In this section, the advantages of the B&B method are illustrated by an example that compares the performances of the branch-and-bound algorithm and the sequential quadratic programming (SQP) algorithm, see, *e.g.*, Gill et al. (1981). The TS fuzzy model derived for an air-conditioning system, and presented in Sec. 12.3.1 is used as an example. This model is used both as model and system in the predictive control scheme, avoiding model–plant mismatches and disturbances, and allowing for a proper comparison between the two algorithms. Figure 10.3 gives the sum of squared errors (SSE) as a performance measure,

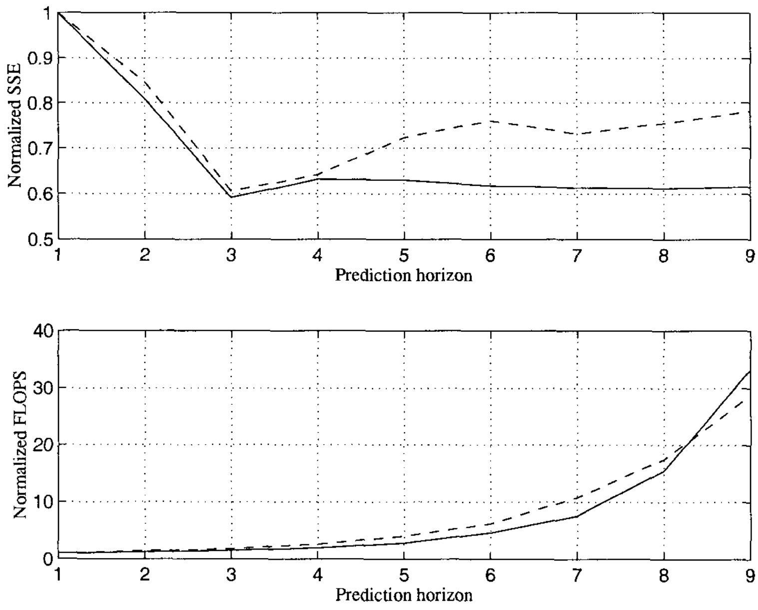


Fig. 10.3 Sum squared error and FLOPS for the optimization methods. Solid line – B&B optimization; dashed line – sequential quadratic programming.

and the number of floating-point operations (FLOPS) as a measure of the computational costs of the two algorithms. The computational requirements and the SSE of the B&B method for $H_c = 1$ are normalized to 1 (or 100%). The comparison is made for control horizons from 1 up to 9 steps. One can see that for the SQP optimization method, the error is always bigger than the error obtained when the B&B method is utilized, because the SQP method often converges to local minima. The computational costs of SQP are higher until $H_c = 8$, and are lower afterwards. However, the control horizon is usually kept much smaller (approximately 2 to 6), where the branch-and-bound method gives better computational efficiency. On the basis of this comparison, it can be concluded that the B&B optimization method is superior to SQP with respect to the performance achieved, and also with respect to the computational costs when the control horizon is kept small.

This example shows that for large control horizons, another optimization method is necessary to find the optimal control actions in predictive control. It is clear that the same happens for large numbers of control alternatives and for MIMO systems, with several control inputs. In fact, for these systems the computational effort grows exponentially, limiting the use of the B&B algorithm. However, Roubos et al. (1999) have recently proposed a modified B&B algorithm able to deal with relatively slow MIMO systems. Another non-convex optimization method is presented in Sec. 10.4, where genetic algorithms are used for optimization in predictive control.

10.1.3 *Evaluation of the B&B method applied to MBPC*

The results obtained using the proposed B&B algorithm have been compared to other optimization techniques, such as Sequential Quadratic Programming (SQP). The experience shows that the B&B algorithm, even with a rough lower bound estimate, $J_L(j+2)$, is in general faster and more accurate than *enumerative search*, which explores the complete search space. It is also faster and more accurate than the SQP method, which tends to converge to local minima for nonlinear systems. However, the computational time increases exponentially with the control horizon, demanding that this be kept low. A factor of extreme importance is the number of control alternatives n_d. This number should be as small as possible for computational reasons. However, if n_d is too small, the coarse discretization of the control signal results in poor control performance. Therefore, a good compromise between the computational effort and the size of n_d must be made for each control problem.

Three major advantages of the B&B algorithm applied to predictive control over other non-convex optimization methods are the following.

(1) The global *discrete* minimum containing the optimal solution is always found, guaranteeing good control performance.
(2) The algorithm does not need an initial guess, and hence its performance cannot be negatively influenced by a poor initialization, as in the case of iterative optimization methods.
(3) The B&B method implicitly deals with constraints. Moreover, the constraints improve the efficiency of bounding, restricting the search space by eliminating non-feasible sub-problems. Most optimization algorithms such as SQP, have

difficulties in dealing with constraints, which is reflected in their performance.

Two serious drawbacks of B&B are the exponential increase of the computational time with the control horizon and the number of alternatives, and the discretization of the possible control actions. This discretization can cause oscillations of the outputs around the reference trajectory. A possible solution for this last problem is described in Sec. 7.5, where a predictive control strategy is combined with inverse control. Section 11.3 presents another solution to cope with the same problem by using fuzzy predictive filters.

10.2 Branch-and-bound optimization for fuzzy predictive control

In fuzzy predictive control, the membership functions for the fuzzy criteria can have an arbitrary shape and the decision function is usually nonlinear, which often results in a non-convex optimization problem. Hence, convex optimization algorithms (*e.g.* such as the ones presented in Sec. 11.1) cannot be used. However, the decision problem can be formulated as a discrete choice problem where a selection is made out of a set of possible alternatives. For formulating the discrete choice problem, the control space is discretized and the problem is reduced to searching the best control action in the discretized control space. Due to this discretization an approximate solution is obtained. The search for a solution can be performed by using the branch-and-bound (B&B) method. Previously, a B&B algorithm for classical model-based predictive control applications was presented in Sec. 10.1. This section discuss the application of the B&B algorithm to model predictive control with fuzzy decision criteria in the objective function. More details can be found in Sousa (2000).

Remember that the branch-and-bound method (Horowitz and Sahni 1978) is a structured search technique belonging to a general class of enumerative schemes. When the control actions are discretized, branch-and-bound can be utilized as the optimization method in predictive control. The model predicting the future outputs of the system $\hat{y}(\tau+1), \ldots, \hat{y}(\tau+H_p)$ is already given in Eq. (10.3), and it is noted here for the sake of clarity:

$$\hat{y}(\tau+j) = f(\mathbf{x}(\tau+j-1), u(\tau+j-1)), \quad j = 1, \ldots, H_p .$$

The considerations made in Sec. 10.1 about this model remain valid here. The control actions $u(\tau), \ldots, u(\tau+H_c-1)$ are discretized in n_d possible input values, as in Sec. 10.1, *i.e.*,

$$\Omega = \{\omega_i | i = 1, 2, \ldots, n_d\} .$$

The tree represented in Fig. 10.1 remains also valid, and at each time step (level of the tree shown in Fig. 10.1), n_d control alternatives are considered, yielding $N \leq n_d$ branches.

In fuzzy predictive control, the objective function is defined as the aggregation of fuzzy goals and constraints (fuzzy criteria). The symbol $\zeta_{j\ell}$ denotes criterion ℓ considered at time step $\tau + j$, with $j = 1, \ldots, H_p$ and $\ell = 1, \ldots, m$, where m is the total number of fuzzy criteria. These definitions are presented in Sec. 9.2.1. The aggregation of the different criteria is given by Eq. (9.2). In order to apply the branch-and-bound method to fuzzy predictive control, the aggregation operators $\circledast_g$, $\circledast_c$ and $\circledast$ must be t–norms. These norms guarantee that the membership degree of the policy μ_π decreases with the time, and the corresponding cost functions increase, which is a necessary condition to apply the B&B algorithm.

Let $j = 0, 1, \ldots, H_p$ denote the jth level of the tree ($j = 0$ at the initial node) and let i denote the branch corresponding to the control alternative ω_i. The partial optimization problem for a branch i at level j is defined by the maximization of the criteria at this point. Defining it in a recursive way, one obtains

$$\mu_\zeta^{(j)}(\omega_i) = T\left(\mu_\zeta^{(j-1)}, \mu_{\zeta_{j1}}(\omega_i), \ldots, \mu_{\zeta_{jm}}(\omega_i)\right) . \tag{10.8}$$

The operator T represents an arbitrary triangular norm, and thus, $\mu_\zeta^{(j)}$ is a decreasing function with respect to j. The membership value $\mu_\zeta^{(0)}$ for the cost level zero is set to 1 because this is the neutral element for t–norm operators. The optimization problem can be converted from a maximization into a minimization by taking the fuzzy complements of the membership values for the considered criteria. This transformation allows for the formulation of the optimization problem in a similar way to the classical B&B algorithm, where a new branch is generated if the cost at a certain point added to a lower bound of the remaining cost is smaller than an upper bound of the total cost. In order to transform the maximization in a minimization problem let $\mu_J^{(j)}(\omega_i)$ be the fuzzy complement of the membership degree representing the confluence of decision criteria, *i.e.*,

$$\mu_J^{(j)}(\omega_i) = \overline{\mu_\zeta^{(j)}} , \tag{10.9}$$

where $\bar{\cdot}$ stands for the fuzzy complement. This value can be seen as a cost because it increases with j; actually, it is the complement of a decreasing function. In formal terms, the partial optimization problem for a branch i at level j is formulated as

$$\min_{\omega_i \in \Omega} \mu_J^{(j)}(\omega_i). \tag{10.10}$$

Note that no 'hard' constraints are explicitly represented, because they are implied by the supports of the membership functions defining the satisfaction of the decision criteria, see Sec. 9.2.1. Moreover, constraints on the control actions are directly applied when the n_d discrete control actions are chosen. Application of the branching alone would result in the search of the entire tree (enumerative search), *i.e.*, $(n_d)^{H_c}$ possibilities, which is computationally prohibitive, except for very small control horizons. In order to reduce the number of alternatives, bounding is applied. At level j, the degree of satisfaction for the decision criteria is known and is given by $\mu_{\zeta}^{(j)}$, as in Eq. (10.8). Let $\mu_{\zeta_U}^{(j+1,\ldots,H_p)}$ be an upper bound of the remaining degree of satisfaction for the levels $j+1,\ldots,H_p$. Similar to the cost $\mu_J^{(j)}(\omega_i)$ for branch i at level j, the membership value of a lower bound for the remaining cost can be given by the complement of $\mu_{\zeta_U}^{(j+1,\ldots,H_p)}$;

$$\mu_{J_L}^{(j)} = \overline{\mu_{\zeta_U}^{(j+1,\ldots,H_p)}}. \tag{10.11}$$

Let μ_{J_U} be an upper bound for the total cost, that is given by the complement of the membership degree representing a lower bound on $\mu_{\zeta}^{(H_p)}$. This lower bound is the confluence of decision criteria when an entire path has been followed. A particular branch i at level j is followed if the cost at level j aggregated with the lower bound on the remaining cost $\mu_{J_L}^{(j)}$ is smaller than μ_{J_U}, *i.e.*, if

$$\begin{aligned}\overline{T(\mu_{\zeta}^{(j)},\mu_{\zeta_U}^{(j+1,\ldots,H_p)})} &= S\left(\overline{\mu_{\zeta}^{(j)}},\overline{\mu_{\zeta_U}^{(j+1,\ldots,H_p)}}\right)\\ &= S\left(\mu_J^{(j)},\mu_{J_L}^{(j)}\right) < \mu_{J_U},\end{aligned} \tag{10.12}$$

where S is the t–conorm that is the *dual* of the t–norm used in Eq. (10.8). For the sake of clarity, the dependency on ω_i is not explicitly shown in Eq. (10.12). The transition between level j and level $j+1$ is depicted in Fig. 10.4. The efficiency of the bounding mechanism depends on the quality of the bound estimates. The upper bound μ_{J_U} should be as close as possible to the optimum and the lower bound $\mu_{J_L}^{(j)}$ as large as possible, in order to decrease the number of new branches to be created by the branch-and-bound algorithm. The availability of these estimates depends on the particular problem. The algorithm for the branch-and-bound method applied to fuzzy predictive control is presented in Algorithm 10.2.

Algorithm 10.2 Branch-and-bound algorithm for fuzzy predictive control.
Choose the control and prediction horizons, H_c and H_p, respectively. Choose the number of discrete control actions ω_i, $i = 1,\ldots,n_d$.

Step 1: **Initialize algorithm.** At each level j (time $\tau + j$), starting from level 0, the smallest $\mu_J^{(j)}(\omega_i)$ is chosen, and branching is made for all possible

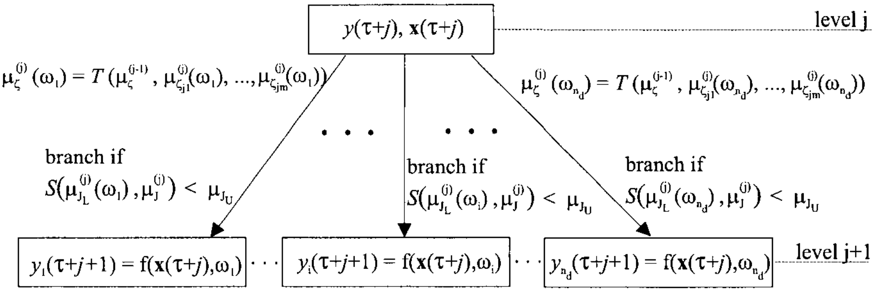


Fig. 10.4 Branch-and-bound optimization in fuzzy predictive control. Reprinted from (Sousa 2000) by permission of John Wiley and Sons, Inc., ©2000 John Wiley & Sons, Inc.

discrete control actions n_d. The best cost at step H_p, $\mu_J^{(H_p)}$ is chosen as the initial lower bound,

$$\mu_{J_U} = \mu_J^{(H_p)}.$$

The remaining $n_d - 1$ nodes created at level H_c are eliminated because they do not constitute an optimal solution. The algorithm goes to the level $H_c - 1$.

Step 2: **Estimate lower bound.** The algorithm is at level j. The branch i with the best cost function $\mu_J^{(j)}$ found so far, and that is not fully explored, is chosen. The lower bound $\mu_{J_L}^{(j)}$ is estimated by

$$\mu_{J_L}^{(j)} = \overline{\mu_{\zeta_U}^{(j+1,\ldots,H_p)}}.$$

If no information over $\mu_{\zeta_U}^{(j+1,\ldots,H_p)}$ is available, this lower bound is set to zero, *i.e.*, $\mu_{J_L}^{(j)} = 0$.

Step 3: **Apply branch condition.** The branching condition

$$S\left(\mu_J^{(j)}, \mu_{J_L}^{(j)}\right) < \mu_{J_U},$$

is applied to the considered branch at level j for all $i, i = 1, \ldots, n_d$ discretized control actions ω_i. This procedure generates N branches. If no branch is generated go to step 6.

If $j + 1 = H_c$,

Step 4: **Compute a new optimal solution.** Compute the outputs from H_c to H_p and the respective costs. Compare the optimal cost found so far, μ_{J_U}, with this N new costs. If a new optimal solution is found, μ_{J_U} is replaced by the new $\mu_J^{(H_p)}$. Update the best solution found so far. Eliminate the nodes with non-optimal solutions.

Else

Step 5: **Branch from the best generated node.** Choose the smallest cost $\mu_J^{(j)}(\omega_i)$ and go to the next level $(j \to j+1)$. Go to Step 2.

Step 6: **Go up in the tree.** Go up in the levels of the tree until a non-totally-explored branch is found and go to Step 2. If all the branches are explored, the optimal solution is the optimal solution found so far, and the algorithm stops.

Although the bound estimates can reduce the number of nodes generated in the tree, the computational complexity of the algorithm remains exponential which makes it prohibitively expensive for large control horizons and too many discrete control alternatives in Eq. (10.4). The B&B optimization technique applied to fuzzy predictive control always finds the global discrete optimum. However, the time to compute the optimum may vary. If the time to compute the global optimum is bigger than the sampling time of the system, the algorithm can be stopped and the last control sequence found so far can be used. Note that this local optimum may be the global optimum for some cases. The main drawbacks of this method are thus the computational complexity for large problems, and the restriction of the possible control actions to a set of discrete alternatives. On one hand, the number n_d of discrete control actions should be small, as the computing time of the branch-and-bound algorithm increases drastically with an increasing number of control alternatives. On the other hand, n_d should be large since a too coarse discretization may result in a rough control policy, inferior to those obtained with a finer discretization. In general, the discretization should be chosen such that oscillations of the outputs around the reference trajectory are sufficiently small. Fuzzy predictive filters presented in Sec. 11.3 can also be applied to solve this problem.

10.3 Application example for fuzzy branch-and-bound

In this section, the fuzzy branch-and-bound algorithm is applied to a test case consisting of the simulation of the air-conditioning system, as described previously in Sec. 9.4.1. The description of the system and the simulation conditions considered in that section are the same as the ones considered here. The model is again a SISO model, where the output, *i.e.*, the supply temperature $\mathrm{T}_s(\tau+1)$, is modeled by $\mathrm{T}_s(\tau+1) = f(\mathrm{T}_s(\tau), u(\tau))$, and where $u(\tau) \in [0.4, 1]$ is the valve opening.

Model-based predictive controllers are designed for the air-conditioning system by using a fuzzy objective function. A criterion consisting of the minimization of the output predicted error $\hat{e}(\tau+j) = r(\tau+j) - \hat{y}(\tau+j)$ is chosen, where

$r(\tau + j)$ are the future references. This criterion is represented by the triangular membership function $\mu_e(\hat{e}(\tau + j))$, as given in Fig. 10.5.

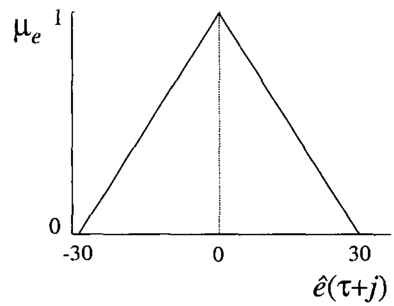


Fig. 10.5 Membership function corresponding to the minimization of the prediction error.

The degrees of satisfaction of the criterion for the H_p steps is combined using Yager's parameterized family of t–norms given in Eq. (9.17). Since it is known that it leads to good control results, the parameter γ is set equal to 2. The lower bound is chosen as $\mu_{J_L}^{(j)} = 0, \forall j$. A rate constraint of $|\Delta u| \leq 0.5$ is imposed on the system in these experiments. The incremental form of the controller is used in the simulations and the interval $[-0.5, 0.5]$ is discretized successively into 11, 21 and 51 equally spaced levels. Hence, $\Delta u(\tau + j - 1)$, with $j = 1, \ldots, H_p$, can take a value only from these discretized control actions.

The branch-and-bound algorithm is compared to the enumerative search for different control horizons. In order to concentrate on the performance of the two optimization schemes, model–plant mismatch and the real-time aspects are not considered in this example. Table 10.1 gives the computational costs in two different metrics,

(i) the computational time (CT), and
(ii) the number of floating-point operations (FLOPS),

for 11 possible control actions at each step. The computational requirements for the B&B method for $H_c = 1$ are taken as 1 (100%). The comparison is made for control horizons from 1 to 4 steps. As Table 10.1 shows, the computational costs of enumerative search are considerably higher than those of B&B. Therefore, branch-and-bound can still be used for larger control horizons, depending on the sampling time of the system under study. Table 10.2 presents the increase of the computational cost with the number of discretizations used for the control actions $u(\tau + j - 1)$, and for $H_c = 2$ using the same normalized scale.

The number of control actions hampers the application of enumerative search

Table 10.1 Comparison of branch-and-bound and enumerative search for different prediction horizons. The numbers are normalized to B&B with $H_c = 1$. Reprinted from (Sousa 2000) by permission of John Wiley and Sons, Inc., ©2000 John Wiley & Sons, Inc.

Control horizon	B&B		Enum. Search	
	CT	FLOPS	CT	FLOPS
$H_c = 1$	1	1	0.98	0.89
$H_c = 2$	2.87	2.95	6.53	12.5
$H_c = 3$	7.77	7.80	54.6	134.8
$H_c = 4$	22.4	21.8	445	1400

Table 10.2 Comparison of branch-and-bound and enumerative search for several numbers of discretizations. The numbers are normalized to B&B with $H_c = 1$, as in Table 10.1. Reprinted from (Sousa 2000) by permission of John Wiley and Sons, Inc., ©2000 John Wiley & Sons, Inc.

Number of discretizations	B&B		Enum. Search	
	CT	FLOPS	CT	FLOPS
11	2.87	2.95	6.53	12.5
21	6.10	6.71	24.7	47.7
51	20.5	25.1	147	292

for a large number of discretizations, while B&B keeps the computational costs at reasonable levels. As the time spent in calculations is dependent on the machine used to control the system, the application of the proposed approach in real-time problems can not be stated as general. However, the application of branch-and-bound clearly allows the application of fuzzy predictive control to systems with smaller sampling times than the use of enumerative search. Figure 10.6 presents the response of the system for $H_c = 2$, $H_p = 2$ and 51 points of discretization for $u(\tau + j - 1)$, and for a reference with various steps.

10.4 Genetic algorithms for optimization in predictive control

When nonlinearity and constraints are present in MBPC, a non-convex optimization problem must usually be solved at each sampling period. The control horizon directly determines the dimension of the optimization problem, which may thus become very complex. Different algorithms, such as sequential quadratic

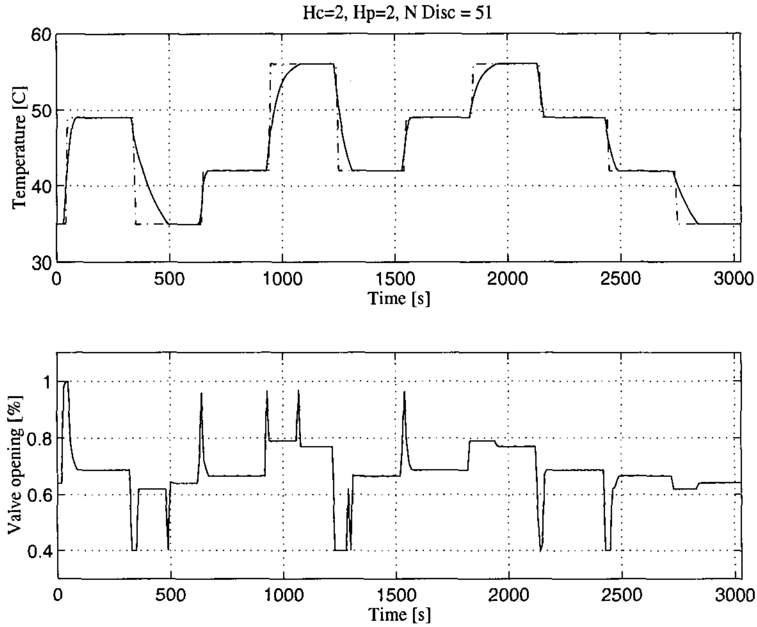


Fig. 10.6 Step responses for the air-conditioning system. Reprinted from (Sousa 2000) by permission of John Wiley and Sons, Inc., ©2000 John Wiley & Sons, Inc.

programming (SQP) or branch-and-bound, can be used to circumvent this problem, as discussed in previous sections. However, SQP usually converges to local minima giving poor solutions, and branch-and-bound, a method that requires a discretization of the control space, requires significant computing power, growing exponentially with the number of possible control actions and with the control horizon. Another possibility is to use a genetic algorithm.

One of the techniques that has proved to be especially suitable for constrained, non-convex optimization problems is evolutionary computing, to which genetic algorithms (GA) belong. Genetic algorithms are optimization methods inspired by the mechanisms of the natural selection and genetics that play a role in the natural evolution of biological organisms. GA have been successfully applied in a variety of fields where optimization in the presence of complicated objective functions and constraints is required (Zurada et al. 1994). The application of GA to model-based predictive control has been addressed by Onnen et al. (1997). Due to the numerical complexity of the GA, they are mostly suitable for processes with slow dynamics, for the time being. However, GA are becoming promising tools for the design of model-based predictive controllers, especially for nonlinear systems, due to their ability to search efficiently in nonlinear, constrained and non-convex optimization problems.

This section investigates the application of GA to the determination of an optimal control sequence in MPBC. A genetic algorithm only unfolds its full capabilities if it is designed properly for a particular application. Therefore, a specific GA must be designed to fulfill the requirements demanded by predictive control, as described in the following. Attention is focused on the application of the proposed method to nonlinear systems with constraints on the process inputs. Advanced genetic operators and other new features are introduced to increase the efficiency of the genetic search. In order to deal with real-time constraints, termination conditions are proposed to abort the evolution, once a defined level of optimality is reached.

10.4.1 *Genetic algorithms*

Genetic algorithms are randomized search algorithms that are based on the mechanics of natural selection and genetics (Goldberg 1989). They combine the principles of natural selection based on 'the survival of the fittest' with a randomized information exchange in order to form a search and optimization algorithm. Although genetic algorithms can be used for a variety of purposes, their most important application is in the field of optimization, because of their ability to search efficiently in large search spaces, which makes them more suitable with respect to the complexity of the optimization problem compared to more conventional optimization techniques.

Since Holland (1975) first proposed the idea of genetic algorithms, many researchers have suggested extensions and variations to the basic genetic algorithm. With the advent of artificial intelligence techniques, many applications of the genetic algorithms have been reported (Davidor 1991), especially in combination with other computational intelligence techniques such as neural networks and fuzzy systems. The importance of genetic algorithms in the field of control is increasing (Linkens and Nyongesa 1995).

10.4.2 *Basic elements of genetic algorithms*

Genetic algorithms code the candidate solutions of an optimization algorithm as a string of characters which are usually binary digits. According to the terminology that is borrowed from the field of genetics, this bit string is usually named a *chromosome*. The solution, which is represented by its chromosome, is called an *individual*. The genetic algorithm considers a number of individuals, which together form a *population*. It modifies and updates the individuals in a population iteratively, searching for good solutions of the optimization problem. Each iteration step is called a *generation*.

The genetic algorithm evaluates the individuals in the population by using a *fitness function*. This function indicates how good a candidate solution is. It can be compared to an objective function in classical optimization. Inspired by the 'survival of the fittest' idea, the genetic algorithms maximize the fitness value, in contrast to classical optimization, where one usually minimizes the objective function. The specification of the fitness function is a very important aspect of the design of genetic algorithms, as the solution of the optimization problem and the performance of the algorithm both depend on this function.

A genetic algorithm evaluates a number of solutions (values) and then generates new solutions for the next step of the iteration, depending on the previous information. The genetic algorithms are distinguished from other numerical optimization methods by the way in which they generate new solutions. Figure 10.7 depicts a schematic representation of a genetic algorithm. The terms in this figure are explained in the sequel.

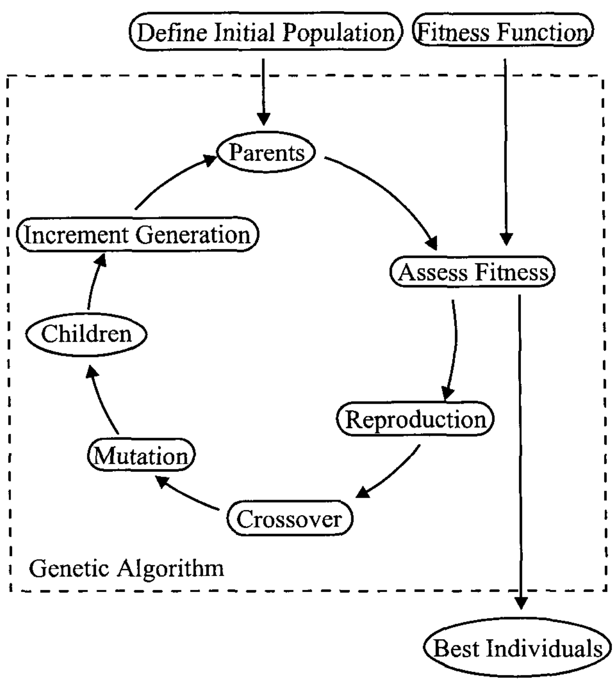


Fig. 10.7 Artificial evolution in genetic algorithms.

The algorithm starts with the generation of an initial population. This population contains individuals which represent initial estimates for the optimization problem. It should be noted that GA evaluate a set of solutions in the population at each iteration step, in contrast to methods like gradient descent, which evaluate a single solution at each iteration step. The fitness of the individuals within the population is assessed, and new individuals (children) are generated for the next generation. The generation is then incremented and children are transformed into parents. A number of genetic operators are available to generate the new individuals.

- **The reproduction or selector operator** chooses chromosomes according to their fitness for mating, *i.e.*, for producing offspring. Fitter individuals get a higher probability of mating, and their genetic material is exploited.
- **Crossover** exchanges genetic material in the form of short allele strings (a part of a chromosome) between the parent chromosomes. This reordering or recombination includes the effects of both exploration and exploitation.
- **Mutation** introduces new genetic material by random changes to explore the search space.

It has been observed that genetic algorithms are valuable optimization tools, especially for non-convex optimization in the presence of constraints. A theoretical understanding of the GA's working principle is provided by the *building block hypothesis* (Goldberg 1989, Michalewicz 1994). Basically, it can be said that a good individual is built up of building blocks of various sizes. The crossover and mutation operators shuffle the elements of the building blocks, searching for even better ones. Since individuals with high fitness can reproduce more, the successful building blocks will have a higher chance of survival across the generations. Thus, the evolution will exploit the available genetic material to explore the search space and accumulate successful genetic material as it continues. As each chromosome includes several building blocks, many more blocks than individuals are processed simultaneously during the evolution. This is one reason for the GA's efficiency in searching complex spaces.

The practical implementation of genetic algorithms requires the selection of a number of operators, as well as the values of various parameters from these operators. The operators that are used most often in the literature are *roulette-wheel reproduction*, *fitness ranking*, *probabilistic and deterministic tournament selection* and *steady-state reproduction* for the reproduction; *multipoint crossover* and *uniform crossover* for the crossover; and *uniform mutation* and *dynamic mutation* for the mutation (Goldberg 1989). The convergence of a genetic algorithm is not uniquely defined, and the evolution can, in principle, continue indefinitely. Therefore, some termination conditions are required to stop the evolution. Usually, it is desired to stop the evolution after a fixed number of generations. Other termination conditions can be used as well, such as the number of generations during which the best individual in the population does not change, or the number of generations during which the highest fitness that is achieved does not change.

10.4.3 *Implementation of constraints*

Most optimization problems are constrained problems, where the set of possible solutions must satisfy various conditions. In addition to the 'hard' constraints that one needs to satisfy, there may be also 'soft' constraints which allow for tradeoffs between constraints. The soft constraints can usually be violated to a degree, provided that this violation leads to improvement in some other part of the optimization goal. Genetic algorithms can handle both types of constraints in a unified manner. Three methods for implementing constraints in a genetic algorithm are

(1) the penalty function method,
(2) the behavioral memory method, and
(3) the domain-specific GA method.

In the penalty function method, the constraints are incorporated in the fitness function, usually as a penalty term. An individual that violates a constraint is thus penalized by reproducing less or not reproducing at all. The behavioral memory method considers a number of constraints in a multiple-step process. Each constraint is considered separately in consecutive evolutions, using a penalty term in the fitness function. The final population of each step, in which a single constraint is considered, is used as the initial population for the next step of the evolution. In this way, the genetic material that proves to be successful when considering a particular constraint in the preceding steps is passed on to the succeeding steps. The domain-specific GA is a genetic algorithm that is designed with a particular application in mind, so that one takes advantage of the additional knowledge about the constraints involved in the problem. Beside the fitness function, the coding method and the genetic operators can be designed specifically for the problem that is being investigated. The domain-specific GA is the most successful way of dealing with constraints in a particular problem, provided that such a GA can indeed be designed. The other two approaches waste genetic material, as the evolution process can lead to many individuals that do not satisfy the constraints. Since these individuals cannot reproduce, the successful building blocks that may be present in their chromosomes disappear from the population. For this reason, this section uses a domain-specific GA for dealing with the constraints. The choice of various design parameters is discussed in the following paragraphs.

10.4.4 *Fitness function*

The first task in the design of GA is to specify the principles for the fitness evaluation. In the approach presented here, the objective function to be minimized is the objective function usually applied in predictive control, given by Eq. (10.1). Only SISO systems are considered for the sake of simplicity. However, this approach can be extended to MIMO systems in a simple way. The objective function for this type of systems is thus given by

$$J = \sum_{j=1}^{H_p} w_{1j}(r(\tau + j) - \hat{y}(\tau + j))^2 + \sum_{j=1}^{H_c} w_{2j}(\Delta u(\tau + j - 1))^2, \qquad (10.13)$$

which accounts for minimizing the variance of the process output from the reference, also minimizing the energy at the same time. The compromise between the two goals is given by the choices of w_{1j} and w_{2j}. As the GA operators are designed to maximize the fitness function, the above minimization problem has to be transformed into a maximization one. This can be done, for instance, by using the transformation

$$f = \frac{1}{1 + J}. \qquad (10.14)$$

This transformation ensures that the fitness values are always positive. Moreover, it scales the fitness values into the interval $[0, 1]$, which can be used to specify conditions for terminating the genetic search prematurely.

10.4.5 *Encoding control variables and implementing constraints*

The next step necessary to apply GA to predictive control is to derive a feasible coding principle that is able to cope with constraints and with the specific characteristics of the variables used in predictive control.

A straightforward coding principle is to represent each change Δu in the control action by one gene, where the sequence of the genes (chromosome) corresponds to the prediction steps. This encoding method automatically implements the rate constraints. This type of constraint is usually applied in industrial processes to avoid sudden changes in the control action, ensuring safety and energy saving. For single-input systems, the number of genes, N_g, is equal to the control horizon H_c. At each step τ, the first gene in a chromosome encodes the change in the control action $\Delta u(\tau)$ to be added to $u(\tau - 1)$. Simple unsigned binary coding has been applied to encode the information. Let L_g denote the number of bits per gene, which is related to the number of digits in a real value. Then, the chromosome length L_c is equal to $L_c = N_g \cdot L_g$.

The encoded values in the form of bit strings have to be decoded to extract the control actions. The strings $00\ldots0$ and $11\ldots1$ correspond to the maximum negative and positive changes in the control action at each step, respectively. The absolute control action is derived by integrating over the time steps. However, this encoding method does not take account of the absolute constraints on the control actions. A commonly used solution to this problem is the use of penalty terms in the objective function. This principle, however, diminishes the efficiency of GA because of the waste of genetic material due to the unfeasible solutions in the population. Another approach is proposed here, which implements both the rate and level constraints in the coding mechanism.

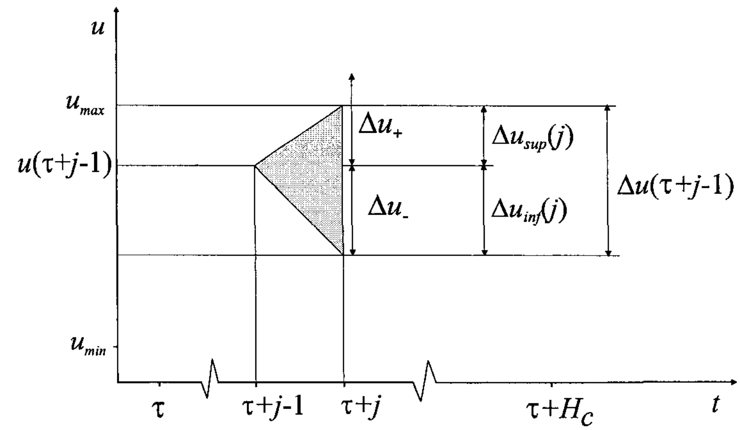


Fig. 10.8 Genes encoding relative change values with the possible values for $u(\tau + j)$.

Let $u_{\max}$ and $u_{\min}$ denote the level constraints, *i.e.*, the maximum and minimum allowed control actions. Similarly, let Δu_+ and Δu_- denote the rate constraints, *i.e.*, the maximum positive and negative changes in the control action, respectively. For a certain control action at time step $\tau + j - 1$, with $1 \leq j \leq H_c$ (corresponding to gene j), the maximum negative and positive changes from the previous control action $u(\tau + j - 2)$ are given by

$$\begin{aligned} \Delta U_{\text{sup}}(j) &= \min[\Delta u_+, u_{\max} - u(\tau + j - 2)], \\ \Delta U_{\text{inf}}(j) &= \min[\Delta u_-, u(\tau + j - 2) - u_{\min}] \,. \end{aligned} \tag{10.15}$$

Figure 10.8 illustrates the principles of this constraint implementation. The maximum positive and negative changes $\Delta U_{\text{inf}}(j)$ and $\Delta U_{\text{sup}}(j)$ are now discretized in n_d values each. The set of possible changes in the control action, $\Delta U(j)$ is given by

$$C_- = \left\{ -\Delta U_{\text{inf}}(j)\frac{n_d - i}{n_d} \middle| i = 0, 1, \ldots, n_d - 1 \right\},$$
$$C_+ = \left\{ \Delta U_{\text{sup}}(j)\frac{n_d - i}{n_d} \middle| i = 0, 1, \ldots, n_d - 1 \right\},$$
$$\Delta u(\tau + j - 1) \in \{C_-, 0, C_+\} . \tag{10.16}$$

These values are coded such that the string $00\ldots0$ corresponds to $\Delta u(\tau + j - 1) = -\Delta U_{\text{inf}}(j)$ and $11\ldots1$ corresponds to $\Delta U_{\text{sup}}(j)$. Note that the genes are encoded sequentially, from the first one corresponding to $\Delta u(\tau)$ to the last one corresponding to $\Delta u(\tau + H_c - 1)$. This procedure assures that all chromosomes encode valid control sequences, avoiding the waste of genetic material.

10.4.6 *Genetic operators*

Genetic operators cannot be optimized independently of each other, and have to be considered as sets of parameters. The set of genetic operators that are used must usually be tailored for the given application. Operators from the literature (Goldberg 1989, Potts et al. 1994) can serve as a reasonable initial setting.

For the application of GA to predictive control, the following operators are used. The reproduction operator is a combination of the deterministic tournament selection and the steady-state reproduction. A pair of individuals is randomly chosen to compete for mating, and the fitter individuals stay in the population. Steady-state reproduction preserves M_{ss} best individuals, and re-introduces them into the population of the next generation. Therefore, the partly optimized chromosomes will not get lost due to disruption of building blocks during crossover. For control applications, a steady-state size of $M_{ss} = 2$ individuals is found to be suitable.

Uniform crossover is used as the crossover operator. It randomly generates a crossover mask to specify which bits are taken from parent 1 (represented by a 1 in the mask) and which are taken from parent 2 (represented by a 0 in the mask) to form the children. Two different offsprings are produced by using the mask and its inverse, as shown in Fig. 10.9. The uniform crossover probability is $P_u = 0.5$, which means that there is an equal probability of having a 1 or a 0 in the crossover mask.

Finally, the uniform mutation operator is used for mutation. This operator inverts each bit of the chromosome with probability P_m ($P_m = 0.01$ in this application), introducing new genetic material into the population.

crossover mask	`11100100`	inverted mask	`00011011`
	`-------`		`-------`
Parent 1	`10100101`	Parent 1	`10100101`
Offspring 1	`10101111`	Offspring 2	`01100001`
Parent 2	`01101011`	Parent 2	`01101011`

Fig. 10.9 The uniform crossover.

10.4.7 *Population structure*

Usually, the population consists of a constant number of individuals throughout the entire evolution. However, the choice of the population size and of the initial population influences the success of GA. First, the size of the population must be chosen. The size of the population should be related to the size of the search space, ensuring a sufficient number of initial search points for the genetic search. The number of individuals (chromosomes) per population N_p, depends on the number of possible control actions $2n_d + 1$, the control horizon H_c (equal to the number of genes per chromosome N_g), the gene length L_g and a constant K_p to be properly chosen

$$N_p = K_p \cdot (N_g + L_g). \tag{10.17}$$

Various tests have been made for the presented application, and $K_p = 8$ is found to be a suitable setting. As an example, if the control horizon is $H_c = 5$ and the gene length is $L_g = 5$, the number of chromosomes per population is thus $N_p = 8 \cdot (5 + 5) = 80$. In general N_p is a tradeoff between a large number of individuals in the population, ensuring sufficient exploration of the search space, and low computational effort.

Secondly, a suitable initial population must be chosen. When no additional information is available, a random initialization is often used. This method guarantees a high genetic diversity, and it is widely used in many research works involving GA. However, with a random initial population, the convergence may be slower than if some initial knowledge about the possible solutions is available, as in the case of MBPC. Predictive control uses the receding horizon principle, which implies that an evolution has to be calculated at each time step. On the one hand, this feature imposes a real-time constraint, but, on the other hand, the past evolutions give important information that can be used to improve the initial population of the current evolution. Two possibilities of using the information from the past evolutions are introduced: the *inter-evolution steady-state* principle and a *learning initial population.*

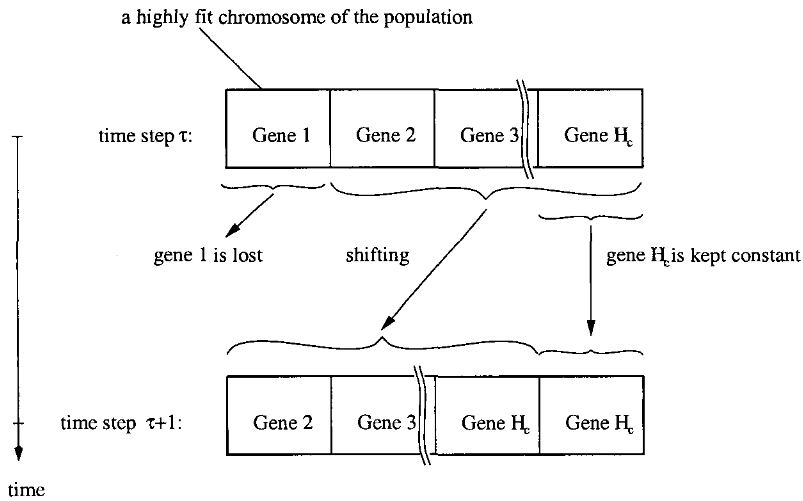


Fig. 10.10 Shift operation with the inter-evolution steady-state principle.

The *inter-evolution steady-state* principle (ISS) preserves optimal solutions of the previous evolution for reintroduction into the initial population of the next evolution. This method keeps the M_{iss} best individuals for the next evolution. The remaining part of the population is randomly initialized. Because genes correspond to time steps, a modification by shifting the genes of the preserved chromosome is applied before reintroducing this chromosome to the next initial population. Thus, at time τ, the genes from $\tau + 2$ to $\tau + H_c$ are shifted one position, and the last gene takes the same value as at $\tau + H_c$, as can be seen in Fig. 10.10. The use of this technique enhances the quality of the population in the first generation because the evolution is likely to start with a highly fit solution, known from the optimization at the previous time step.

It is possible to improve the initial set of solutions by including a *learning initial population*, where 'learning' denotes the existence of a memory for successful solutions. This type of population includes information about already solved problems or repeated situations, containing in memory the most successful individuals of a number of previous evolutions (and not just the best individuals of the last evolution as in the inter-evolution steady-state case). Therefore, the initial population contains a new part where these individuals are stored. In or-

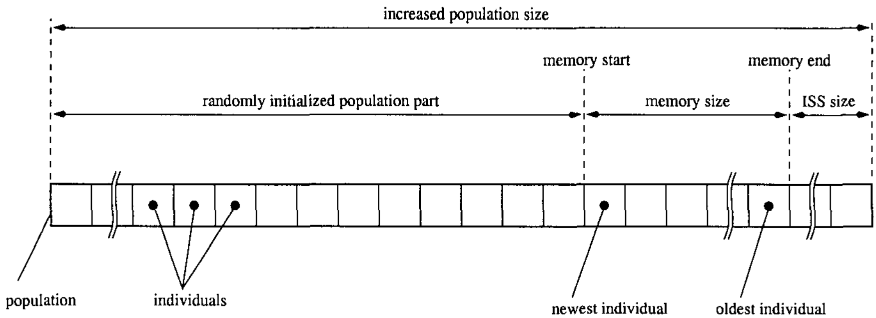


Fig. 10.11 Population structure for improved initial populations.

der to keep genetic diversity, identical solutions are only stored once. At each evolution, a newly learnt individual replaces the oldest one, following the FIFO (first-in, first-out) buffer principle. This memory is copied from one evolution to another, otherwise the stored information would be modified during the genetic operations. The application of this method must be confined to systems where the reference signal includes repeated situations, as in periodic signals. For this type of signal the memory size should be greater than the number of steps to fulfill one period. Figure 10.11 shows the described population structure, also including the principle of inter-evolution steady state.

The use of the techniques described can save up to 20% of the number of generations needed to calculate an acceptable solution, when compared to the situation where only a random initial population is used (Onnen et al. 1997).

10.4.8 *Termination conditions*

Fixing the number of generations per evolution may restrict the genetic algorithm's efficiency. This setting implies that the duration of the genetic search is fixed, regardless of the search success. Moreover, it is difficult to determine beforehand the number of generations needed to find (near)-optimal solutions. Thus, an assessment of the quality level of the genetic algorithm should be made on-line. Three different approaches have been investigated to provide the conditions to abort the evolution.

- **Absolute fitness limit** - the genetic search stops when the highest fitness in the population reaches a predefined value. This method can only be used if the possible maximum fitness or a desired fitness is exactly known, which is the case when using Eq. (10.14).

- **Convergence rate** - uses a condition on the rate of convergence of the maximum fitness over the entire evolution process to abort the search. If the maximum fitness is unchanged for a given number of generations N_w, the evolution stops.
- **Convergence rate of the first gene** - this condition is quite similar to the previous one. The genetic search stops once the first gene is unchanged for N_{fw} generations. Note that the first gene represents the current control input $u(\tau)$.

Here, the convergence rate of the first gene is used. Experimental results show that in 90% of the test runs the first gene stops changing earlier than the other genes of a chromosome. Note that the rest of the optimized control sequence does not affect the control quality.

In summary, the genetic algorithm designed to cope with the specific requirements of predictive control has the following characteristics.

- Termination conditions to abort the evolution are proposed in order to cope with the real-time requirements of MBPC, after a specified level of optimality is achieved.
- A coding scheme is developed to implement level and rate constraints on the controlled process inputs. This scheme also provides a way to efficiently encode optimization variables by not wasting genetic material.
- A method of initializing the population is suggested, which introduces a specified number of best solutions from the previous time step to the new population. Following the receding horizon principle, the genes of the previous solution are shifted by one step.
- A learning feature is introduced, that stores successful individuals over a given time period, in order to cope more effectively with periodic reference signals.

10.5 Application example with genetic algorithms

This section considers the pressure dynamics of a simulated batch fermenter as an example of the application of genetic algorithms in MBPC. The simulation results with GA are compared to those obtained with the branch-and-bound method, in terms of the achieved control accuracy and the computational costs.

The predictive control scheme based on GA optimization is applied in the simulation of pressure control in a laboratory fermenter. This system is described in Sec. 7.6. The nonlinear differential equation given by Eq. (7.45), and reminded here, is used for the simulation model of the system.

$$\frac{dP}{dt} = \frac{1000.RT}{22,4.V_h} \cdot \left[\Phi_g - (\pi R_H^2) \sqrt{\frac{2P_0}{\rho_0 K_f} ln(\frac{P}{P_0})} \right] .$$

The symbols and respective values are described in Sec. 7.6. The maximum changes in the valve position are $\Delta u_+ = \Delta u_- = 10\%$ of the total range per sample and the level constraints are $u_{\min} = 0\%$ and $u_{\max} = 100\%$ of the valve position. The control and prediction horizons are chosen equal, $H_p = H_c$. The discretization n_d of the control universe is set to 2, thus providing 5 possible changes in the control action. For the sake of simplicity, only the error criterion is considered in Eq. (10.13), and thus $w_{1j} = 1$ and $w_{2j} = 0$, for $j = 1, \ldots, H_p$. Figure 10.12 shows an example of a typical simulation experiment.

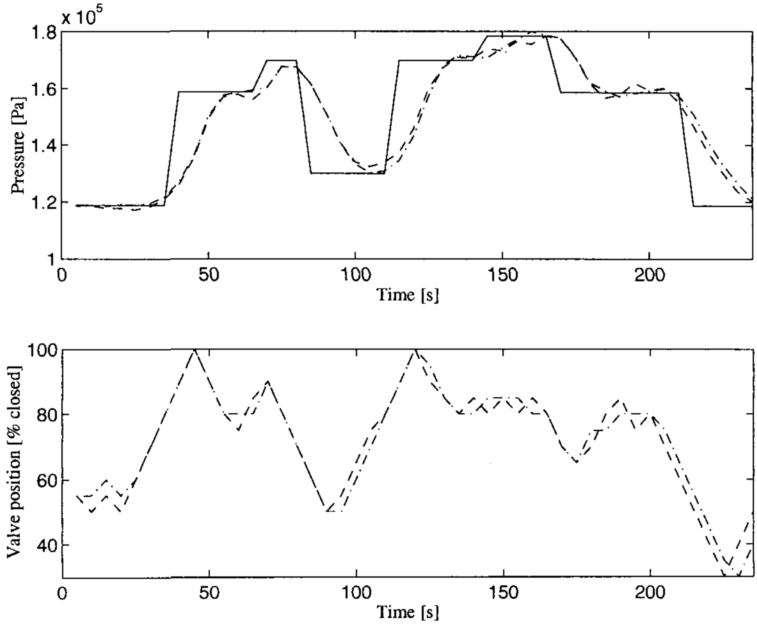


Fig. 10.12 Example of time responses for a given reference with $H_p = 4$ (solid line: reference, dashed-dashed line: GA, and dashed-dotted line: branch-and-bound.

The performance of the GA is compared to a branch-and-bound method from Sec. 10.2. The branch-and-bound uses five equal discretization intervals, contrary to the GA discretization method, which uses different discretization intervals, see Sec. 10.4. Three comparison criteria are considered.

(1) CPU time (in seconds).
(2) FLOPS (floating-point operations).
(3) Control accuracy (sum-squared error).

Figure 10.13 shows that the computational effort as a function of H_c increases much faster for the branch-and-bound technique than for the GA. For the application presented, the GA outperforms the branch-and-bound method for $H_c > 6$. For shorter horizons, the GA needs more computational effort.

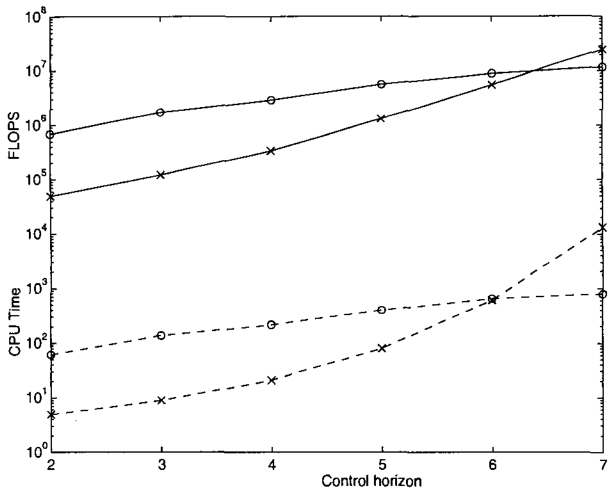


Fig. 10.13 Comparison of the computational costs in terms of CPU time (- -) and FLOPS (—) for the GA (o) and the B&B method (×).

Figure 10.14 compares the two optimization methods in terms of sum-squared error (SSE). The control accuracy is similar, except for the control horizon of 2 steps, where the GA performs considerably worse.

These experimental results thus show that GA outperforms the branch-and-bound method for longer control horizons (above 6, with the process considered here) in terms of computational costs. The control accuracy achieved is comparable to the global optimum found by the branch-and-bound method. Therefore, GA optimization for MBPC can best be applied to processes with relatively slow dynamics and long control horizons. Note that the computational costs for MIMO systems increase linearly for GA, while they increase exponentially for the B&B method. Therefore, it is expected that GAs will outperform B&B for multivariable systems.

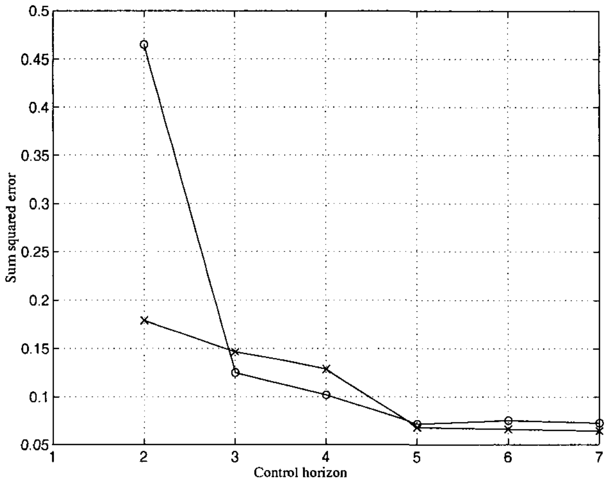


Fig. 10.14 Comparison of the sum-squared error calculated over the complete reference: for the GA (o) and the B&B method (×).

10.6 Summary and concluding remarks

Different optimization algorithms to be applied in fuzzy predictive control have been considered in this chapter. Two optimization techniques

(i) branch-and-bound method,
(ii) genetic algorithms,

that can be used to deal with the problems in predictive control, using classical objective functions, when the model of the process is nonlinear, are presented and compared. The branch-and-bound technique has been extended to predictive control with fuzzy objective functions.

The branch-and-bound algorithm is applied to MBPC. This algorithm is faster and more accurate than *enumerative search* and the SQP method. However, the computational effort increases exponentially with the control horizon and the number of discrete control alternatives. Therefore, these two parameters must be chosen such that they constitute a good compromise between computational time and desired accuracy. Another important advantage of branch-and-bound is that it implicitly deals with constraints. Moreover, the presence of constraints improves the efficiency of the method by eliminating nodes that do not contain the optimal solution.

An extension of the branch-and-bound algorithm for predictive control with fuzzy objective functions, which is derived from the B&B algorithm for classical MBPC is considered. In order to apply this B&B algorithm, the fuzzy goals and the fuzzy constraints must be combined by using a t–norm. The computational costs are reduced considerably in comparison to a full enumerative search. Therefore, this approach can be applied in real-time for processes with relatively large sampling periods. A real-time application of this B&B algorithm to an air-conditioning system is presented in Sec. 12.4.4.

Genetic algorithms can also be applied when a non-convex optimization problem must be solved in the presence of constraints. GA have been applied to model-based predictive control. Some special characteristics for the GA required for this application are considered. Termination conditions, a coding scheme to implement level and rate constraints, a method for initializing the population, and the introduction of a learning feature are proposed to cope with the optimization in MBPC. The performance of the designed GA has been compared to the branch-and-bound method. GA outperforms the branch-and-bound method for long control horizons in terms of computational costs. The control accuracy achieved is comparable to the branch-and-bound method.

Chapter 11

Advanced Optimization Issues

When the optimization problem in MBPC is non-convex, general search methods can be used to find the solution to the optimization problem at every time step, as discussed in Chapter 10. These general search methods are very demanding computationally, but they are unavoidable when the optimization problem to be solved is complex and non-convex. Unfortunately, the optimization problem in fuzzy MBPC is in general non-convex. However, under specific circumstances, the optimization can be formulated as a convex problem. In that case, efficient optimization methods, such as interior point methods, can be used to solve the convex optimization problem in polynomial time. It is thus important to determine under which circumstances the optimization in fuzzy MBPC remains convex, so that time consuming general search methods can be replaced by fast convex optimization methods for those cases. Some specific circumstances under which the optimization problem in fuzzy predictive control remains convex are discussed in this chapter.

One of the problems encountered when discrete search methods are used to solve the optimization problem (because it is non-convex due to non-convex constraint set and/or complicated objective functions) is that the computational complexity increases exponentially with the number of discrete control alternatives. The discretization of the control space should be such that the control accuracy is sufficiently high, while the computational complexity is low. One solution to this problem is to use fuzzy predictive filters that adapt the discretization of the control space in order to avoid limit cycles in steady-state, while keeping the number of discrete control alternatives, *i.e.*, the number of discrete control actions, small.

The outline of the chapter is as follows. Convex optimization in fuzzy predictive control is discussed in Sec. 11.1. Some particular conditions under which the optimization problem remains convex are discussed. An example of convex optimization in fuzzy predictive control is given in Sec. 11.2. The trade-off between the accuracy and the computational complexity in discrete search methods is considered in Sec. 11.3. The concept of fuzzy predictive filters to solve the problems introduced by the discretization of the control space in branch-and-bound or genetic algorithms is discussed in this section. The concept of a fuzzy predictive filter is illustrated in Sec. 11.4 with an example, before the concluding remarks in Sec. 11.5.

11.1 Convex optimization in fuzzy predictive control

In model predictive control with fuzzy objective functions, various forms of aggregation for the several criteria can be chosen giving greater flexibility for expressing the control goals. However, usually these aggregation operators result in a non-convex optimization which is computationally not tractable with conventional optimization techniques. As it is often not possible to find a global optimum in non-convex optimization, and as it requires large computational effort, usually fuzzy predictive control can only be applied to systems with slow dynamics, where the sampling time is long enough to perform the complicated optimization step. It is desirable to have a convex optimization problem for finding the global optimum in relatively short time, so that the method can be applied to a large class of systems. This section shows that under certain conditions the optimization in fuzzy predictive control is convex. This work was introduced in (Sousa, Kaymak, Verhaegen and Verbruggen 1996). The fuzzy predictive control scheme with the convex optimization problem is applied in Sec. 11.2 to the control of a simulated non-minimum phase, unstable linear system to illustrate the applicability of the scheme.

Fuzzy predictive control should find the best control actions maximizing the membership function μ_π as described in Sec. 9.2.2. Transforming the maximization into a minimization problem defines the optimization problem in a more classical way. Using this transformation, particular membership functions and the Yager t–norm, fuzzy criteria can be aggregated in a way that leads to a convex optimization problem.

The general form of a nonlinear constrained optimization problem is defined in Eq. (8.2). The fuzzy multicriteria decision making problem is defined as an unconstrained problem for the formulation presented in Eq. (8.2), because the general goal function is defined as a confluence of fuzzy goals *and* fuzzy constraints. Therefore, the necessary and sufficient conditions for the fuzzy optimization problem, described by the optimal policy of Eq. (9.3) in Sec. 9.2.2 to be a convex programming problem, is that the function $J(\mathbf{v})$ in Eq. (8.2) is a convex function. For convex programming problems, any local minimum $\mathbf{v}^*$ is the global minimum, considerably reducing the computational effort. As the optimization in fuzzy predictive control must be done on-line, it is important to determine the conditions under which the selection of the fuzzy decision parameters results in a convex optimization.

Assume that the system under control is a linear SISO system, and is described by the auto-regressive linear model

$$y(\tau+1) = \sum_{j=1}^{p_y} a_j\, y(\tau - j + 1) + \sum_{j=1}^{m_u} b_j\, u(\tau - j + 1)\,, \tag{11.1}$$

where $y(\tau), \ldots, y(\tau - p_y + 1)$ and $u(\tau), \ldots, u(\tau - m_u + 1)$ are the shifted model outputs and inputs, respectively, and p_y, m_u are the integers related to the model order.

The minimization of the predicted output error between the reference and the predicted model output over the entire prediction horizon is the only goal considered. This goal is represented by a membership function at each time step. Thus, the optimization criterion is represented at each time step by a symmetric triangular membership function, which is defined around zero output error, *i.e.*,

$$\mu(r(\tau+j) - \hat{y}(\tau+j)) = \max\left(1 - \frac{|r(\tau+j) - \hat{y}(\tau+j)|}{K_e}, 0\right)\,, \tag{11.2}$$

where $r(\tau + j)$ is the reference, $\hat{y}(\tau + j)$ is the predicted model output, $j = 1, \ldots, H_p$, and K_e is the spread of the membership function, as discussed in Sec. 9.3.2. This spread depends on the problem and it should be selected such that the intersection of fuzzy goals over the prediction horizon is not empty. In practical terms, this means that μ_π should not be zero over the whole optimization space. This can be achieved by selecting a particular range within which the variables may vary and then by choosing the spread accordingly so that μ_π does not become zero (assuming that a feasible solution exists within the constrained set). For the system of Eq. (11.1) under study, any value that does not lead to an empty intersection of fuzzy goals defined for each time step in the prediction horizon can be chosen, because the global optimum remains the same.

Note that in addition to the goal of minimizing the error, it is also possible to define crisp constraints in the optimization space $u(\tau) \times \cdots \times u(\tau + H_c - 1)$, provided that they form a convex set. These (convex) constraints can be represented by membership functions for crisp sets that are defined on the appropriate universe of discourse. Fuzzy constraints on the optimization space are not considered here, although it can be expected that the results can be generalized to the case with fuzzy constraints by using the resolution principle of fuzzy sets (Klir and Yuan 1995). The Yager t–norm, given, *e.g.*in Eq. (9.6), is used as the decision function for combining the decision criteria over the prediction horizon. Under these conditions, the following theorem can be formulated.

Theorem 11.1 Convex optimization in fuzzy predictive control. *Let the system to be controlled be described by Eq. (11.1), i.e.the system is linear, and let the goal of the optimization be the minimization of the prediction error over the prediction horizon, where the membership functions for the fuzzy goals at each step are given by Eq. (11.2). Further, let the constraints on the optimization space* $u(\tau) \times \cdots \times u(\tau + H_c - 1)$ *be crisp (not fuzzy) and convex, and finally let the Yager t–norm in Eq. (9.6) be used for the aggregation of the criteria. Under these conditions, the optimization problem at each step of fuzzy predictive control is convex, provided that a feasible solution exists in the optimization space.*

Proof. Equation (11.1) can be rewritten as an affine function of u for the predicted outputs and for a particular point j in the prediction horizon

$$\hat{y}(\tau + j) = y_0 + \sum_{l=1}^{j} \alpha_l u(\tau + l - 1) \tag{11.3}$$

with $j = 1, \ldots, H_p$. The variable y_0 (constant at time τ for a particular value of j) depends on the parameters a_j, b_j, on the output values $y(\tau), \ldots, y(\tau - p_y + 1)$ and on the control actions $u(k), \ldots, u(k - m_u + 1)$. The values for α_l are related to the parameters of the impulse response and can be derived from a_j and b_j. The error is given by $\hat{e}(\tau + j) = r(\tau + j) - \hat{y}(\tau + j)$ and it can be written also as an affine function of u,

$$\hat{e}(\tau + j) = e_0 + \sum_{l=1}^{j} \alpha_l' u(\tau + l - 1) \tag{11.4}$$

with $e_0 = r(\tau + j) - y_0$ and $\alpha_l' = -\alpha_l$. Considering that the error remains inside the universe of discourse defined, the membership function for the error defined in Eq. (11.2) can be described by

$$\mu(\hat{e}(\tau + j)) = 1 - \frac{|\hat{e}(\tau + j)|}{K_e}. \tag{11.5}$$

Substituting Eq. (11.4) in Eq. (11.5) one obtains

$$\mu(\hat{e}(\tau + j)) = 1 - \frac{|e_0 + \sum_{l=1}^{j} \alpha_l' u(\tau + l - 1)|}{K_e}. \tag{11.6}$$

Depending on whether the error is positive or negative, Eq. (11.6) can be written in an affine form,

$$\mu(\hat{e}(\tau + j)) = e_0' + \sum_{l=1}^{j} \beta_l u(\tau + l - 1), \tag{11.7}$$

where for $e(\tau + j) > 0$, $e'_0 = 1 - \frac{e_0}{K_e}$ and $\beta_l = -\alpha'_l$, and for $e(\tau + j) < 0$, $e'_0 = 1 + \frac{e_0}{K_e}$ and $\beta_l = \alpha'_l$.

Since the optimal policy is found by a maximization in fuzzy decision making, the term

$$\left\{ \sum_{j=1}^{H_p} (1 - \mu_j(\hat{e}(\tau + j)))^\gamma \right\}^{1/\gamma} \tag{11.8}$$

should be minimized. This is a γ–norm of $1 - \mu_j(\hat{e}(\tau + j))$. Since

$$1 - \mu_j(\hat{e}(\tau + j)) = \frac{|\hat{e}(\tau + j)|}{K_e}$$

is an affine function of u according to Eq. (11.6), Eq. (11.8) is a γ–norm of an affine function. It is known that the minimization of the norm of an affine function with convex constraints on the optimization space results in a convex optimization problem (Boyd and Barret 1991). Thus, the maximization of the Yager t–norm with the given membership functions and the possible convex crisp constraints results in convex optimization. □

The controller design problem stated above is a convex programming problem, where any local minimum x^* is the global minimum, and therefore it can be efficiently solved. Effective algorithms exist for solving a convex optimization problem, where the growth of computational effort with the number of variables and criteria has been observed to be quite moderate. The *descent methods* form a large family of algorithms usually applied in convex optimization. They produce solutions that have decreasing objective values in successive iterations. Usually, these methods require the computation of a descent direction for the function at a point, that can be a difficult task in itself. Another possibility is to use cutting-plane or ellipsoid methods. These methods are described, *e.g.* by Boyd and Barret (1991). They have simple stopping criteria guaranteeing that the optimum has been found to a given accuracy. However, for smooth problems (like the problem under study, as can be seen in the example given in Sec. 11.2), many of the descent methods present faster convergence. From the methods that use gradient information, the most favored are the quasi-Newton methods. These methods build up curvature information at each iteration to formulate a quadratic model problem. The main difference between various quasi-Newton methods consists of the different ways of computing the update of the Hessian matrix in the quadratic formulation.

11.2 Application example with convex fuzzy optimization

Convex fuzzy optimization that is described in Sec. 11.1 is applied to the control of the simulated non-minimum phase, open-loop unstable linear system, presented previously in Sec. 9.4.1. This system is described by Eq. (9.14), which is repeated here as a reminder,

$$G(s) = \frac{s-1}{s^3+s^2+s+2}.$$

The sampling time is 1 s. The prediction horizon H_p is chosen as 6 (related to the settling time) and the control horizon H_c as 2. Both of these are values that give good step responses for this system. A crisp constraint on the rate of the control action is defined as $|\Delta u| \leq 0.5$. In order to have a convex optimization problem, the function defining this constraint must be convex. It is assumed that for this simple example the process model is equal to the plant. Step responses with several values of the Yager parameter γ have been studied. The results are shown in Fig. 11.1.

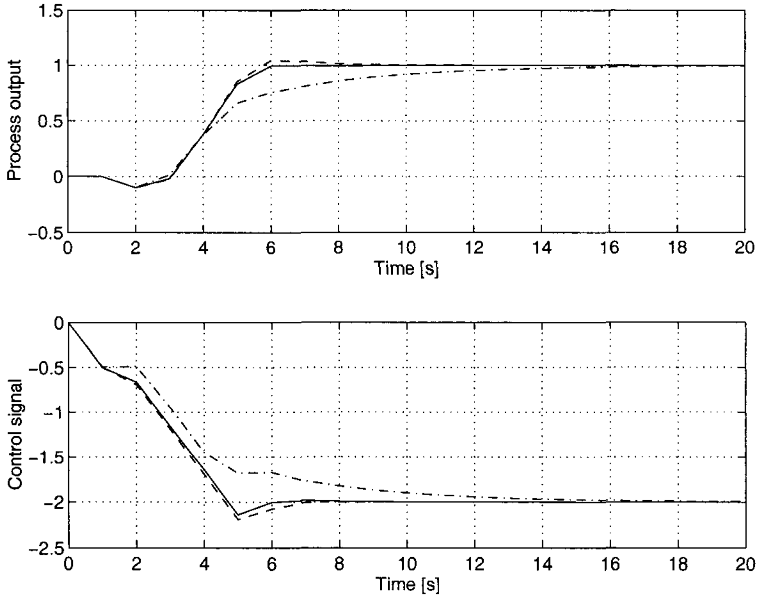


Fig. 11.1 Response of a fuzzy predictive controller using Yager t–norm as the decision function. Dashed: $\gamma = 1.9$, solid: $\gamma = 2.8$, dash–dotted: $\gamma = 4$. Reproduced from (Sousa, Kaymak, Verhaegen and Verbruggen 1996), ©1996 IEEE.

A convex programming technique is used for the optimization at each step for

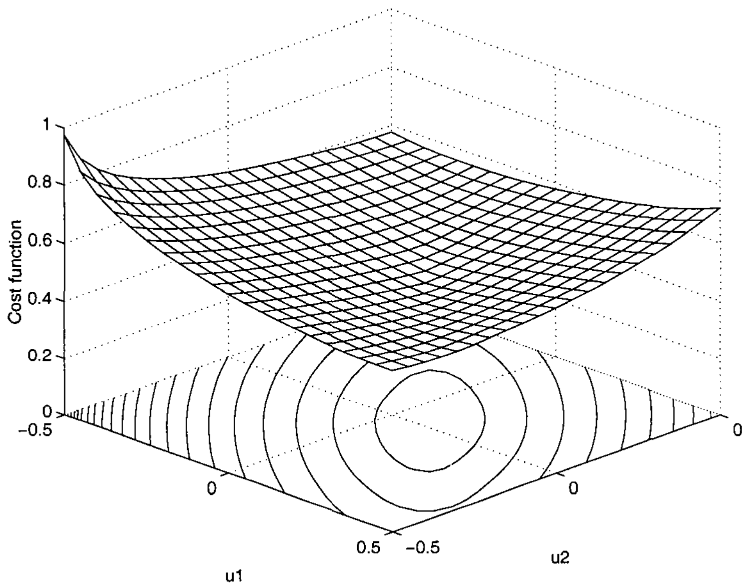


Fig. 11.2 Example of the cost function defined in Eq. (11.8) at time step $\tau = 7$ ($t = 7$s) with $\gamma = 2.8$ for the control actions $u1 = u(\tau + 1)$ and $u2 = u(\tau + 2)$. Reproduced from (Sousa, Kaymak, Verhaegen and Verbruggen 1996), ©1996 IEEE.

several values of γ. In this example, a gradient descent method is utilized, providing fast convergence as required. The gradient-based method using the updating method of the Hessian matrix as described in Fletcher (1970) is applied in this example. An example of a surface resulting from the optimization of Eq. (11.8) is plotted in Fig. 11.2 for the time step $\tau = 7$ and for $\gamma = 2.8$. This figure represents the resulting surface as a function of the first two control actions $u(\tau + 1)$ and $u(\tau + 2)$ (the control horizon in this example). Notice that it is a convex surface as expected. Lines of constant cost (contour lines) are also plotted in the same figure in order to demonstrate the convex nature of the optimization problem.

Hence, the optimization problem in fuzzy predictive control is convex when the controlled system is linear, the goal is the minimization of the predicted error over the prediction horizon, and the decision criteria are combined using the Yager t–norm as the decision function. This means that efficient convex optimization techniques can be used for fuzzy predictive control of linear systems. Processes with relatively fast dynamics can also be controlled since the convex optimization techniques demand less computational effort than the non-convex optimization techniques that have been used for fuzzy predictive control in Chapter 10. Therefore, the advantages of fuzzy predictive control can, in particular cases, be combined with the advantages of convex optimization, for which efficient computational algorithms are available.

11.3 Fuzzy predictive filters

When the optimization problem in MBPC is non-convex, it can be addressed by discrete search techniques, such as branch-and-bound or genetic algorithms, as described in Chapter 10. The discretization of the control space, however, introduces a trade-off between the accuracy, related to the number of discrete alternatives (search space), and the computational complexity. Additional problems introduced by the discretization are oscillations around non varying references (chattering) and slow step responses. Fuzzy predictive filters have been introduced to help this problem (Sousa and Setnes 1999), and it has been applied in real-time to a robotic manipulator (Baptista et al. 2001). In these systems, a fuzzy filter scales the gain of an adaptive set of possible control actions by using simple fuzzy criteria considering the current state of the system and the predicted error. Note that besides the name, the proposed filter is different from other fuzzy adaptive filters reported in literature (Wang and Mendel 1993, Plataniotis et al. 1996), as it is used to derive a set of feasible alternatives for the discrete optimization algorithm. In the proposed approach, the search space for the optimization is kept limited, while the performance of the controller is increased.

11.3.1 *Basic principles*

A *fuzzy predictive filter* is composed of an adaptive set of incremental control alternatives and a rule base of simple fuzzy prediction rules for scaling these alternatives. The predictive rules consider the error between the system's output and the desired reference in order to infer a scaling factor, or gain, $\gamma(\tau) \in [0, 1]$ for the discrete incremental control actions. Basically, the gain is decreased when the system is close to a steady state situation, *i.e.*, the error and the change in error are both small, and increased if the error is big or the output moves away from the reference. As a result of this, when the system is close to a steady-state operation, the gain is small and the possible control actions are all close to each other, diminishing to a great extent the variation of the output (chattering). On the other hand, when the reference contains sudden changes, the gain is increased and the control actions can vary much more, allowing for a fast response of the system under control. The fuzzy predictive filter reduces the accuracy problem introduced by the discretization of the control actions, while at the same time the number of necessary control alternatives is kept low, thereby speeding up the optimization. An illustration of the fuzzy predictive filter is given in Fig. 11.3. The design consist of two parts: the choice of the adaptive discrete control alternatives, and the construction of the fuzzy rules for the gain filter.

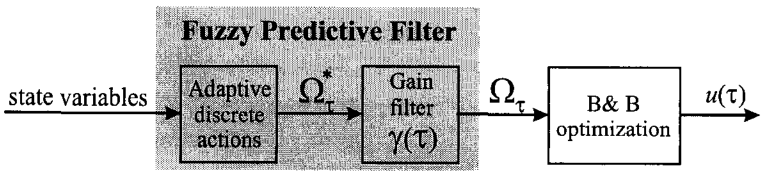


Fig. 11.3 Fuzzy predictive filter, where the state variables are the current control action, the predicted error and the change in error. Reproduced from (Sousa and Setnes 1999),©1999 IEEE.

11.3.2 *Adaptive control alternatives*

Instead of using a fixed set of incremental control alternatives, $\Omega = \{\omega_i | i = 1, 2, \ldots, n_d\}$, as shown in Eq. (10.4), the fuzzy predictive filter makes use of an adaptive set of control alternatives. Let $u(\tau - 1) \in U$ represent the control action at time instance $\tau - 1$, where $U = [U^-, U^+]$ is the domain of the manipulated variable. The upper and lower bounds of the possible change in the control signal at time τ, u_τ^+ and u_τ^-, respectively, are given by

$$\begin{aligned} u_\tau^+ &= U^+ - u(\tau - 1)\,, \\ u_\tau^- &= U^- - u(\tau - 1)\,. \end{aligned} \tag{11.9}$$

The values $u_\tau^+$and u_τ^- are thus the maximum changes allowed for the control action when it is increased or decreased, respectively. The adaptive set of incremental control alternatives are now defined as

$$\Omega_\tau^* = \{0, \lambda_l\, u_\tau^+, \lambda_l\, u_\tau^-\, | l = 1, 2, \ldots, N\}\,, \tag{11.10}$$

where the designer has to choose the distribution λ_l instead of choosing fixed alternatives as in Eq. (10.4). The choice of λ_l sets the maximum change allowed at each time instant by scaling the maximum variations u_τ^+ and u_τ^-, and the l parameter determines the number of possible control actions. The values of λ_l can be such that the control alternatives are, *e.g.*linear or logarithmically distributed. For example, λ_l can be selected such that

$$\lambda_l = \frac{1}{4^{l-1}}, \quad l = 1, 2, 3\,. \tag{11.11}$$

Hence, the following seven changes in the control actions can be applied, in this case, at each sampling period

$$\Omega_\tau^* = \left\{ u_\tau^-, \frac{u_\tau^-}{4}, \frac{u_\tau^-}{16}, 0, \frac{u_\tau^+}{16}, \frac{u_\tau^+}{4}, u_\tau^+ \right\} . \tag{11.12}$$

From Eq. (11.10) it follows that the cardinality of Ω_τ, *i.e.*, the number of discrete control alternatives, is given by $n_d = 2l + 1$ including the zero element. For fast processes, *i.e.*, processes with small sampling periods, l must be small. In drastic cases it can be reduced to one, and only three changes in the control actions are allowed: reduce, maintain or increase the current control.

The variation of the control action given by Ω_τ^* in Eq. (11.10) may for some situations be too drastic, and may yield an undesirable behavior of the system, such as overshoots or oscillations. The proper choice of the maximum changes and the values λ_l are process dependent. However, the use of the predictive gain filter makes these choices far less critical than in the case of fixed alternatives, as it scales the gain of these actions depending on the predicted deviation of the output from the reference signal.

11.3.3 *Gain filter*

The fuzzy predictive filter applies a scaling factor, or gain, $\gamma(\tau) \in [0,1]$, to the adaptive set of control actions Ω_τ^* in order to obtain a scaled version Ω_τ that is presented to the optimization routine,

$$\Omega_\tau = \gamma(\tau) \cdot \Omega_\tau^* . \tag{11.13}$$

The scaling factor $\gamma(\tau)$ can be defined in different ways. When the system under control is at steady state, the fuzzy predictive filter should scale down the control alternatives to enable convergence to the (non-discrete) optimal value and eliminate the chattering effect. On the other hand, when big changes are predicted, the gain should be high to enable a fast response. Thus, the factor $\gamma(\tau)$ must be chosen based on at least the predicted error between the reference and the system's output. The predicted error is defined as

$$\hat{e}(\tau + H_p) = r(\tau + H_p) - \hat{y}(\tau + H_p), \tag{11.14}$$

where $r(\tau + H_p)$ is the reference to be followed at time $\tau + H_p$. Further, the change in the error gives an indication on the evolution of the system, and this information should also be considered in the derivation of $\gamma(\tau)$. The change in error is defined as

$$\Delta e(\tau) = e(\tau) - e(\tau - 1). \tag{11.15}$$

Note that the difference operator can amplify high frequency noise if present. When the error signal $e(\tau)$ is corrupted by significant noise, it should be filtered before the change in error is computed, *i.e.*, $\Delta e(\tau) = e_f(\tau) - e_f(\tau - 1)$, where e_f denotes the filtered error.

Considering the predicted error and the change in the error, simple heuristic rules can be constructed for the gain. When both $\hat{e}(\tau + H_p)$ and $\Delta e(\tau)$ are small, the system is close to a steady state situation. The set of control alternatives should then be scaled down to allow finer control actions, *i.e.*, $\gamma(\tau) \rightarrow 0$, in order to approach zero steady state error without introducing oscillations around the set-point. When the predicted error and the change in error are both high, bigger corrective steps should be taken, *i.e.*, $\gamma(\tau) \rightarrow 1$. The two fuzzy criteria, 'small predicted error' and 'small change in error', are defined by the membership functions $\mu_e(\hat{e}(\tau + H_p))$ and $\mu_{\Delta e}(\Delta e(\tau))$, respectively. These two criteria can be aggregated by means of a conjunction that can be represented by the minimum operator. The aggregation of these criteria is then given by

$$\mu_\gamma(\hat{e}(\tau + H_p), \Delta e(\tau)) = \min(\mu_e, \mu_{\Delta e})\,. \tag{11.16}$$

Note that γ is the complement of the aggregated membership μ_γ. Thus, the scaling factor $\gamma(\tau)$ can be easily derived by taking the fuzzy complement of μ_γ as

$$\gamma(\tau) = \overline{\mu_\gamma} = 1 - \mu_\gamma\,. \tag{11.17}$$

From the definition of μ_γ in Eq. (11.16) and the definition of $\gamma(\tau)$ in Eq. (11.17) it follows that when one of the two variables, error or change in error, is not small the gain $\gamma(\tau)$ is increased. Only when both conditions are fulfilled, *i.e.*, both error and change in error are small, the gain is decreased. This property is given by the complement of the minimum of the two membership functions in Eq. (11.16).

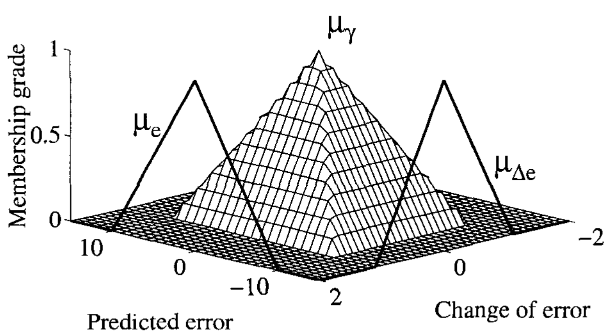


Fig. 11.4 Two-dimensional membership function $\mu_\gamma = \min(\mu_e, \mu_{\Delta e})$. Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

In summary, Eq. (11.10) represents an adaptive set Ω_τ^* of incremental control actions at time instance τ. These are determined by the available control space at time τ, as defined in Eq. (11.9). The actions are scaled by the gain $\gamma(\tau) \in [0, 1]$ to create a set of alternatives Ω_τ that are passed on to the optimization routine. The value of $\gamma(\tau)$ is determined by simple fuzzy criteria regarding the error state of the system. The proposed fuzzy predictive filter has only a few design parameters; namely λ_l, and the membership functions μ_e and $\mu_{\Delta e}$, all of which are not really critical, and thus allows for the use of some heuristics to cope with uncertainties in their definitions. Possible constraints on the control signal concerning, *e.g.*the range and rate of change of the control variable can be implemented by properly selecting the parameters λ_l.

11.4 Application example for fuzzy predictive filters

A HVAC system (see Sec. 9.4 and Sec. 12.2) is used as an example to test the fuzzy predictive filter proposed in Sec. 11.3. The system consists of a fan-coil unit inside a test cell (room) under control. The system should keep the temperature of the room at a certain reference value, ensuring that enough ventilation and renovated air are supplied to the room.

A simplified Takagi–Sugeno fuzzy model is constructed from process measurements using 800 samples with a sampling period of 30 s. The model predicts the supply air temperature T_s based on its present value, the mixed air temperature T_m, and the heating valve position u, thus

$$\mathbf{x}(\tau) = [T_s(\tau), u(\tau - 1), T_m(\tau)]^T. \tag{11.18}$$

The rules have the following form,

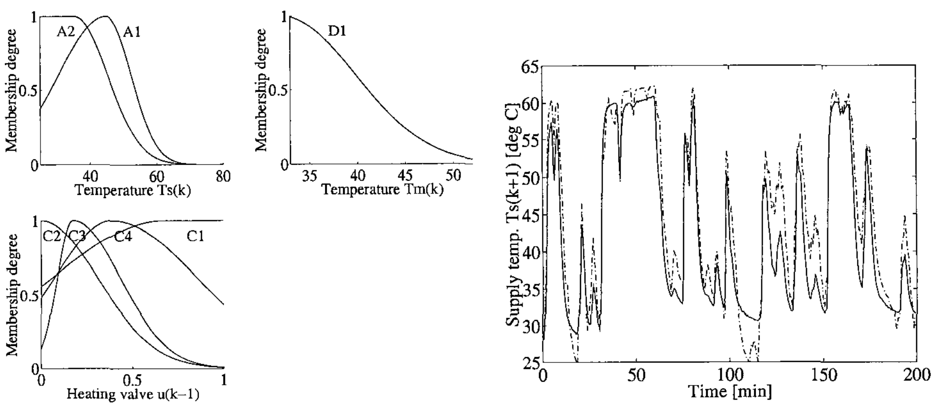
$$\begin{aligned} R^k :\ & \textbf{If } T_s(\tau) \text{ is } A_1^k \text{ and } u(\tau-1) \text{ is } A_2^k \text{ and } T_m(\tau) \text{ is } A_3^k \\ & \textbf{Then } T_s(\tau+1) = y^k\,, \quad k = 1, \ldots, 10\,, \end{aligned}$$

where $y^k = \theta_k^T [\mathbf{x}(\tau)^T 1]^T$. The final model is shown in Table 11.1, and it is simplified by using model simplification techniques presented by Setnes, Babuška, Kaymak and van Nauta Lemke (1998). The antecedent membership functions of the model are shown in Fig. 11.5, together with the validation in a free-run simulation using unseen data.

The model is implemented in the internal model control scheme (see Sec. B.2) as depicted in Fig. 11.6, and applied to the control of the fan-coil unit with a step-like reference. Both the error signal $e_m(\tau)$ and the mixed air temperature measurement $T_m(\tau)$ are passed through first-order low-pass digital Butterworth

Table 11.1 Fuzzy model of a HVAC system. Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

$T_s(\tau)$	$u(\tau-1)$	$T_m(\tau)$	$T_s(\tau+1)$
-	C_1	-	$-0.48T_s(\tau) + 0.42T_m(\tau) - 0.09u(\tau-1) + 71.4$
A_1	C_2	-	$0.90T_s(\tau) + 0.53T_m(\tau) + 12.2u(\tau-1) - 16.4$
A_2	C_3	D_1	$0.73T_s(\tau) + 0.18T_m(\tau) - 15.6u(\tau-1) + 9.06$
-	C_4	-	$1.99T_s(\tau) - 0.28T_m(\tau) + 28.2u(\tau-1) - 65.1$



(a) Fuzzy sets in the model *(b) Recursive simulation of the fuzzy model*

Fig. 11.5 Fuzzy sets used in the model (a). Prediction (dashed line) of unseen validation data in a recursive simulation (b). Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

filters, F_1 and F_2, respectively. The filter parameters are empirically chosen in order to reliably filter the measurement noise and to provide fast responses *.

Simulations experiments have been performed, where the model–plant mismatch is simulated by using a different model to represent the plant. Predictive control as presented in Sec. A.1 is applied to the system, where the prediction and control horizons are set to $H_p = 4$ and $H_c = 2$, respectively, and the weight parameters in Eq. (10.1) are set to $w_{1j} = 1$ and $w_{2j} = 500$. The branch-and-bound method for predictive control as described in Sec. 10.1 is applied with the lower

*The filtered error $e_{mf}(\tau)$ is given by $e_{mf}(\tau) = b_1 e_m(\tau) + b_2 e_m(\tau-1) - a_2 e_{mf}(\tau-1)$, with $b_1 = b_2 = 0.086,\ a_2 = -0.83$. The filtered mixed air temperature is given by $T_{mf}(\tau) = d_1 T_m(\tau) + d_2 T_m(\tau-1) - c_2 T_{mf}(\tau-1)$, with $d_1 = d_2 = 0.245,\ c_2 = -0.51$.

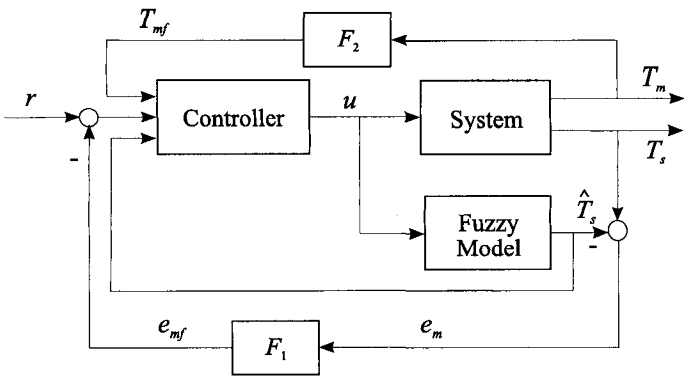


Fig. 11.6 Implementation of MPC in the system using an IMC structure. The controller uses the fuzzy predictive filter. Reproduced from (Sousa and Setnes 1999),©1999 IEEE.

bound $J_{L_i} = 0, \forall i$. The applied set of possible incremental control actions is

$$\Omega = \{-0.05, -0.02, -0.01, 0, 0.01, 0.02, 0.05\}. \tag{11.19}$$

Finally, the fuzzy predictive filter was applied. The scaling factor $\gamma(\tau)$ is computed as in Eq. (11.17), with $\mu_\gamma(\tau)$ given by Eq. (11.16). Figure 11.4 depicts the aggregated membership function μ_γ, where the definition of the two fuzzy criteria μ_e and $\mu_{\Delta e}$ for the HVAC system are also shown. The parameters λ_l have been selected as in Eq. (11.11) and Eq. (11.12) with $l = 3$.

Simulation results obtained both with the fixed alternatives and with the fuzzy predictive filter are presented in Fig. 11.7 and Fig. 11.8. From the results it is seen that both schemes present good control performance. However, the fuzzy predictive filter allows, at the same time, for more vigorous and also for finer control actions. For a non-varying reference, the filtered control actions are practically constant, while the fixed actions in Eq. (11.19) introduce small oscillations in the control. While this has no fatal consequences for the system under study, it is undesirable in general as it increases the control effort and can provoke undesired oscillations. The sum square error decreases 25% using the fuzzy predictive filter. The computational time, however, increases about 20%. Simulations with only three control alternatives have also been performed. The fuzzy predictive filter achieves similar performance in terms of the error, while the computational time is decreased (30%). For the B&B optimization with three fixed alternatives, the computational time decreases considerably (50%), but the sum squared error for the best case increases by at least 40%. Moreover, the chattering effect is quite large, and the control performance is not acceptable. The simulation results are summarized in Table 11.2, relative to the approach with seven fixed alternatives.

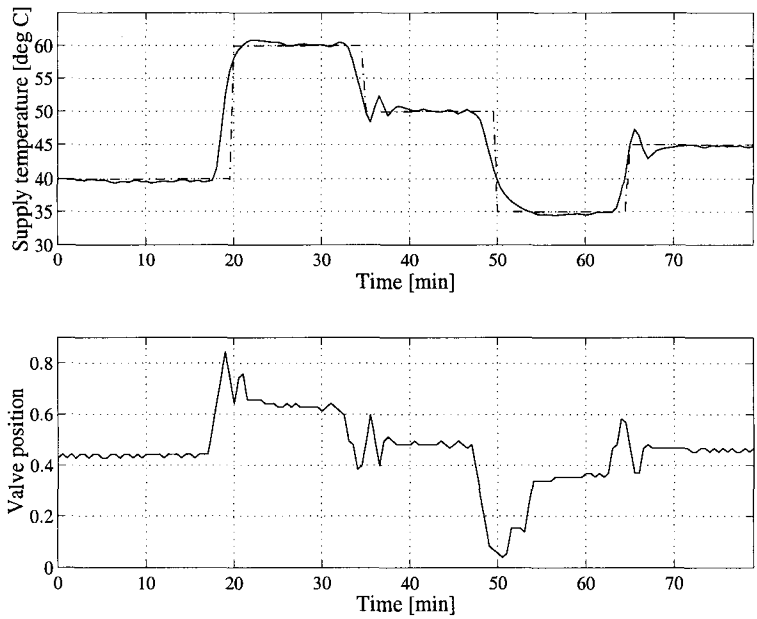


Fig. 11.7 Simulation responses of the HVAC system without filtering the control actions. Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

Table 11.2 Summary of simulation results normalized to the case with 7 fixed control actions. Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

Performance	fixed 7	scaled 7	fixed 3	scaled 3
error	1	0.75	1.4	0.75
computations	1	1.2	0.5	0.7
chattering	moderate	none	big	none

11.5 Summary and concluding remarks

A couple of issues regarding improvements to the performance of optimization in MBPC have been considered. First, special conditions, under which model-based predictive control with fuzzy objective functions remains convex, have been studied. Second, a technique to deal with the problem of reduced accuracy, resulting from the discretization of the control space by generalized search methods, has been described.

When the optimization problem in fuzzy predictive control is convex, efficient

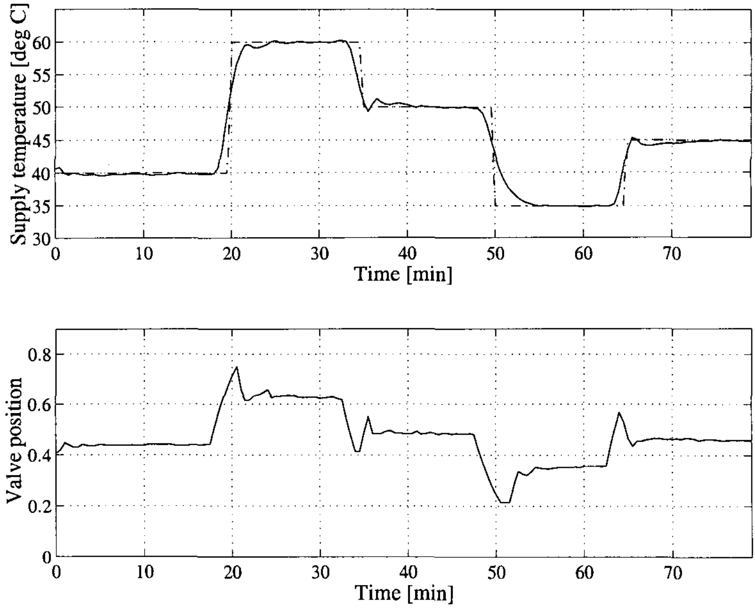


Fig. 11.8 Simulation responses of the HVAC system using fuzzy predictive filters. Reproduced from (Sousa and Setnes 1999), ©1999 IEEE.

optimization methods can be used to solve the problem with polynomial time algorithms. This implies that MBPC with fuzzy objective functions can be used in these cases, even for systems with small sampling times. A set of conditions, under which the convexity of optimization in MBPC can be guaranteed, is given by

(1) the system under control is linear,
(2) only one fuzzy goal at each time step within the prediction horizon is defined, and it strives for the minimization of the prediction error,
(3) the decision criteria are combined by the Yager t-norm, and
(4) the conventional (crisp) constraints on the optimization problem are convex.

Under these conditions, the advantages of fuzzy objective functions in predictive control can be combined with the advantages of convex optimization.

Non-convex optimization problems in model predictive control can be solved using discrete search techniques, such as branch-and-bound or genetic algorithms. The discretization leads, however, to an approximate solution that can generate oscillations around non-varying references and slow step responses. Fuzzy predictive filters can help these problems, keeping at the same time the search space limited, obtaining at each time instance a suitable set of control actions.

The control actions are determined based on simple fuzzy criteria regarding the current and the predicted error. The use of fuzzy predictive filters in control can lead to faster closed-loop response with smaller oscillations in the steady-state. This makes it possible to decrease the number of discrete control alternatives needed.

This page is intentionally left blank

Chapter 12

Application Example

The application of fuzzy decision making methods to control engineering has been demonstrated mainly by simulation examples in the previous chapters. In this chapter, an actual control application is considered, and some of the control methodologies that are discussed in this book are applied to temperature control of an air-conditioning system. In the last decades, the number of air-conditioning systems and heating and ventilating systems installed in buildings has been increasing continuously. By controlling the indoor climate, human comfort can be significantly increased. An increasing number of air-conditioning units installed ask for better control of indoor temperatures, which can combine the increase of comfort with energy saving. Future air-conditioning systems are systems demanding sophisticated control. The combination of several goals, such as energy saving and human comfort, is highly desirable. These goals can be described in a hierarchical structure for intelligent building system control (Shoureshi and Rahmani 1992). This hierarchy consists of a supervisor, a coordinator and a local level control system. An expert control of an air-conditioning plant using a fuzzy rule-based supervisor is presented by Ling and Dexter (1994). Note that these two approaches use linguistic rules based on expert knowledge, always requiring some trial-and-error method to tune the parameters of the controller. The control using hierarchical levels presented by Shoureshi and Rahmani (1992) is an interesting approach, but quite complex. However, using fuzzy goals and constraints, it is possible to simplify the control scheme by concatenating the three levels, supervision, coordination and local controllers, into only one. Moreover, different goals can be used for different control situations. The first step for implementing this approach is presented in this chapter, where predictive controllers with fuzzy objective functions, as presented in Sec. 9.3, are applied to an air-conditioning system.

The air-conditioning system, described in Sec. 12.2, is a good pilot system to demonstrate the method studied, because different control strategies can be tested.

The process has reasonably slow dynamics, which implies that it is feasible to implement control techniques requiring non-convex optimization. In this pilot plant, the working conditions are very close to real air-conditioning systems. Nonlinear fuzzy models are developed for the air-conditioning system. After a general discussion in Sec. 12.1 on air-conditioning systems, the pilot system (the test room) considered in this chapter is described in Sec. 12.2. Section 12.3 presents several models developed for this system. Section 12.4 presents the controllers applied to the system and their respective results, before the concluding remarks in Sec. 12.5.

12.1 Air-conditioning systems

A heating, ventilating and air conditioning (HVAC) system consists of a *primary system* of heat exchangers, pipes or dampers, supplying a medium such as hot water, steam or chilled water to the *terminal system*, which is any heating or cooling unit responsible for the conditioning of a room or building (Levenhagen and Spethmann 1993). Several types of air-conditioning systems are manufactured. In general they are classified into

(1) all-air systems,
(2) air-and-water systems, and
(3) refrigerant-based systems.

Often, air-conditioning systems are designed based on the assumption that the indoor air is well-mixed and only one temperature is assigned to it. However, this is clearly an over-simplification of reality, because the indoor temperature distributions and air flows can not be neglected. The temperature in the room thus changes from place to place, and the control system must consider these factors. Otherwise, the performance of the control is usually poor. The modeling of dynamic indoor temperatures and air flows can be derived by using computational fluid dynamics. This theory, however, is too complex to be considered for control applications, because a huge number of equations based on finite air volumes must be solved iteratively. A possible solution for this problem is to simplify these equations by linearizing them in a state-space model, which can be used for control purposes. This approach is based on the fact that detailed temperature distributions are not necessary, because most people are insensitive to small temperature differences inside a room.

Normally, only one temperature is measured in the control of indoor thermal conditions. Other temperatures in the room must be estimated using a model. Traditional systems used in air-conditioning systems assume that the measured variable is the controlled variable. For most of the air-conditioning systems, the temperature sensor is mounted to measure the temperature of the returned air or the supplied air. The temperature of the working zone is not measured, due to the inconvenience of placing a sensor in a zone where it can be easily damaged. These factors are taken into account for selecting the measured temperatures used for the particular air-conditioning system under control, which is described in the next section.

12.2 Fan-coil systems

One common type of air-conditioning system that uses air and water is the fan-coil unit system. In this type of HVAC systems, the conditioned air is supplied to the unit at medium or high pressure. Such a system (depicted in Fig. 12.1) is considered in this chapter. Hot water at 65 °C is supplied to the coil which exchanges the heat between the hot water and the surrounding air. In the fan-coil unit, the air coming from outside (primary air) is mixed with the return air from the room (recirculated or secondary air). The flows of primary and secondary air are controlled by the outside and return dampers, and by the velocity of the fan, which forces the air to pass through the coil, heating or cooling the air. The global control

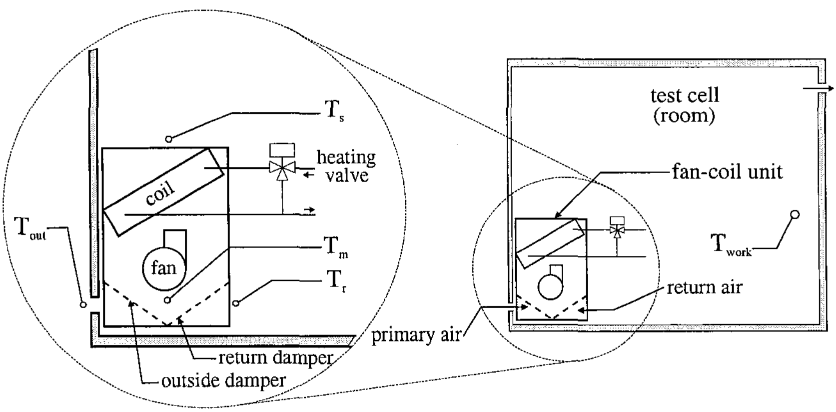


Fig. 12.1 Air conditioning system.

goal for this system is to keep the temperature of the working area in the test cell, T_{work}, at a prescribed reference value, while ensuring that enough ventilation and renovated air is supplied to the room. Three different control actions can be used for this purpose.

(1) Velocity of the fan. The fan has three different velocities: low, medium and

high.

(2) Position of the dampers (outside and return). The dampers can be set in a number of discrete positions, controlling the amounts of air coming from outdoors and returned from the test cell.

(3) Position of the heating valve. The amount of water entering the heat exchanger is controlled by the heating valve, which operates in the range from completely open to completely closed. If this valve is completely open, the quantity of supplied hot water is maximal, and if it is closed, no hot water is supplied to the coil.

In order to control the system, some assumptions are usually made. The fan is kept at low speed in order to maintain human comfort by minimizing the noise level. However, this speed is enough to assure the refreshment of the air in the room. Both dampers are half open, allowing ventilation from the outside, and the return of some air from the test cell to the fan-coil. Thus, in the experiments carried out, only the heating valve is used as a control input. As shown in Fig. 12.1, temperatures can be measured at different locations in the test cell.

The main goal of an air-conditioning system is to control the temperature of the working area T_{work}, assuring that enough renovated air is supplied to the system. Most of the air-conditioning systems control the supply temperature T_s or the return air temperature T_r, assuming that this is the temperature of the working area T_{work}, see Fig. 12.1. This procedure usually leads to poor control performance. A different approach is to build a model relating the measured temperature to the temperature in the working zone. However, the control of the supply temperature in a fan-coil unit is not an easy task, since the model relating the heating valve to this temperature is nonlinear. Moreover, it is strongly influenced by the temperature T_m of the mixed-air before the fan, which should be considered in the model. Therefore, a well controlled supply temperature is necessary to control the temperature in the working area. In this book, the supply temperature is controlled over a wide range of temperatures. The control of the working zone can be obtained directly by applying linear state-space models. Note that the control tests performed in this chapter are made for the fan-coil unit in normal and extreme conditions of functioning. The results are a first step for controlling the air-conditioning system.

The temperature under control, *i.e.*, the supply temperature T_s changes quite quickly compared to all the other temperatures considered, and it is subjected to a large number of disturbances, because the measuring point is very close to the area where the air passing the coil is supplied to the room. The relation between this temperature and the position of the heating valve is nonlinear, and no linear modeling techniques can be applied over the range of all possible temperatures.

12.3 Fuzzy models of the air-conditioning system

The considered air-conditioning system has one input — the opening of the heating valve, and one controlled variable — the supply temperature. The simplest way of modeling the system would be to consider it as a SISO system. However, this model turns out to be quite poor, because it considers neither the buffer effect present in the room nor the disturbance introduced by the outside temperature. In order to consider both these effects, the mixed-air temperature before the fan, $\mathrm{T_m}$ in Fig. 12.1, is also considered as an output of the model. This temperature is measured at an ideal point because it contains both the effects of the outside temperature and of the temperature inside the room. Moreover, it can be included in any industrial air-conditioning system because it is positioned in a safe place (far from the working zone). Besides these considerations based on physical understanding of the process, correlation analysis carried out for other measured temperatures confirms that this temperature is the most relevant to be considered as an output. To summarize, the developed (fuzzy) models have the opening of the heating valve as input ($u(\tau) \in [0, 1]$, with 0 standing for the valve completely closed) and have two outputs: supply temperature $\mathrm{T_s}$ and mixed-air temperature $\mathrm{T_m}$. Delayed values of these three variables are also used for the modeling.

Two fuzzy models are derived in order to apply fuzzy model-based control as described in Chapter 7 and Chapter 9 to the air-conditioning system.

(1) A Takagi–Sugeno fuzzy model obtained by using product-space clustering techniques, as described in Chapter 5.
(2) A Takagi–Sugeno fuzzy model, which is affine in the input $u(\tau)$, such that the model is invertible.

The following sections describe the identification procedures and present the fuzzy models derived.

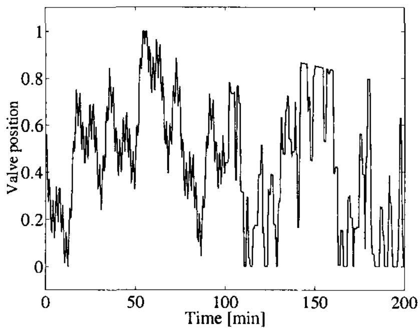
12.3.1 *TS fuzzy model of the air-conditioning system*

This model is constructed from process measurements. The antecedent membership functions and the consequent parameters are estimated from a set of input–output measurements by fuzzy clustering and least-squares methods, as presented in Sec. 5.4.2. After several tests, the sampling period of 30 s was found to be sufficient in order to describe the dynamics of the system. The identification data set contains $N_d = 800$ samples, collected in two different day periods (morning and afternoon), using the input signal data shown in Fig. 12.2a. The excitation signal u consists of a multisinusoidal signal with five different frequencies and amplitudes and of pulses with random amplitude and width. This signal is chosen

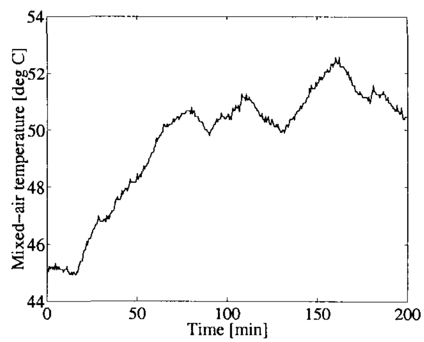
Table 12.1 Model parameters.

output	K	p_1, p_2	m_u
T_s	5	[1, 1]	[2]
T_m	5	[1, 1]	[1]

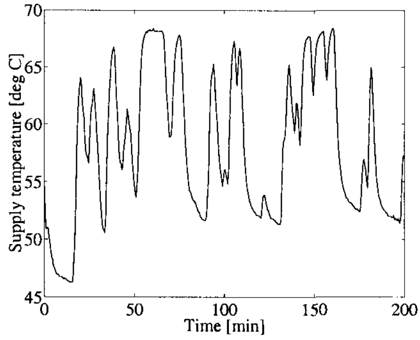
to cover the entire range of the control valve positions and to excite the important frequencies in the expected range of process dynamics. The mean value of this excitation signal is decreased in order to avoid overheating of the test cell, see Fig. 12.2a. Figure 12.2 presents all the data used to identify the fuzzy models.



(a) Valve opening u.



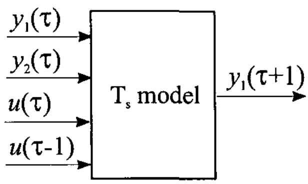
(b) Mixed-air temperature T_m.



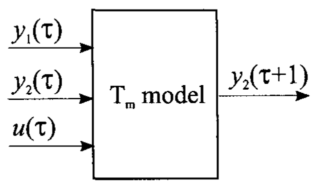
(c) Supply temperature T_s.

Fig. 12.2 Identification data.

The global model can be divided into two models, one for each of the outputs $\mathrm{T_s}$ and $\mathrm{T_m}$. All the variables u, $\mathrm{T_s}$ and $\mathrm{T_m}$ are considered in the premises, making these two sub-models MISO models. Note that a MIMO model can always be decomposed into several MISO models, as discussed in Sec. 5.1. The parameters concerning the number of clusters (rules) K, the orders of the two outputs in each model p_1 and p_2, the order of the input m_u, and the considered delays are presented in Table 12.1. For simplicity of notation, let $y_1 = \mathrm{T_s}$ and let $y_2 = \mathrm{T_m}$ in the following. The number of clusters is initialized to 10 and is reduced using the compatible cluster merging technique from Chapter 6. The final number of clusters is found to be 5. The orders of the inputs and outputs are chosen off-line by comparing several candidate structures of first-order and second-order models in terms of the prediction error criterion. The two MISO models obtained are depicted in Fig. 12.3.



(a) MISO model for the supply temperature $\mathrm{T_s}$.



(b) MISO model for the mixed-air temperature $\mathrm{T_m}$.

Fig. 12.3 Structure of MISO TS fuzzy models.

First, the nonlinear MISO model for the supply temperature is considered. This model is described by the nonlinear function

$$\hat{y}_1(\tau+1) = f_1(y_1(\tau), y_2(\tau), u(\tau), u(\tau-1)), \tag{12.1}$$

where f_1 is a nonlinear mapping. The nonlinear MISO model for mixed-air supply temperature is described by the nonlinear function

$$\hat{y}_2(\tau+1) = f_2(y_1(\tau), y_2(\tau), u(\tau)), \tag{12.2}$$

where f_2 is again a nonlinear mapping.

The complete MIMO model consisting of the two models is validated by using a separate data set, which is measured on another day. Figure 12.4 compares the supply temperature and the mixed-air temperature of the measured and recursively predicted model outputs in a 'free-run' test.

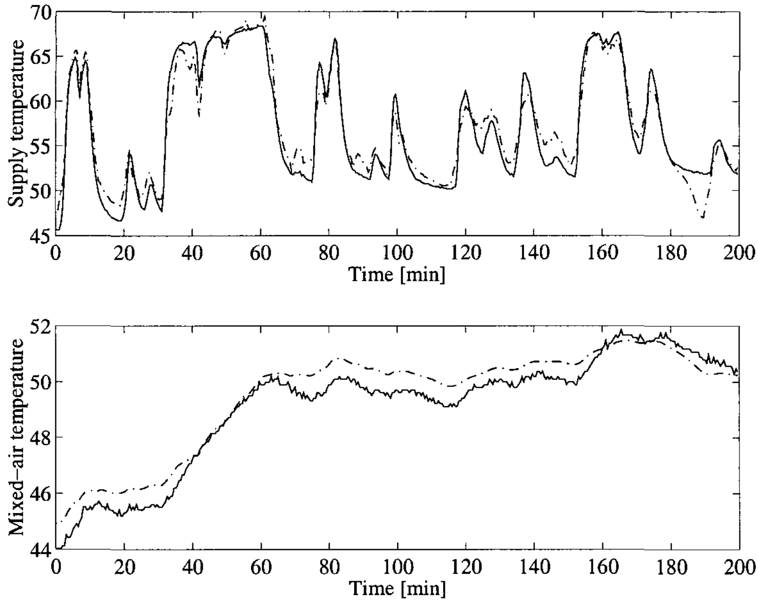


Fig. 12.4 Model validation. Solid line – measured output, dashed line – model output.

Note that both sub-models can follow the real data reasonably well. A widely used measure to test the validity of a model is the Variance Accounted For (VAF). Let the real output be y and the predicted output by the model be given by $\hat{y}$. Denoting 'var' as the variance, the VAF is given by

$$\text{VAF} = 100 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)} \times 100\% . \tag{12.3}$$

For VAF $= 100\%$, the model explains all the variability in the real outputs. The VAF's of the two sub-models for the considered temperatures are given in Table 12.2. From these values it can be concluded that the sub-model for the mixed-air temperature is very good, and the sub-model for the supply temperature is a little worse, but still good.

12.3.2 *Affine TS model of the air-conditioning system*

Inverse control, as presented in Sec. 7.3, can only be applied if the TS model derived for the system under control is affine in the control input $u(\tau)$. The model presented in Sec. 12.3.1 can be made affine by suppressing $u(\tau)$ from the antecedents, and re-identifying the consequent parameters. In doing so, however, the

Table 12.2 VAF values for the validation of the model.

output	VAF
$\mathrm{T_s}$	97.21
$\mathrm{T_m}$	99.58

model revealed a non-minimum phase behavior, which hampers the application of inverse control. Thus, a new model where the non-minimum phase behavior is not present must be identified. Comparing several alternatives, the best affine TS model found for the system is presented in the following.

The identified TS fuzzy model should have the affine structure of Eq. (7.33). By denoting again $y_1 = \mathrm{T_s}$ and $y_2 = \mathrm{T_m}$, the MISO model for the supply temperature is given by

$$\hat{y}_1(\tau + 1) = f_1(y_1(\tau), y_1(\tau - 1), y_2(\tau), y_2(\tau - 1), u(\tau)) . \tag{12.4}$$

The orders of the inputs and outputs are chosen by comparing several candidate structures of first-order and second-order models in terms of the prediction error criterion, while keeping several local linear models with a minimum phase structure. The premises in this model do not contain the membership functions for $u(\tau)$, and the fuzzy model can be inverted, as explained in Sec. 7.3.

The affine TS fuzzy model is validated by a different data set. Figure 12.5 compares the measured supply temperature to the supply temperature that is recursively predicted by the model in a free-run test. When comparing Fig. 12.5 with Fig. 12.4 it is clear that this model is substantially inferior to the non-affine one presented in Sec. 12.3.1. The VAF of this model has the value of VAF $= 89.7$, which confirms the results observed in Fig. 12.5.

12.4 Controllers applied to the air-conditioning system

In order to compare some of the control approaches presented in this book, four different controllers are applied to the air-conditioning system.

(1) PID control.
(2) Inverse control based on an affine TS fuzzy model.
(3) Predictive control based on classical cost functions.
(4) Predictive control based on fuzzy cost functions.

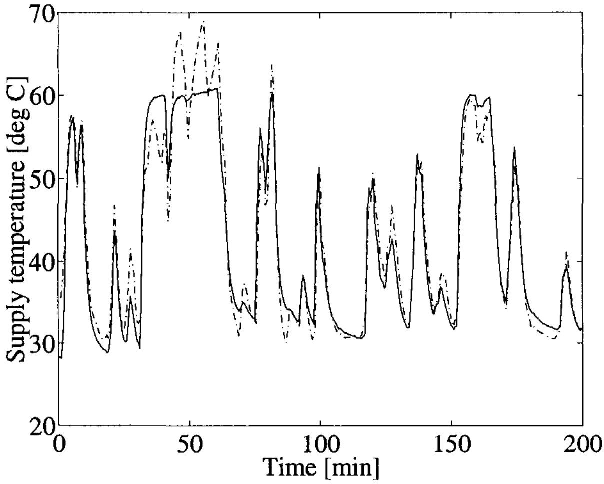


Fig. 12.5 Model validation. Solid line – measured output, dashed line – model output.

Except for the PID controller, the controllers are model-based controllers and implemented inside an IMC scheme in order to cope with model–plant mismatch and disturbances. This IMC scheme is depicted in Fig. 12.6.

The inputs of the controllers are the reference r, the predicted supply temperature $\hat{y}_1$, and the filtered mixed-air temperature y_{2f}. The error signal, $e_m(\tau) = y_1(\tau) - \hat{y}_1(\tau)$, is passed through a first-order low-pass digital Butterworth filter

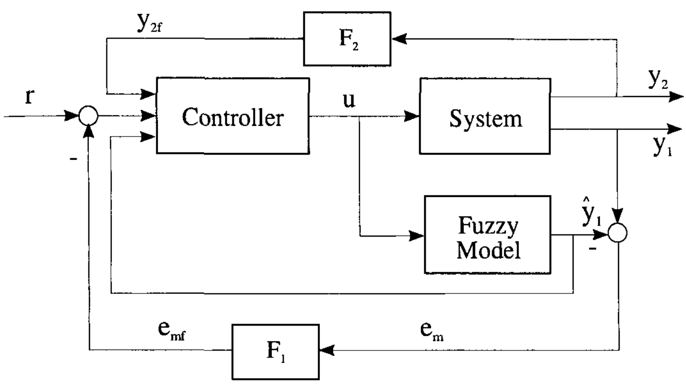


Fig. 12.6 Implementation of the several controllers in the air-conditioning system, using an IMC structure.

F_1. The filtered error $e_{\mathrm{mf}}(\tau)$ is given by

$$e_{\mathrm{mf}}(\tau) = b_1 e_{\mathrm{m}}(\tau) + b_2 e_{\mathrm{m}}(\tau - 1) - a_2 e_{\mathrm{mf}}(\tau - 1)\,, \tag{12.5}$$

where $b_1 = b_2 = 0.086$, and $a_2 = -0.83$. Another first-order low-pass digital Butterworth filter F_2 is designed for y_2,

$$y_{2\mathrm{f}}(\tau) = d_1 y_2(\tau) + d_2 y_2(\tau - 1) - c_2 y_{2\mathrm{f}}(\tau - 1)\,, \tag{12.6}$$

with $d_1 = d_2 = 0.245$, and $c_2 = -0.51$. Note that the filter parameters are chosen based on simulations, in order to reliably filter measurement noises, and to provide fast responses. The mixed-air temperature is fed back directly to the controller, because the sub-model for this temperature is very good (see Table 12.2), and the use of the filter F_2 is enough to guarantee a good control performance. In general, the feedback of the process is not directly used by the controller, because this procedure can cause instability in the closed-loop system. The four different controllers and their respective results are presented in the following sections.

12.4.1 *PID control of the air-conditioning system*

The well-known proportional+integral+derivative (PID) controller is applied to the system (Åström and Hägglund 1995). The parameters of the PID are the following: $K_P = 0.03$, $K_D = 0.06$ and $K_I = 0.003$. These parameters are tuned in order to obtain a fast response avoiding oscillations. Figure 12.7 depicts the response of the supply temperature to several steps in the reference. It is clear that for some temperatures the response is good, but for other temperatures the response is too fast and causes undesirable oscillations. It is possible to eliminate the oscillations, at the cost of very slow responses for low temperatures, which is highly undesirable. Therefore, a PID controller cannot be used to control the system over the whole range of temperatures, unless the parameters are chosen in such a way that the system becomes too slow at certain regions, leading to poor control performance. The poor response of the PID controller confirms the highly nonlinear character of this system.

12.4.2 *Inverse control based on affine TS fuzzy model*

The model presented in Sec. 12.3.2 can be inverted using the inversion method for affine TS fuzzy models, presented in Sec. 7.3. This inverted model is used as the controller, in the control scheme shown in Fig. 12.6. The control structure is applied in real-time control, and the results are depicted in Fig. 12.8. Note that a slowly varying reference must be chosen. In fact, faster changes in the reference

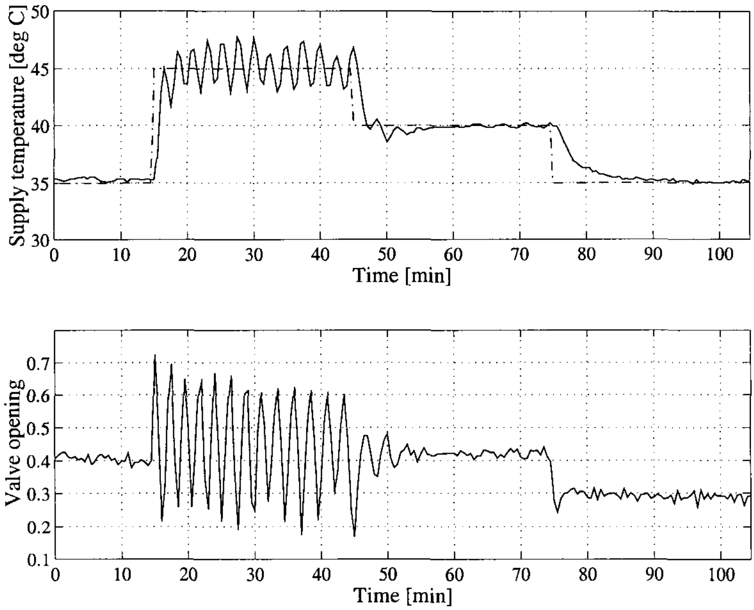


Fig. 12.7 Real-time response of a PID controller. The solid line is the measured output, the dashed line is the reference.

cause oscillations in the supply temperature, due to the severe model–plant mismatch at some temperatures. This phenomenon disappears when predictive control is applied, even for small control and prediction horizons (see Sec. 12.4.3). The fact that the affine TS fuzzy model used is considerably worse than the other models developed for this system (see Sec. 12.3.2 and Sec. 12.3.1) explains the compulsory slow behavior of this control system. Note that when the reference is chosen as in Sec. 12.4.1, the results using this controller are similar to the ones obtained using the PID controller.

12.4.3 *Predictive control based on classical cost functions*

Given the results obtained when using inverse control and PID control, predictive control is applied to the air-conditioning system in order to overcome some of the problems described previously. The TS fuzzy model presented in Sec. 12.3.2 revealed good VAF values for both modeled temperatures. This model is thus suitable to be used in a predictive control scheme. Note that this TS fuzzy model is nonlinear, requiring a non-convex optimization technique to find the best control action, which is applied to the system at each sampling instant. The results presented in Sec. 10.1.2 strongly suggest the use of the branch-and-bound algo-

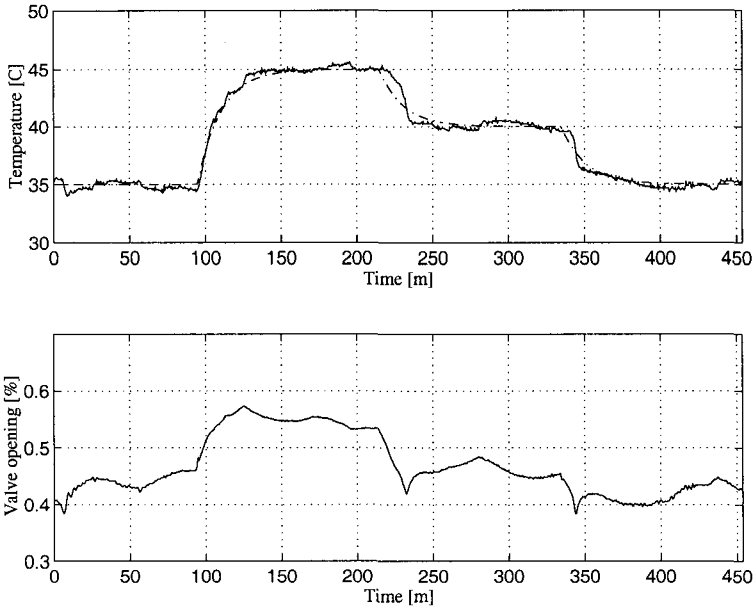


Fig. 12.8 Real-time response using inverse control based on affine TS fuzzy models. The solid line is the measured output, the dashed line is the reference.

rithm applied to predictive control, as described in Sec. 10.1. The main problem is the discretization of the control actions $u(\tau)$, because the B&B method requires a finite, and preferably small, number of possible control actions. Another disadvantage is that predictive control is directly applied without using the control scheme combining inverse control and predictive control as presented in Sec. 7.5. Therefore, the discretization must be carefully chosen in order to avoid the possible chattering effect due to the rough discretization. Note that the control action (heating valve) ranges from completely closed (0) to completely open (1), *i.e.*, the control action is in the interval $[0,1]$. The possible changes in the control actions are chosen, considering the discussed points, as the following at each time step $\tau+j, j=1,\ldots,H_c$,

$$\Omega = [-0.05 \ \ -0.02 \ \ -0.01 \ \ 0 \ \ 0.01 \ \ 0.02 \ \ 0.05]. \tag{12.7}$$

This choice introduces a rate constraint of $\Delta u(\tau+j) \leq 0.05$, which does not significantly alter the performance of the system, but it smoothes the control actions, avoiding undesirable oscillations in the closed-loop system. Further, reasonably small changes of 0.01 are also considered, avoiding the effect of chattering. The parameters chosen for the controller are presented in Table 12.3, and are chosen

Table 12.3 Parameters of the classical predictive controller.

H_c	H_p	α_i	β_i	γ_i
2	4	1	500	50

according to the general guidelines given in Sec. 9.3.2 in the paragraph describing classical objective functions.

The control horizon is chosen as $H_c = 2$ in order to cope with the second order dynamics of the model with respect to the control action $u(\tau)$. A prediction horizon of $H_p = 4$ is shown to be sufficient for this system. Increasing the prediction horizon does not introduce significant improvement in the control results. The calculation time of the control action is about 1 s, which is more than sufficient for the real-time application (note that the sampling period is 30 s). The objective function of this controller is a particular case of Eq. (9.9), and is stated as

$$J = \sum_{j=3}^{4} (\hat{e}(\tau+j))^2 + \sum_{j=2}^{2} 500\,(\Delta u(\tau+j-1))^2 + \sum_{j=3}^{4} 50\,(\Delta\hat{y}(\tau+j))^2 \,. \quad (12.8)$$

The parameters w_{1j}, w_{2j} and w_{3j}, as well as the parameters n_{1l}, n_{1u}, n_{2l}, n_{2u}, n_{3l}, n_{3u} in Eq. (9.9), are chosen based on the scaling between the several variables, and on simulations of the closed-loop system. Only the second change of the control action Δu is considered in Eq. (12.8), which introduces a smooth constraint in this variable at time step $\tau + 1$. Then, the first change in control action $\Delta u(\tau)$ can vary freely in the interval of discretized control actions Ω considered. By just using the error $\hat{e}$ and the change in the output $\Delta\hat{y}$ from $H_c + 1$ to H_p, the control system allows for an increase of freedom in changing the control actions in the first steps. Simulations have shown that this procedure allows for smoother control actions and faster responses, with no overshoot. Real-time results for this predictive controller are presented in Fig. 12.9. The overshoots in the real-time results are caused by the linear filter in the IMC scheme. In fact, the IMC scheme controls the simulated output $\hat{y}_1$, and not the output y_1 itself. Thus, when the system is stabilized at a certain temperature, the error between the output of the model and the real output is also stabilized at a certain value. As the model is nonlinear, in the presence of a step in the reference, the local model describing the system changes, and the error e_{m} also changes. If the filter F_1 is not included in the IMC scheme, this problem is solved in one step. However, this procedure introduces undesired oscillations in the system. Thus, before the error e_{m} stabilizes at its new value (in a new steady-state), these (possibly severe) changes in the error e_m can generate overshoots, unless the change in the reference is smooth enough to avoid them.

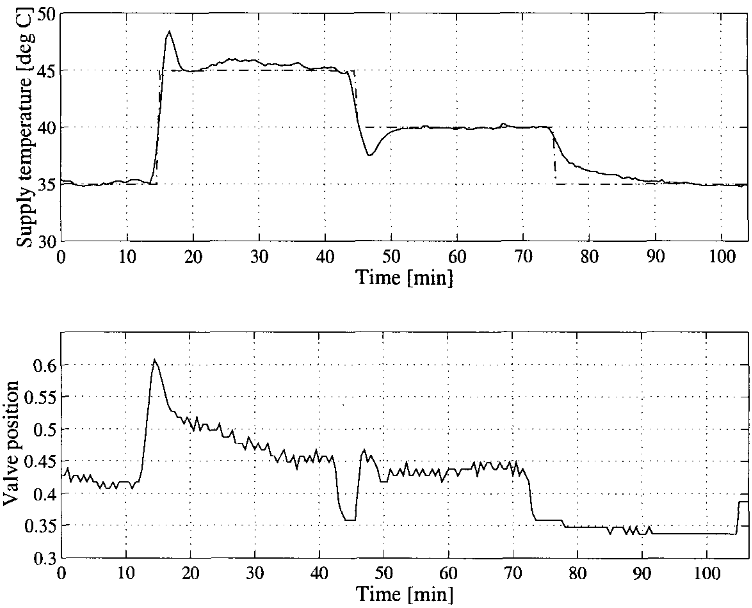


Fig. 12.9 Real-time response with classical predictive control using a TS fuzzy model. The solid line is the measured output, the dashed line is the reference.

In order to eliminate overshoots, the trajectory to be followed is shaped by introducing a filtered reference. A first-order low-pass digital Butterworth filter for the reference is designed for this purpose, which is given by

$$r_f(\tau) = 0.086\, r(\tau) + 0.086\, r(\tau - 1) - 0.83\, r_f(\tau - 1)\,. \tag{12.9}$$

The design of this filter follows the guidelines presented for the other linear filters previously in this chapter. The results obtained using the shaped reference are presented in Fig. 12.10. Note that both overshoots for times t = 15 min and t = 45 min are reduced from 34% and 50% to residual overshoots of about 1%. This is obtained at the cost of significantly increasing the rise time and the settling time. Let us consider the step at time t = 15 min. For this step, the rise time as defined in Eq. (8.9), and for $\lambda = 0.8$ has the value of ϕ_{rise} = 1min for the controller without the shaped reference, while the controller with the shaped reference has the rise time of ϕ_{rise} = 14min; it is thus 14 times slower. The settling time as defined in Eq. (8.10) with $\epsilon = 0.05$ is ϕ_{set} = 15min for the controller with the shaped reference, and ϕ_{set} = 4min for the controller without the filtered reference. Also

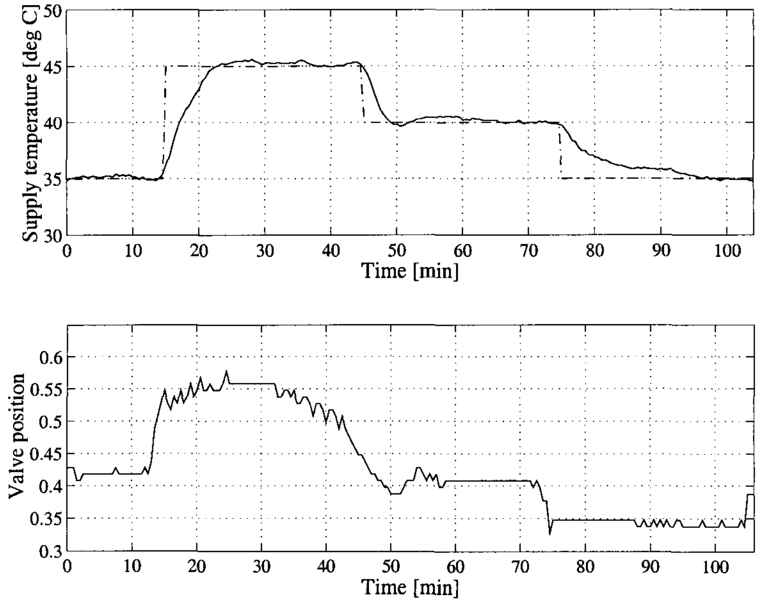


Fig. 12.10 Real-time response with classical predictive control using a TS fuzzy model, using a shaped reference. The solid line is the measured output, the dashed line is the reference.

here, the result is much better, and the controller achieves the steady-state much faster. A compromise between small overshoot and reasonably fast response can be obtained by a proper shape of the reference.

Note that although overshoots can be highly undesirable for many systems, this is not a big issue in air-conditioning systems, and sometimes it can be desirable even to increase human comfort. Imagine that a person enters a very cold room. Comfort is given by increasing the temperature as fast as possible, disregarding the fact that this action results in an overshoot. The same happens if the room is too hot, and the air-conditioning system is cooling down the room. In this system, the overshoot in the supply temperature is not felt in the temperature of the working area, and the overshoot can remain with no problems. Although HVAC systems can usually have overshoots, the shaped reference is still introduced in order to generalize the results to other types of systems.

12.4.4 *Predictive control based on fuzzy cost functions*

The results obtained in the previous section revealed good control performance. The major problems are the overshoots, that can be eliminated by shaping the reference, and a slow response when the reference changes from 40 °C to 35°C, as

Table 12.4 Parameters for the fuzzy predictive controller.

K_e^+	K_e^-	K_y^+	K_y^-	S_y^+	S_y^-	K_u^+	K_u^-	H_u^+	H_u^-
30	-30	1.5	-1.5	0.3	-0.3	1	-1	0.05	-0.05

shown in Fig. 12.9. The overshoot is due to the introduction of the filter F_1 in the IMC control scheme presented in Fig. 12.6. Predictive control using fuzzy objective functions is applied in this section. Fuzzy criteria as described in Sec. 9.3 are used. The branch-and-bound method for predictive control with fuzzy decision functions introduced in Sec. 10.2 is applied in order to find the discrete optimal control actions. The changes in control actions Ω, necessary to perform the optimization, are the same as the ones in Eq. (12.7), allowing for the comparison of the control results. The control and prediction horizons are the same as in Sec. 12.4.3, *i.e.*, $H_c = 2$ and $H_p = 4$. These values turned out to be suitable for controlling the system, and allow for a proper comparison of the fuzzy objective function with the classical one.

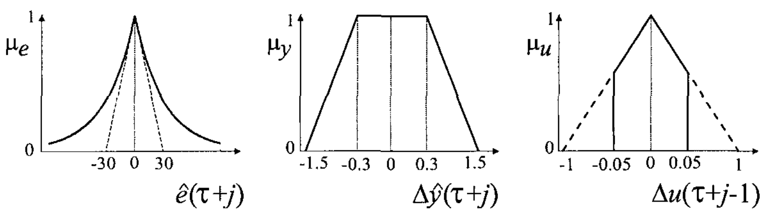


Fig. 12.11 Membership functions of the error, change in output and change in the control action for the air-conditioning system.

The membership functions for the error, change in output and change in the control action, μ_e μ_u and μ_y respectively, are given in Fig. 12.11. Note that these membership functions are the same as in Fig. 9.3, which are defined for a general system. The values of the parameters K_e^+, K_e^-, K_y^+, K_y^-, S_y^+, S_y^-, K_u^+, K_u^-, H_u^+ and H_u^-, described in Sec. 9.3.2 are given in Table 12.4. These values are chosen based on the considerations made for general systems presented in Sec. 9.3. The discussion presented in Sec. 9.4.3 for the paragraph on the air-conditioning system is generally valid here. However, small adjustments for some parameters of the air-conditioning system are required in this section.

The fuzzy goals are combined using the Yager t–norm,

$$\mu_{\pi'} = \sum_{j=3}^{4}(\bar{\mu}_e(\hat{e}(\tau+j)))^\gamma + \sum_{j=2}^{2}(\bar{\mu}_u(\Delta u(\tau+j-1)))^\gamma +$$

$$+\sum_{j=3}^{4}(\bar{\mu}_y(\Delta\hat{y}(\tau+j)))^{\gamma}$$

$$\mu_\pi = \max(0, 1 - \mu_{\pi'}^{1/\gamma}), \qquad \gamma > 0. \tag{12.10}$$

The parameter for this t–norm is chosen as $\gamma_Y = 2$, which allows for a good balance between fast response and small overshoots, as discussed in Sec. 9.3.1. The results obtained with this controller are presented in Fig. 12.12.

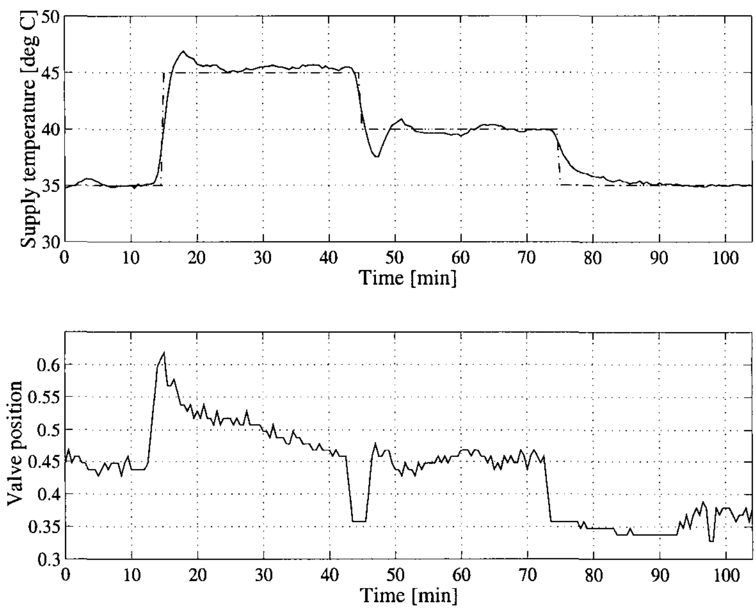


Fig. 12.12 Real-time response using predictive control with fuzzy objective functions. The solid line is the measured output, the dashed line is the reference.

Comparing the response of the system to the one presented in Fig. 12.9, one sees that they are similar. However, the predictive controller with fuzzy objective functions can reduce the overshoot at time $t = 15$ min, from a value of 34% to 20%. The other significant overshoot for time $t = 45$ min remains the same (around 50%). Therefore, there is a slight improvement in the performance of the system. The values for rise and settling times are very similar. Thus, both predictive controllers, classical and fuzzy, present good control performances, and the fuzzy predictive controller can reduce overshoots at some regions. In terms of computational time required, they are similar for both controllers, because both require a non-convex optimization technique. The reference shaping can be applied to reduce (eliminate) overshoots when fuzzy predictive control is applied. The results obtained using fuzzy objective functions are very similar to the ones using classical objective functions (shown in Fig. 12.10). The reduction of these overshoots can be obtained by using a different scheme than IMC to cope with model–plant mismatches and disturbances. The use of fuzzy compensation as presented in Sec. 7.7 was revealed to be able to cope with this problem, but it introduces undesirable oscillations.

12.5 Summary and concluding remarks

Air-conditioning systems are widely used in different spaces like buildings and vehicles. These systems require improved controllers, demanding human comfort and energy saving. Furthermore, these systems have quite general and fuzzy goals, which can be translated to predictive control using fuzzy decision functions.

This chapter presented real time control results from the application of various fuzzy control methods, discussed in this book, to an air-conditioning system. Four controllers are applied to the system.

(1) PID controller
(2) Inverse controller based on an affine TS fuzzy model
(3) Predictive controller based on classical objective functions
(4) Predictive controller based on fuzzy decision functions

The first two are able to control the system when the reference changes very slowly. In general, both predictive controllers can cope with rapidly changing references, and they have good control performance. However, overshoots occur when the reference is not shaped, sometimes due to the use of the IMC scheme. The introduction of fuzzy criteria in the objective function can reduce this phenomenon at certain regions of the system. The shape of the reference can also help to suppress these overshoots.

This page is intentionally left blank

Chapter 13

Future Developments

The relation between fuzzy decision making, fuzzy modeling and fuzzy control has been considered in this book. Starting from a normative formulation of fuzzy decision making, fuzzy modeling and fuzzy control have been investigated from a fuzzy decision making point-of-view. Several combinations of the fuzzy decision making theory and the fuzzy control theory have been explored, leading to some new and novel fuzzy control schemes. The research into these new schemes is not yet complete. Furthermore, the combination of fuzzy decision making and control opens new directions for research into specific applications. This chapter provides some directions for future research that the authors assess to be promising. Only time will tell, however, how correct this assessment will turn out to be.

13.1 Theoretical analysis of FAME controllers

Fuzzy aggregated membership (FAME) controllers from Chapter 4 are designed by direct application of fuzzy aggregation in controllers. In contrast to conventional fuzzy controllers, where several interacting factors, some of which cannot be controlled explicitly, contribute to the nonlinearity of the controller, FAME controllers' nonlinearity can be specified explicitly. The nonlinear behavior of FAME controllers can be analyzed in a more explicit fashion because of this property. The explicit specification of nonlinearity in FAME controllers simplifies the theoretical analysis for such controllers. At the moment, the theoretical properties of the FAME controllers are not fully explored, and hence more research can be expected in this direction.

13.2 Decision support for fuzzy modeling

In Chapter 6, various ways in which the decision making formulation of a problem can help extend the methods used for fuzzy modeling have been considered to provide decision support for certain steps in fuzzy modeling. One example has been the determination of the optimal number of rules in a fuzzy model, when it is obtained by fuzzy product-space clustering. Fuzzy decision making can be used to support other decisions at various stages of fuzzy modeling, such as feature selection, deciding on the trade-off between model accuracy and model transparency, and data filtering (*e.g.*selection of relevant data). In recent years, the trade-off between model accuracy and model transparency has been drawing additional attention (Roubos and Setnes 2001, Guillaume 2001).

13.3 Cooperative control systems

More and more control systems are being implemented in multiple interacting loops. Complex control systems are implemented, where each controller operates in a local context, but many decentralized control systems must operate cooperatively in order to achieve the overall system's design goals. Multiple agent approaches have made their way into control engineering. In these distributed systems, individual goals of controllers must be harmonized with overall system goals. Usually, a hierarchy of levels with different goals is established, which implies that the goals at one level of the hierarchy need to be translated to goals at another level of the hierarchy. Multiactor, multistage decision making is expected to become important for such systems. Since many goals will need to be approximate, and a trade-off between various goals are needed, fuzzy decision making methods can be used. More research into fuzzy decision making in distributed cooperative control systems is expected to clarify the conditions under which fuzzy decision making methods can benefit the controller design for such systems.

13.4 Control with approximate models

One of the promising ideas of fuzzy sets theory has been to investigate and deal with the subjective uncertainty conveyed in the way human beings represent and use information about their environment. The development of fuzzy sets theory in the past years, however, has seen an increased interest in the use of fuzzy systems as nonlinear mappings from their crisp inputs to their crisp (defuzzified) outputs. Many applications of fuzzy modeling and fuzzy control are based on the specification of such mappings. Due to this development, the interest for the original premises of fuzzy sets theory seems to have declined, at least in the field of control engineering. An important reason for this development is that modern control methods based on system models cannot deal with the fuzzy outputs of fuzzy models to design a controller. The extension of control methods to deal with such systems will then bring many systems that fall outside the scope of control theory into its scope.

When the computations of the system and the possible optimization are formulated as a fuzzy decision making problem, there is not a fundamental difference in the way a model with crisp outputs and a model with fuzzy outputs are treated. In addition to the ways discussed in Chapter 9, fuzzy predictive control is also interesting for control engineering in such a setting, as it can deal with approximate models of systems that could not have been dealt with by means of conventional control techniques. Model-based control requires a fairly accurate process model with crisp inputs and crisp outputs. When fuzzy models are used in model based control, their outputs are defuzzified, which reduces the models to nonlinear crisp mappings from their inputs to their outputs. Many systems, however, are known only partially, which implies that the system outputs will be known only approximately. These models, where the inputs can be fuzzy and the outputs are fuzzy, are called here *approximate models*. Approximate models are characterized by fuzzy outputs determined from the system inputs and partial (approximate but not inaccurate) information about the system. In many systems, detailed information for predicting precise system outputs are not available. However, the approximate behavior of the system can be described, for instance, as a set of fuzzy rules. Consider the container crane example discussed in Sec. 9.5. Although the system can be modeled as a coupled set of differential equations, its approximate behavior can also be described by a set of fuzzy rules, which relate the degree of swinging with the velocity or the acceleration of the trolley. The model then becomes less complex, although the prediction becomes less precise. Similarly, it becomes possible by using the approximate models to study many economic systems (where the outputs can only be estimated approximately) using techniques from control engineering. Therefore, the application of fuzzy decision making methods for control with approximate models is an important research topic. This also paves the way for model-based predictive control where the system models are comprised of approximate mathematical models.

Several authors have started considering the formulation of approximate models from measured data. For example, Setnes, van Nauta Lemke and Kaymak (1998) consider identification of approximate models with a Takagi–Sugeno inference structure from approximate measurements of a system. These models can be applied to model-based predictive control within the fuzzy decision making paradigm. The consequence of different alternatives can be evaluated, and once the evaluation is made, they need to be compared to find the best alternative. Hence, fuzzy predictive control opens new possibilities for analyzing systems using all the available information (precise information as well as approximate information).

13.5 Relation to robust control

The use of approximate models in fuzzy control can also be used to study the effect of uncertainty in the model parameters, on the controller parameters. The methods from the robust control theory and the fuzzy set theory can now be combined by using the resolution principle (Klir and Yuan 1995) for the fuzzy sets. By using methods from robust control theory, controllers can be built with guaranteed stability (Palm et al. 1997). The influence of fuzzy model parameters on the controller parameters can be studied in this way, which establishes a relation between fuzzy control and robust control theory.

13.6 Hierarchical fuzzy goals in control applications

In this book, fuzzy goals and fuzzy constraints have often been defined at the same level, without assuming a hierarchy. However, sometimes it is clear that some goals are more important than others, due to *e.g.*safety or economical reasons. The framework of fuzzy decision making in control allows for the definition of hierarchies between the different goals, by defining, for instance, different weights for different goals. It should be mentioned that weight factors can also be used when a hierarchical structure between the different criteria is clearly present. Therefore, this extension will permit the application of model-based predictive control using fuzzy objective functions to more complex processes. Beyond the use of weights, other possible solutions to combine hierarchical fuzzy goals can also be tested. One possibility is to divide general fuzzy goals into sets of fuzzy sub-goals. The problem can be tackled at different levels of the hierarchy, depending on the focus of the control system at a particular moment.

13.7 B&B for MIMO systems

The branch-and-bound algorithms presented in this book are for SISO systems. Their extension to MIMO systems is highly desirable. However, this extension is not straightforward as for the case of genetic algorithms, due to the exponential increase of the computational time. One possible solution is to use fuzzy predictive filters as presented in Sec. 11.3, which must be extended for the multivariable case. Another possibility is to follow a two-step approach, where one uses a smaller number of discretizations, finds a suboptimal solution, and then uses this solution as an initialization of a finer discretization afterwards (Roubos et al. 1999). This method does not guarantee that the solution found is the global optimum, but it may often be a good (satisficing) solution. Note that the finer discretization can be used several times. More research is needed on the possible number of discretizations to be used, and the possible number of times that the finer discretization must be computed. The system performance under the different situations also needs to be tested.

This page is intentionally left blank

Appendix A

Model-Based Predictive Control

The concept of predictive control was first introduced by Richalet et al. (1978) and Cutler and Ramaker (1980) almost simultaneously, describing Dynamic Matrix Control and the Model Algorithmic Control methods, respectively. Since then, a large number of publications have been written on the subject, where Clarke and Mohtadi (1989), who describe Generalized Predictive Control, and Soeterboek (1992), who defines Unified Predictive Control, are two of the most relevant ones.

Model-based predictive control (MBPC) consists of a broad range of control methods having one common feature; the controller is based on the prediction of the future system behavior by using a process model. The basic concepts appearing in all the predictive control approaches are the following (Camacho and Bordons 1995).

- Use of an available (nonlinear) model to predict the process outputs at future discrete time instances over a prediction horizon.
- Computation of a sequence of future control actions using the model of the system by minimizing a certain objective function, which requires that the predicted outputs errors are as close as possible to the desired reference trajectories, under given operation constraints.
- Receding horizon principle, so that at each sampling instant the optimization process is repeated with new measurements, and the first control action obtained is applied to the process.

Because of the explicit use of a process model and the optimization approach, MBPC can be applied to complex processes, *e.g.*multivariable, non-minimum phase, open-loop unstable, nonlinear process or processes with a long time delay. It can also deal with constraints efficiently. Moreover, MBPC has been well received both by the academic world and by the industry. There are a large number of industrial applications for different processes, such as distillation towers and follow-up servos (Richalet 1993), or clinical anesthesia (Linkens and Mahfouf 1994).

Section A.1 describes the basic principles usually found in classical MBPC. Three main problems found in MBPC are discussed afterwards. The modeling of a process is the first step to be performed in order to apply predictive control, and it is described in Sec. A.2. When the model is nonlinear and constraints are present, the optimization problem becomes more complex. Possible solutions for this problem are presented in Sec. A.3. Section A.4 addresses problems related to output errors caused by model–plant mismatch or disturbances.

A.1 Basic definitions

A.1.1 *Control and prediction horizons*

The future plant outputs for a determined *prediction horizon* H_p are predicted at each time instant τ by using a model of the process. The predicted output values $\hat{\mathbf{y}}(\tau + j)$, $j = 1, \ldots, H_p$ depend on the states of process at the current time τ (given, for instance, by the past input and outputs) and on the future control signals $\mathbf{u}(\tau + j)$, $j = 1, \ldots, H_c$, where H_c is the *control horizon.* The control signals change only inside the control horizon, and they remain constant afterwards, *i.e.*,

$$\mathbf{u}(\tau + j) = \mathbf{u}(\tau + H_c - 1), \quad j = H_c, \ldots, H_p - 1. \tag{A.1}$$

The basic principle of model predictive control is depicted in Fig. A.1. The control horizon is usually chosen to be equal to the order of the model. For optimization reasons, or when fuzzy objective functions are utilized (see Chapter 9), this number can be slightly reduced, thereby decreasing the computational costs. The prediction horizon is usually related to the response time of the process for the reference considered. For nonlinear systems, the response time may change, and an estimate of this time must be found. In the presence of fuzzy objective functions, the response time can be slightly reduced due to the flexibility introduced by the fuzzy goals (see Sec. 9.3).

A.1.2 *Objective function*

The sequence of future control signals is obtained by the optimization of a given objective function, which describes the control goal. In classical MBPC, objective functions are usually given by the quadratic form,

$$J(\mathbf{u}) = \sum_{j=1}^{H_p} w_{1j}(\mathbf{r}(\tau + j) - \hat{\mathbf{y}}(\tau + j))^2 + w_{2j}(\Delta\mathbf{u}(\tau + j - 1))^2, \tag{A.2}$$

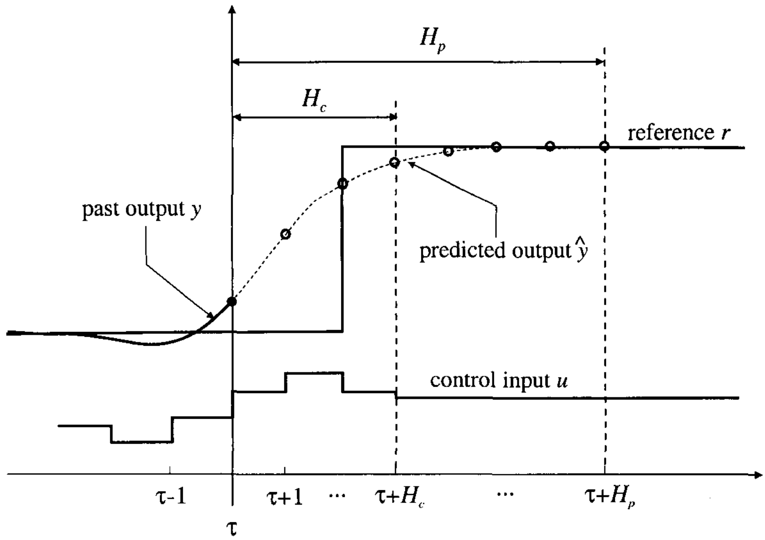


Fig. A.1 Basic principle of predictive control.

or some small modifications of it, where $\hat{\mathbf{y}}$ are the predicted process outputs, $\mathbf{r}$ is the reference trajectory, and $\Delta\mathbf{u}$ is the change in the control signal, weighted by the parameters w_{2j}. Sometimes, a shaped reference trajectory $\mathbf{w}$ is used instead of the reference trajectory $\mathbf{r}$, since the controlled system cannot be expected to be arbitrarily fast. The first term in Eq. (A.2) accounts for the minimization of the output errors, and the second term represents the minimization of the control effort. The term considering the control effort can be given directly by the control actions $\mathbf{u}$, which usually minimize the energy cost. However, the direct weighting of the controller outputs can result in steady-state errors when the process does not contain one or more integrators. This effect is avoided by weighting the change in the control action as in Eq. (A.2). The parameters w_{1j} and w_{2j} determine the weighting between the two terms in the global criterion. Tuning rules for this parameter can be found, *e.g.*, in (Soeterboek 1992). A discussion of a general form of classical objective functions in predictive control is presented in Sec. 9.3.2. Other terms, commonly called 'soft constraints', can also be considered in the classical objective function of Eq. (A.2). Note that for systems with input time delays, only outputs from the time instant τ plus the considered delay until H_p must be considered in Eq. (A.2), because only these outputs can be influenced by the actual control actions $\mathbf{u}(\tau)$. For non-minimum phase systems, the first steps including the non-minimum phase behavior are also not included in the classical MBPC objective function. However, when using fuzzy objective functions, these first steps may be included in the criteria, still leading to controllers with good performance. An example of a linear non-minimum phase system using fuzzy criteria is presented in Sec. 9.3, which compares classical objective functions and fuzzy objective functions using different fuzzy aggregation operators. Note that when fuzzy criteria are used, the objective function can combine different terms in different ways, rather than just by the sum of the terms shown in Eq. (A.2).

A.1.3 *Reference trajectory*

Predictive controllers know the desired reference a priori, and the system can react before the change has effectively been made. The delay effects can thus be avoided in a model predictive control scheme. In the optimization of the objective function as in Eq. (A.2), the reference used is sometimes different from the real reference $\mathbf{r}(\tau + j)$. Normally, a smooth approximation from the actual value of the outputs towards the known reference is considered. This shaped reference $\mathbf{w}(\tau + j)$ is usually approximated by means of a first order system.

$$\begin{aligned} \mathbf{w}(\tau) &= \mathbf{y}(\tau) \\ \mathbf{w}(\tau + j) &= \lambda \mathbf{w}(\tau + j - 1) + (1 - \lambda)\mathbf{r}(\tau + j), \quad j = 1, \ldots, H_p . \end{aligned} \tag{A.3}$$

The parameter λ is in the interval $[0, 1]$. When it is close to one, the shaped reference is smoother, and it influences the dynamic response of the system. Other forms of shaping the reference can also be used (Clarke and Mohtadi 1989). This shaped reference avoids sudden changes in the control actions and local instability of the system, at the cost of slower responses.

A.1.4 *Receding horizon principle*

When the (nonlinear) model of the process predicts the process output exactly and the system is not subjected to disturbances, the errors between the predicted and the measured outputs are zero, *i.e.*, there is no model–plant mismatch. This control structure is thus a simple feedforward controller. This situation is ideal, and usually the predicted outputs $\hat{\mathbf{y}}(\tau + 1)$ are different from the process outputs $\mathbf{y}(\tau + 1)$. For this reason, only the control signals $\mathbf{u}(\tau)$ are applied to the process. The control signals at the next sampling instances $\mathbf{u}(\tau + 1), \ldots, \mathbf{u}(\tau + H_p - 1)$ are discarded, because at the next sampling instant the process outputs $\mathbf{y}(\tau + 1)$ are known and the optimization can be repeated using the updated data. The new $\mathbf{u}(\tau + 1)$ that are calculated using this strategy are usually different from the ones obtained previously, due to the new information available. This technique intends to reduce errors due to model–plant mismatches and disturbances, but another control scheme can still be needed to cope with this problem (see Sec. A.4).

A.1.5 *Classical MBPC scheme*

As the process outputs are fed back to the optimization algorithm in order to recompute the optimal control actions at each sampling instant, the MBPC scheme is a combination of open-loop (prediction part) and feedback (optimization at every time instant). This control structure is called an 'open-loop' feedback control structure, and is presented in Fig. A.2. Note that with this control scheme, the model is used depending on the updated values of the process. The controller contains the model of the system, the objective function, an optimizer and the reference generator. The optimizer calculates the optimal solution for the given objective function using the model of the system and the given reference.

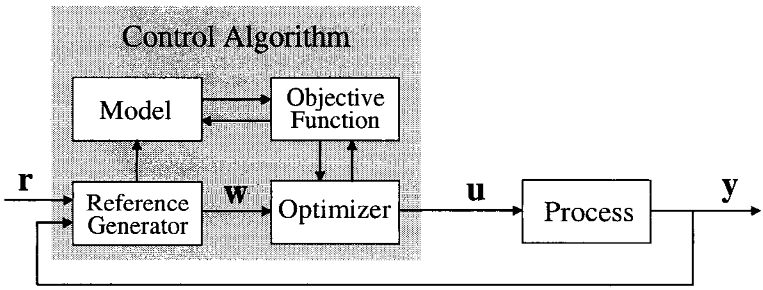


Fig. A.2 Classical model predictive control scheme.

The process inputs and outputs, as well as state variables, can be subjected to constraints, which are incorporated in the optimization problem as 'hard' or 'soft' constraints. Commonly, magnitude (or level) and rate constraints are considered for the control actions, and level constraints are considered for the outputs. For objective functions using fuzzy criteria, the distinction between hard and soft constraints often disappears. Fuzzy sets describing the several criteria include both constraint types a priori. A detailed explanation on the use of fuzzy criteria for the objective function in MBPC is given in Sec. 9.1.

A.2 Modeling in MBPC

The performance of MBPC depends largely on the accuracy of the process model. If the accuracy of the model decreases, the performance of the controller also decreases. Hence, a large part of the classical MBPC design effort is related to modeling and identification (Richalet 1993), where 'classical MBPC' means that fuzzy models are not utilized, and cost functions are as in Eq. (A.2).

The model of the process must be able to predict the future process output, must be simple to implement in the control algorithm, must perform fast simulations, and, preferably, have a physical background, such that it can be understood by an operator or designer. Conventional modeling approaches based on physical modeling or linear system identification cannot derive reliable models for complex or partly known systems. For these types of problems, fuzzy models, as presented in Sec. 5.2, can be used advantageously. For this reason, several examples in this book use this type of modeling techniques to derive nonlinear models. We have also given examples with linear models sometimes, such as in Sec. 9.4.1, in order to emphasize the advantages of MBPC with fuzzy objective functions, even when a linear system is considered.

A.3 Optimization problems

If a linear model of the system is used in the control algorithm, the objective function is described by a quadratic functon as in Eq. (A.2), and no constraints are active, the optimization problem has an analytical solution. However, different parameters in Eq. (A.2), such as control and prediction horizons H_c and H_p, and the weight factors w_{1j} and w_{2j}, must still be tuned. When a convex constraint is present, the optimization problem becomes a quadratic programming (QP) problem to be solved at each time instant (Camacho and Bordons 1995). This nonlinear optimization problem is convex and can be solved by using gradient-descent methods with a guaranteed global solution.

However, both nonlinear models and constraints are present in the most general case. Then, the optimization problem is non-convex. The most relevant techniques used in this case are the *Sequential Quadratic Programming* (SQP) method, see, *e.g.*Gill et al. (1981) and the *simplex method* introduced by Nelder and Mead (1965), which are both iterative optimization techniques. These methods usually hamper the application of MBPC to fast systems, because being iterative methods, they generally have high computational costs, which make them not suitable to be used in systems with short sampling times. Moreover, the convergence can result in local minima, which usually results in poor performance of the MBPC scheme. Alternative optimization methods for non-convex optimization problems can be used when the solution space is discretized. By discretizing the solution space, the problem is transformed into a discrete optimization problem, where techniques such as branch-and-bound or genetic algorithms can be applied. Chapter 10 and Chapter 11 discuss implementation issues regarding classical and fuzzy MBPC.

A.4 Compensation of model–plant mismatch and disturbances

The MBPC scheme as presented in Fig. A.2 must deal with model–plant mismatch and the influence of disturbances due to the open-loop feedback control strategy. Sometimes, shaping the reference is enough to reduce this problem. Moreover, some MBPC schemes contain a model of the disturbances, *e.g.* generalized predictive control (Clarke and Mohtadi 1989), which can reduce their influence significantly. However, this strategy is difficult to apply in the presence of nonlinear systems, where the modeling of the disturbances is often quite difficult. Therefore, a different and preferably robust control scheme is desired. When models for the disturbances are difficult to identify, it is preferable to incorporate MBPC in a control scheme that can eliminate the effect of the disturbances and model errors. The *internal model control* (IMC) scheme is able to deal with these phenomena. This technique is explained briefly in Sec. B.1. *Fuzzy compensation* introduced in Chapter 7 is another scheme to tackle the problem of model–plant mismatch. This technique can have advantages over IMC because additional information of the model is used in the system.

This page is intentionally left blank

Appendix B

Nonlinear Internal Model Control

Control techniques based on a nonlinear process model can effectively cope with the dynamics of nonlinear systems. Some model-based control techniques, such as MBPC, however, can induce steady-state errors due to model–plant mismatch or disturbances, depending on the number of integrators in the process, the type of disturbances, *e.g.* offset or process disturbances, and the required reference following accuracy. Beyond shaping the reference that can cope with this problem to some extent, a scheme is needed to compensate for these errors. One example of such a scheme is the *Internal Model Control* (IMC) explained below.

B.1 Classical internal model control

A classical approach to reject steady-state errors and disturbances is the use of an integral control action. The integral action can be implemented using, for instance, an additional outer-loop integral controller to eliminate the steady-state error between the outputs of the system and the references. If the integral action is applied to nonlinear systems, the integral gain must be different for different regions of the system outputs. If the parameter is too large it provokes undesirable oscillations, and if it is too small the system converges slowly to the set-point. To cope with this situation, it is possible to have a supervisory controller that tunes the integral parameter for the different regions. A major drawback of this method is that this solution requires tuning rules which are mostly based on trial-and-error. Another, more robust solution to eliminate steady-state errors is the use of the IMC scheme presented in this section. An unifying overview of internal model control emphasizing the robust characteristics of this control concept is presented by Garcia and Morari (1982). Internal model control (IMC) consists generally of three parts.

(1) A model to predict the effect of the control action on the system.
(2) A controller based on an inverse of the process model.

(3) A filter to increase robustness.

A general IMC scheme for SISO systems is depicted in Fig. B.1, where P denotes the process, M is a model of process, C represents the controller and F is a filter. Note that this scheme can be generalized for MIMO systems.

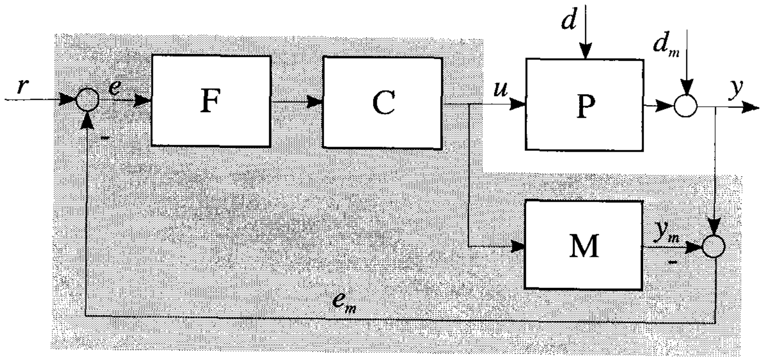


Fig. B.1 General internal model control scheme.

The disturbances are separated in process disturbances d and the measurement (additive) disturbances d_m. The output is simply given by $y = \mathrm{P}\,u$. Supposing that there is no filter ($\mathrm{F} = 1$) the following relationships can be derived from Fig. B.1:

$$e = r - y + y_m \tag{B.1a}$$
$$y_m = \mathrm{M}\,\mathrm{C}\,e \tag{B.1b}$$
$$e_m = (\mathrm{P} - \mathrm{M})\,\mathrm{C}\,e \tag{B.1c}$$

Note that all the mappings P, C and M can be nonlinear, and the effect of the disturbance d is considered in P. The following properties of nonlinear IMC can be stated from the three equations in Eq. (B.1) (Economou et al. 1986).

Proposition B.1 **(Stability)** *If* P *and* C *are input–output stable, and if a perfect model of the plant is available, i.e.,* $\mathrm{M} = \mathrm{P}$*, then the closed-loop system is input–output stable, too.*

Proposition B.2 **(Perfect control)** *If the right inverse of the model operator* M^r *exists,* $\mathrm{C} = \mathrm{M}^r$*, and the closed-loop system is input–output stable with this controller, then the control is perfect, i.e.,* $y = r$.

Proposition B.3 **(Zero offset)** *If the right inverse of the steady-state model operator* M_{∞}^{r} *exists,* $\mathrm{C} = \mathrm{M}_{\infty}^{r}$*, and the closed-loop system is stable with this controller, then for asymptotically constant inputs offset-free control is achieved.*

Proposition B.1 can easily be proven, since the feedback loop has no influence when $\mathrm{M} = \mathrm{P}$, $e_m = 0$, and the system is in open loop. As both controller and process are open loop stable, the global system is also stable. Proposition B.2 is a direct consequence of Eq. (B.1a) and Eq. (B.1b), when $\mathrm{C} = \mathrm{M}^r$. Finally, Proposition B.3 is also a result from the direct application of Eq. (B.1a) and Eq. (B.1b), by using the limit as $t \to \infty$.

Some remarks concerning the practical significance of the above properties can be given (Economou et al. 1986). General guidelines for the design of a feedback controller are not available, if a nonlinear system is considered. This is even more difficult if some desired performance specifications are desired. The IMC scheme reduces the design problems for systems that are input–output stable or that can be stabilized by output feedback. For systems with these characteristics, and when a good model of the plant is available, Proposition B.2 defines the structure and parameters of the controller resulting in perfect control. Thus, in this simple case, IMC transforms the controller design in a feedforward control problem, which can be solved for nonlinear systems also. Moreover, IMC still preserves the advantages of feedback control, especially the elimination of unknown plant disturbances, as suggested by Proposition B.2 and Proposition B.3.

The filter F in Fig. B.1 is introduced to increase the robustness of the control system, when the system is subjected to model–plant mismatches, and process or measurement disturbances. The filter can also project the error signal e in the appropriate space, such that the input space of the controller is in the range of the operator M and of the system P, by reducing the loop gain. Finally, the filter can smooth out noisy or rapidly changing signals, reducing the transient response of the IMC controller. For nonlinear systems the filter must be designed for the part of the system where the dynamics is faster. If this is not the case, the system can show undesirable overshoots or even oscillations.

B.2 MBPC in an internal model control scheme

The model predictive controller can be incorporated in the internal model control scheme, as presented in Fig. B.2. Note that the filter is included in the feedback loop, filtering the noise, stabilizing the loop by decreasing the gain, and providing more robustness to the loop. The use of predictive control in an IMC structure allows for the reduction of model errors and disturbance effects, in an effective way. IMC has first been used for inverse control, as in Sec. B.1, but note that predictive control can be regarded as a generalization of inverse control. In fact, when $H_c = H_p = 1$, $w_{1j} = 1$, $w_{2j} = 0$ in Eq. (A.2), and a control command exists such that $\mathbf{y}(\tau+1) = \mathbf{r}(\tau+1)$, a global optimum of the objective function in Eq. (A.2) results in $J = 0$. Thus, without constraints and without penalizing the control action, this can be obtained by the inversion of the model, which can be computed numerically by means of a function minimization, as discussed in Sec. A.3. This situation is of course ideal and normally unrealistic. The extension of the control and prediction horizons, the generalization of the objective function and the inclusion of constraints is a generalization of inverse control, and thus the IMC scheme can be applied advantageously to MBPC.

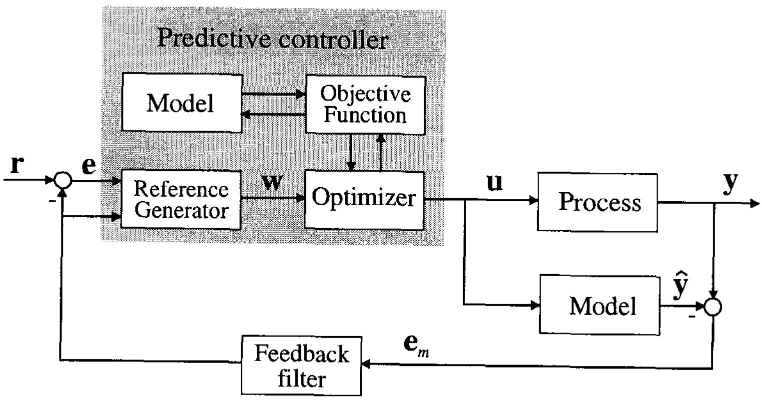


Fig. B.2 MBPC in an internal model control scheme.

Bibliography

Åström, K. J. and Hägglund, T. (1995). *PID Controllers: Theory, Design and Tuning*, Research Triangle Park, USA, Instrument Society of America.

Baas, S. M. and Kwakernaak, H. (1977). Rating and ranking of multiple-aspect alternatives using fuzzy sets, *Automatica*, **13**, 47–58.

Babuška, R. (1998). *Fuzzy Modeling for Control*, Boston, Kluwer Academic Publishers.

Babuška, R., Sousa, J. M. and Verbruggen, H. B. (1995). Model-based design of fuzzy control systems, *Proceedings of the Third European Congress on Fuzzy and Intelligent Technologies, EUFIT'95*, Aachen, Germany, pp. 837–841.

Babuška, R., Sousa, J. M. and Verbruggen, H. B. (1998). Inverse fuzzy model based predictive control, *in* D. Driankov and R. Palm (eds), *Advances in Fuzzy Control*, Physica-Verlag, Heidelberg, New York, pp. 129–154.

Backer, E. (1995). *Computer-Assisted Reasoning in Cluster Analysis*, New York, Prentice Hall.

Baldwin, J. F. and Pilsworth, B. W. (1982). Dynamic programming for fuzzy systems with fuzzy environment, *Journal of Mathematical Analysis and Applications*, **85**, 1–23.

Baptista, L. F., Sousa, J. M. and da Costa, J. S. (2001). Fuzzy predictive algorithms applied to real-time force control, *Control Engineering Practice*, **9**(4), 411–423.

Batur, C., Srinivasan, A. and Chan, C.-C. (1993). Inverse fuzzy model controllers, *Proc. of the American Control Conference*, San Francisco, USA.

Bellman, R. E. (1957). *Dynamic Programming*, Princeton, New Jersey, Princeton University Press.

Bellman, R. E. and Zadeh, L. A. (1970). Decision making in a fuzzy environment, *Management Science*, **17**(4), 141–164.

Berenji, H. R. and Khedkar, P. (1992). Learning and tuning fuzzy logic controllers through reinforcements, *IEEE Transactions on Neural Networks*, **3**, 724–740.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function*, New York, Plenum Press.

Bezdek, J. C. and Pal, S. K. (eds) (1992). *Fuzzy Models for Pattern Recognition*, New York, IEEE Press.

Bonissone, P. P. (1979). A pattern recognition approach to the problem of linguistic approximation in system analysis, *Proceedings of the IEEE International Conference on Cybernetics and Society*, Denver, Colorado, pp. 793–798.

Bortolan, G. and Degani, R. (1985). A review of some methods for ranking fuzzy subsets, *Fuzzy Sets and Systems,* **15**, 1–19.

Boullart, L., Krijgsman, A. and Vingerhoeds, R. A. (eds) (1992). *Application of Artificial Intelligence in Process Control,* Oxford, Pergamon Press.

Boyd, S. P. and Barret, C. H. (1991). *Linear Controller Design – Limits of Performance,* Englewood Cliffs, Prentice Hall.

Braae, M. and Rutherford, D. A. (1979). Theoretical and linguistic aspects of the fuzzy logic controller, *Automatica,* **15**, 553–577.

Brown, M. and Harris, C. (1994). *Neuro-fuzzy Adaptive Modelling and Control,* New York, Prentice Hall.

Buckley, J. J. (1985). Ranking alternatives using fuzzy numbers, *Fuzzy Sets and Systems,* **15**, 21–31.

Calise, A., Hovakimyan, N. and Idan, M. (2001). Adaptive output feedback control of nonlinear systems using neural networks, *Automatica,* **37**(8), 1201 – 1211.

Camacho, E. F. and Bordons, C. (1995). *Model Predictive Control in the Process Industry,* London, Springer Verlag.

Chang, C. S., Phoa, Y. H., Wang, W. and Thia, B. S. (1996). Economy/regularity fuzzy logic control of dc railway systems using event-driven approach, *IEE Proceedings Electrical Power Applications,* **143**(1), 9–17.

Chen, S.-H. (1985). Ranking fuzzy numbers with maximizing set and minimizing set, *Fuzzy Sets and Systems,* **17**, 113–129.

Chen, S.-J. and Hwang, C.-L. (1992). *Fuzzy Multiple Attribute Decision Making: Methods and Applications,* number 375 in *Lecture Notes in Economics and Mathematical Systems,* Berlin, Springer.

Chen, X. and Bushnell, M. (1996). *Efficient Branch and Bound Search with Applications to Computer-Aided Design,* Boston, Kluwer Academic Publishers.

Clarke, D. W. and Mohtadi, C. (1989). Properties of generalised predictive control, *Automatica,* **25**(6), 859–875.

Cutler, C. R. and Ramaker, B. L. (1980). Dynamic matrix control - a computer control algorithm, *Proc. JACC,* San Francisco, USA.

Cybenko, G. (1989). Approximations by superpositions of a sigmoidal function, *Mathematical Control Signal Systems,* **2**, 303–314.

Davidor, Y. (1991). *Genetic Algorithms and Robotics: a heuristic strategy for optimization,* Singapore, World Scientific.

Dias Jr., O. P. (1993). Ranking alternatives using fuzzy numbers: a computational approach, *Fuzzy Sets and Systems,* **56**, 247–252.

Dombi, J. (1982). A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness induced by fuzzy operators, *Fuzzy Sets and Systems,* **8**(2), 149–163.

Driankov, D., Hellendoorn, H. and Reinfrank, M. (1993). *An Introduction to Fuzzy Control,* Berlin, Springer Verlag.

Dubois, D. and Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications,* Vol. 144 of *Mathematics in Science and Engineering,* London, Academic Press. ISBN 0-12-222750-6.

Dubois, D. and Prade, H. (1983). Ranking fuzzy numbers in the setting of possibility theory, *Information Sciences,* **30**, 183–224.

Dubois, D. and Prade, H. (1985). A review of fuzzy set aggregation connectives, *Information Sciences,* **36**, 85–121.

Dubois, D. and Prade, H. (1986). New results about properties and semantics of fuzzy set-theoretic operators, *in* P. P. Wang and S. K. Chang (eds), *Fuzzy Sets*, Plenum Press, New York, USA, pp. 59–75.

Dubois, D. and Prade, H. (1988). *Possibility Theory: an Approach to Computerized Processing of Uncertainty*, New York, Plenum Press.

Dubois, D. and Prade, H. (1992). Gradual inference rules in approximate reasoning, *Information Sciences*, **61**, 103–122.

Dullerud, G. E. and Paganini, F. (2000). *A Course in Robust Control Theory: A Convex Approach*, Vol. 36 of *Texts in Applied Mathematics*, New York, Springer.

Dunn, J. C. (1974). A graph theoretic analysis for pattern classification via Tamura's fuzzy relation, *IEEE Transactions on Systems, Man and Cybernetics*, **4**(3), 310–313.

Dyckhoff, H. and Pedrycz, W. (1984). Generalized means as model of compensative connectives, *Fuzzy Sets and Systems*, **14**, 143–154.

Economou, C. G., Morari, M. and Palsson, B. O. (1986). Internal model control. extension to nonlinear systems, *Ind. Eng. Chem. Process Des. Dev.*, **25**, 403–411.

Edwards, W. (1977). How to use multiattribute utility measurement for social decision making, *IEEE Transactions on Systems, Man and Cybernetics*, **7**, 326–340.

Efstathiou, J. (1984). Practical multi-attribute decision-making and fuzzy set theory, *in* H.-J. Zimmermann, L. A. Zadeh and B. R. Gaines (eds), *Fuzzy Sets and Decision Analysis*, Vol. 20 of *Studies in the Management Sciences*, North Holland, Amsterdam, pp. 307–320.

Esogbue, A. O., Theologidu, M. and Guo, K. (1992). On the application of fuzzy sets theory to the optimal flood control problems arising in water resources systems, *Fuzzy Sets and Systems*, **48**, 155–172.

Filev, D. P. and Yager, R. R. (1991). A generalized defuzzification method via BAD distributions, *International Journal of Intelligent Systems*, **6**, 687–697.

Fischer, M., Nelles, O. and Isermann, R. (1998). Adaptive predictive control of a heat exchanger based on a fuzzy model, *Control Engineering Practice*, **6**, 258–269.

Fletcher, R. (1970). A new approach to variable metric algorithms, *Computer Journal*, **13**, 317–322.

Fodor, J., Marichal, J.-L. and Roubens, M. (1995). Characterization of the ordered weighted averaging operators, *IEEE Transactions on Fuzzy Systems*, **3**(2), 236–240.

Francelin, R. A. and Gomide, F. A. C. (1993). A neural network for fuzzy decision making problems, *Proceedings of Second IEEE International Conference on Fuzzy Systems*, Vol. 1, San Francisco, USA, pp. 655–660.

Friedman, J. H. (1991). Multivariate adaptive regression splines, *Annals of Statistics*, **19**(1), 1–141.

Frigui, H. and Krishnapuram, R. (1996). A robust algorithm for automatic extraction of an unknown number of clusters from noisy data, *Pattern Recognition Letters*, **17**, 1223–1232.

Fung, L. W. and Fu, K. S. (1977). Characterization of a class of fuzzy optimal control problems, *in* M. M. Gupta, G. N. Saridis and B. R. Gaines (eds), *Fuzzy Automata and Decision Processes*, North-Holland, New York, pp. 209–219.

Garcia, C. E. and Morari, M. (1982). Internal model control: a unifying review and some new results, *Ind. Eng. Chem. Process Des. Dev.*, **21**, 308–323.

Gath, I. and Geva, A. B. (1989). Unsupervised optimal fuzzy clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(7), 773–781.

Giles, R. (1976). Łukasiewicz logic and fuzzy set theory, *International Journal of Man-Machine Studies*, **8**, 313–327.

Gill, P. E., Murray, W. and Wright, M. H. (1981). *Practical Optimization*, New York, Academic Press.

Gluss, B. (1973). Fuzzy multi-stage decision-making fuzzy state and terminal regulators and their relationship to non-fuzzy quadratic state and terminal regulators, *International Journal of Control*, **17**(1), 177–192.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, New York, Addison–Wesley.

Goodrich, M. A., Stirling, W. C. and Frost, R. L. (1998). A theory of satisficing decisions and control, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, **28**(6), 763–779.

Goodrich, M. A., Stirling, W. C. and Zanaboni, M. (1999). Model predictive satisficing fuzzy logic control, *IEEE Transactions on Fuzzy Systems*, **7**(3), 319–332.

Grabisch, M. (1993). On the use of fuzzy integral as a fuzzy connective, *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 213–218.

Grabisch, M. (1996). The application of fuzzy integrals in multicriteria decision making, *European Journal of Operational Research*, **89**, 445–456.

Grabisch, M., Nguyen, H. T. and Walker, E. A. (1995). *Fundamentals of Uncertainty Calculi With Applications to Fuzzy Inference*, Vol. 30 of *B: Mathematical and Statistical Methods*, Dordrecht, Kluwer Academic Publishers.

Graham, B. and Newell, R. (1988). Fuzzy identification and control of a liquid level rig, *Fuzzy Sets and Systems*, **25**, 255–273.

Guillaume, S. (2001). Designing fuzzy inference systems from data: an interpretability-oriented review, *IEEE Transactions on Fuzzy Systems*, **9**(3), 426–443.

Gustafson, D. E. and Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix, *Proc. IEEE CDC*, San Diego, USA, pp. 761–766.

Hamacher, H. (1978). Über logische verknupfungen unscharfer aussagen und deren zugehörige bewertungsfunktionen, *in* G. J. K. R. Trappl and L. Ricciardi (eds), *Progress in Cybernetics and Systems Research*, Vol. 3, Hemisphere, Washington (DC), USA.

Hardy, G. H., Littlewood, J. E. and Polya, G. (1973). *Inequalities*, second edn, London, Cambridge University Press.

Henson, M. A. and Seborg, D. E. (1990). Input-output linearization of general nonlinear systems, *AIChE Journal*, **36**, 1753–1757.

Heydeman, J., Reijntjes, R., Babuska, R., Kaymak, U. and van Nauta Lemke, H. R. (1996). Fuzzy logic based security assessment of power networks, *Proceedings International Conference on Intelligent Systems Applications to Power Systems ISAP'96*, Orlando, Fl., pp. 405–409.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor, MIT Press.

Holmblad, L. P. and Østergaard, J. J. (1982). Control of a cement kiln by fuzzy logic, *in* M. M. Gupta and E. Sanchez (eds), *Fuzzy Information and Decision Processes*, North Holland, New York, pp. 389–399.

Holt, D., Kaymak, U., Klaassens, J. B. and van Nauta Lemke, H. R. (1997). A fuzzy decision support system for designing magnetic components, *Seventh European Conference on Power Electronics and Applications*, Trondheim, Norway, pp. 2.022–2.027.

Höppner, F., Klawonn, F., Kruse, R. and Runkler, T. (1999). *Fuzzy Cluster Analysis: Meth-*

ods for Classification, Data Analysis and Image Recognition, New Jersey, USA, John Wiley & Sons.

Horowitz, E. and Sahni, S. (1978). *Fundamentals of Computer Algoritms*, Maryland, USA, Computer Science Press.

Howard, R. A. (1971). *Dynamic Probabilistic Systems, Volumes 1 and 2*, New York, USA, Wiley.

Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J. (1992). Neural networks for control systems – a survey, *Automatica*, **28**(6), 1083–1112.

Hwang, C.-L. and Masud, A. S. M. (1979). *Multiple Objective Decision Making: methods and applications*, Vol. 164 of *Lecture Notes in Economics and Mathematical Systems*, Berlin, Springer-Verlag.

Ibaraki, T. (1976). Computational efficiency of approximate branch-and-bound algorithms, *International Journal of Computer and Information Sciences*, **5**(4), 315–344.

Ioannou, P. and Sun, J. (1996). *Robust Adaptive Control*, New Jersey, Prentice Hall.

Ishii, K. and Sugeno, M. (1985). A model of human evaluation process using fuzzy measure, *International Journal of Man-Machine Studies*, **22**, 19–38.

Jamshidi, M., Titli, A., Zadeh, L. and Boverie, S. (eds) (1997). *Applications of Fuzzy Logic*, New Jersey, Prentice Hall.

Jang, J. S. R. (1992). Self-learning fuzzy controllers based on temporal back propagation, *IEEE Transactions on Neural Networks*, **3**(5), 714–723.

Janis, I. L. and Mann, L. (1977). *Decision Making: A Psychological Analysis of Conflict, Choice and Commitment*, New York, Free Press.

Jia, L.-M. and Zhang, X.-D. (1993). On fuzzy multiobjective optimal control, *Engineering Applications of AI*, **2**(6), 153–164.

Johansen, T. A. (1994). Fuzzy model based control: stability, robustness, and performance issues, *IEEE Transactions on Fuzzy Systems*, **2**, 221–234.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: supervised learning with a distal teacher, *Cognitive Science*, **16**(3), 307–354.

Kacprzyk, J. (1978). A branch-and-bound algorithm for the multistage control of a nonfuzzy system in a fuzzy environment, *Control and Cybernetics*, **7**, 51–64.

Kacprzyk, J. (1995). A genetic algorithm for the multistage control of a fuzzy system in a fuzzy environment, *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, pp. 1083–1088.

Kacprzyk, J. (1997). *Multistage Fuzzy Control: a model-based approach to fuzzy control and decision making*, Chichester, Wiley.

Kacprzyk, J., Safteruk, K. and Staniewsky, P. (1981). Control of a stochastic system in a fuzzy environment over an infinite horizon in a non-Bellman-Zadeh setting, *Proceedings First Polish–Finnish Symposium on Systems Analysis and Its Applications, Part I*, Warsaw, pp. 79–91.

Kacprzyk, J. and Staniewsky, P. (1980). A new approach to the control of stochastic systems in a fuzzy environment, *Archiwum Automatyki i Telemechaniki*, **25**, 433–443.

Kandel, A. and Byatt, W. J. (1978). Fuzzy sets, fuzzy algebra, and fuzzy statistics, *Proceedings of the IEEE*, **66**, 1619–1639.

Kaymak, U. and Babuška, R. (1995). Compatible cluster merging for fuzzy modelling, *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems*, Vol. 2, Yokohama, Japan, pp. 897–904.

Kaymak, U., Babuška, R., Setnes, M., Verbruggen, H. B. and van Nauta Lemke, H. R.

(1997). Methods for simplification of fuzzy models, *in* D. Ruan (ed.), *Intelligent Hybrid Systems*, Kluwer Academic Publishers, Boston, pp. 91–108.

Kaymak, U., Babuška, R., van Nauta Lemke, H. R. and Honderd, G. (1998). A fuzzy logic system for steady–state security analysis of power networks, *Journal of Intelligent and Fuzzy Systems*, **6**, 345–359.

Kaymak, U. and Setnes, M. (2000). Extended fuzzy clustering algorithms, *ERIM report series Research in Management*, Erasmus Research Institute of Management, Erasmus University Rotterdam, Netherlands. ERS-2000-51-LIS, www.eur.nl/WebDOC/doc/erim/erimrs20001123094510.pdf.

Kaymak, U. and Sousa, J. M. (1997). Model based fuzzy predictive control applied to a simulated gantry crane, *Proceedings of Second Asian Control Conference, ASCC'97*, Vol. III, Seoul, Korea, pp. 455–458.

Kaymak, U. and van Nauta Lemke, H. R. (1993). A parametric generalized goal function for fuzzy decision making with unequally weighted objectives, *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 1156–1160.

Kaymak, U. and van Nauta Lemke, H. R. (1998). A sensitivity analysis approach to introducing weight factors into decision functions in fuzzy multicriteria decision making, *Fuzzy Sets and Systems*, **97**(2), 169–182.

Kaymak, U., van Nauta Lemke, H. R. and Bruijn, P. M. (1996). Using decision functions for nonlinear controller design, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, New Orleans, USA, pp. 1788–1793.

Kaymak, U., van Nauta Lemke, H. R. and den Boer, T. (1998). A sensitivity-based analysis of weighted fuzzy aggregation, *Proceedings of the IEEE World Congress on Computational Intelligence*, Anchorage, Alaska, pp. 755–760.

Keeney, R. L. and Raiffa, H. (1976). *Decision with Multiple Objectives: preferences and value tradeoffs*, New York, Wiley.

Kim, K. and Park, K. S. (1990). Ranking fuzzy numbers with index of optimism, *Fuzzy Sets and Systems*, **35**, 143–150.

Klawonn, F., Kinzel, J. and Kruse, R. (1994). Modification of genetic algorithms for designing and optimizing fuzzy controllers, *Proceedings of the First IEEE Conference on Evolutionary Computation*, Orlando, U.S.A., pp. 28–33.

Klir, G. J. and Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: theory and applications*, Upper Saddle River, Prentice Hall.

Kolmogorov, A. (1968). Logical basis for information theory and probability, *IEEE Transactions on Information Theory*, **5**(II–14), 662–664.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*, Englewood Cliffs, Prentice-Hall.

Kosko, B. (1994). Fuzzy systems as universal approximators, *IEEE Transactions on Computers*, **43**, 1329–1333.

Kosko, B. (1997). *Fuzzy Engineering*, Upper Saddle River, New Jersey, Prentice Hall.

Krishnapuram, R. and Freg, C.-P. (1992). Fitting an unknown number of lines and planes to image data through compatible cluster merging, *Pattern Recognition*, **25**(4), 385–400.

Kuipers, B. and Åström, K. (1994). The composition and validation of heterogeneous control laws, *Automatica*, **30**(2), 233–249.

Lee, C. C. (1990a). Fuzzy logic in control systems: fuzzy logic controller — part I, *IEEE Transactions on Systems, Man and Cybernetics*, **20**(2), 404–418.

Lee, C. C. (1990b). Fuzzy logic in control systems: fuzzy logic controller — part II, *IEEE*

Transactions on Systems, Man and Cybernetics, **20**(2), 419–435.

Leinfellner, W. and Köhler, E. (eds) (1998). *Game Theory, Experience, Rationality*, Dordrecht, the Netherlands, Kluwer Academic Publishers.

Levenhagen, J. and Spethmann, D. (1993). *HVAC Controls and Systems*, New York, McGraw Hill.

Ling, K. V. and Dexter, A. L. (1994). Expert control of air-conditioning plant, *Automatica*, **30**(5), 761–773.

Linkens, D. A. and Mahfouf, M. (1994). Generalized predictive control in clinical anaesthesia, *Advances in Model-Based Predictive Control*, Oxford University Press.

Linkens, D. A. and Nyongesa, H. O. (1995). Genetic algorithms for fuzzy control – part 1 and 2: offline system development and application, *IEE Proceedings on Control Theory Applications*, **142**(3), 161–185.

Liou, T.-S. and Wang, M.-J. J. (1992). Ranking fuzzy numbers with integral value, *Fuzzy Sets and Systems*, **50**, 247–255.

Ljung, L. (1987). *System Identification: Theory for the User*, New Jersey, Prentice-Hall.

Luce, R. D. and Raiffa, H. (1957). *Games and Decisions*, New York, John Wiley.

Mamdani, E. H. (1974). Applications of fuzzy algorithms for control of simple dynamic plant, *Proceedings IEE*, number 121, pp. 1585–1588.

Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic systems, *IEEE Transactions on Computers*, **26**(12), 1182–1191.

Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, **7**, 1–13.

Mandić, N. J. and Mamdani, E. H. (1984). A multi-attribute decison-making model with fuzzy rule-based modification of priorities, *in* H.-J. Zimmermann, L. A. Zadeh and B. R. Gaines (eds), *Fuzzy Sets and Decision Analysis*, Vol. 20 of *Studies in the Management Sciences*, North Holland, Amsterdam, pp. 285–306.

Mendel, J. M. (2000). *Uncertain Rule-Based Fuzzy Logic Systems: introduction and new directions*, New York, Prentice Hall.

Mesiar, R. (1995). Compensatory operators based on triangular norms and conorms, *Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing*, Vol. 1, Aachen, Germany, pp. 131–135.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin, Springer.

Mitten, L. G. (1970). Branch-and-bound methods: general formulation and properties, *Journal of Operations Research*, **18**, 24–34.

Mizumoto, M. (1989a). Pictorial representations of fuzzy connectives, part I: cases of t-norms, t-conorms and averaging operators, *Fuzzy Sets and Systems*, **31**, 217–242.

Mizumoto, M. (1989b). Pictorial representations of fuzzy connectives, part II: cases of compensatory operators and self-dual operators, *Fuzzy Sets and Systems*, **32**, 45–79.

Murofushi, T. and Sugeno, M. (1991). A theory of fuzzy measures: representation, the Choquet integral and null sets, *Journal of Mathematical Analysis and Applications*, **159**(2), 532–549.

Murray-Smith, R. and Johansen, T. A. (eds) (1997). *Multiple Model Approaches to Nonlinear Modeling and Control*, London, UK, Taylor & Francis.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization, *Computer Journal*, **7**, 308–313.

Nelles, O. (1998). Structure optimization of takagi–sugeno fuzzy models, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **6**(2), 161–170.

Onnen, C., Babuška, R., Kaymak, U., Sousa, J. M., Verbruggen, H. B. and Isermann, R. (1997). Genetic algorithms for optimization in predictive control, *Control Engineering Practice*, **5**(10), 1363–1372.

Palm, R., Driankov, D. and Hellendoorn, H. (1997). *Model Based Fuzzy Control*, Berlin, Springer Verlag.

Patton, R. J., Frank, P. and Clark, R. (1989). *Fault Diagnosis in Dynamic Systems: Theory and Applications*, New Jersey, USA, Prentice Hall.

Pedrycz, W. (1985). Applications of fuzzy relational equations for methods of reasoning in presence of fuzzy data, *Fuzzy Sets and Systems*, **16**, 163–175.

Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems*, second edn, New York, John Wiley.

Plataniotis, K. N., Androutsos, D. and Venetsanopoulos, A. N. (1996). Fuzzy adaptive filters for multichannel image processing, *Signal Processing*, **55**(1), 93–106.

Potts, J. C., Giddens, T. D. and Yadav, S. B. (1994). The development and evaluation of an improved genetic algorithm based on migration and artificial selection, *IEEE Transactions on Systems, Man and Cybernetics*, **24**(1), 73–86.

Procyk, T. J. and Mamdani, E. H. (n.d.). A linguistic self-organising process controller, *Automatica*, **15**, 15–30.

Richalet, J. (1993). Industrial applications of model based predictive control, *Automatica*, **29**, 1251–1274.

Richalet, J., Rault, A., Testud, J. L. and Papon, J. (1978). Model predictive heuristic control: applications to industrial processes, *Automatica*, **14**(5), 413–428.

Rosenmüller, J. (2000). *Game Theory*, C: Game Theory, Mathematical Programming and Operations Research, Boston, Kluwer Academic Publishers.

Roubos, H. and Setnes, M. (2001). Compact and transparent fuzzy models and classifiers through iterative complexity reduction, *IEEE Transactions on Fuzzy Systems*, **9**(4), 516–524.

Roubos, J. A., Mollov, S., Babuška, R. and Verbruggen, H. B. (1999). Fuzzy model-based predictive control using takagi–sugeno models, *International Journal of Approximate Reasoning*, **22**(1/2), 3–30.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning internal representations by error propagation, *in* D. E. Rumelhart and J. L. McClelland (eds), *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA.

Runkler, T. A. and Glesner, M. (1993). Defuzzification with improved static and dynamic behaviour: extended center of area, *Proceedings of the First European Congress on Fuzzy and Intelligent Technologies*, Vol. 2, pp. 845–851.

Sakawa, Y. and Shindo, Y. (1982). Optimal control of container cranes, *Automatica*, **18**(3), 257–266.

Saridis, G. N. (1977). *Self-organizing Controls of Stochastic Systems*, New York, Marcel Dekker.

Schram, G., Gopisetty, S. M. and Stengel, R. F. (1998). A fuzzy logic - parity space approach to actuator failure detection and identification, *Proceedings of 36th Aerospace Sciences Meeting & Exhibit*, Reno, NV, pp. AIAA Report No. 98–1014.

Seber, G. A. F. and Wild, C. J. (1989). *Nonlinear Regression*, New York, John Wiley & Sons.

Setnes, M., Babuška, R., Kaymak, U. and van Nauta Lemke, H. R. (1998). Similarity measures in fuzzy rule base simplification, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **28**(3), 376–386.

Setnes, M., van Nauta Lemke, H. R. and Kaymak, U. (1998). Fuzzy arithmetic-based interpolative reasoning for nonlinear dynamic fuzzy systems, *Engineering Applications of AI*, **11**(6), 781–789.

Shoureshi, R. and Rahmani, K. (1992). Derivation and application of an expert fuzzy optimal control system, *Fuzzy Sets and Systems*, **49**(2), 93–101.

Slotine, J. J. E. and Li, W. (1991). *Applied Nonlinear Control*, New Jersey, USA, Prentice Hall.

Soeterboek, R. (1992). *Predictive Control: a unified approach*, New York, Prentice Hall.

Sousa, J. M. (2000). Optimization issues in predictive control with fuzzy objective functions, *International Journal of Intelligent Systems*, **15**(9), 879–899.

Sousa, J. M., Babuška, R., Bruijn, P. M. and Verbruggen, H. B. (1996). Fuzzy compensation of steady-state errors in nonlinear control systems, *Proceedings of Fourth European Congress on Fuzzy and Intelligent Technologies, EUFIT'96*, Aachen, Germany, pp. 946–950.

Sousa, J. M., Babuška, R. and Verbruggen, H. B. (1995). Adaptive fuzzy model-based control, *Proceedings Third European Congress on Fuzzy and Intelligent Technologies*, Aachen, Germany, pp. 865–869.

Sousa, J. M., Babuška, R. and Verbruggen, H. B. (1997). Branch-and-bound optimization in fuzzy predictive control: an application to an air-conditioning system, *Control Engineering Practice*, **5**(10), 1395–1406.

Sousa, J. M. and Kaymak, U. (2001). Model predictive control using fuzzy decision functions, *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, **31**(1), 54–65.

Sousa, J. M., Kaymak, U. and Verbruggen, H. (1999). Performance criteria: Classical and fuzzy design, *in* H. Verbruggen, H. Zimmermann and R. Babuška (eds), *Fuzzy Algorithms for Control*, Kluwer Academic Publishers, Boston, pp. 159–183.

Sousa, J. M., Kaymak, U., Verhaegen, M. and Verbruggen, H. B. (1996). Convex optimization in fuzzy predictive control, *Proceedings of CDC'96 - 35th IEEE Conference on Decision and Control*, Kobe, Japan, pp. 2735–2740.

Sousa, J. M. and Setnes, M. (1999). Fuzzy predictive filters in model predictive control, *IEEE Transactions on Industrial Electronics*, **46**(6), 1225–1232.

Stein, W. E. (1980). Optimal stopping in a fuzzy environment, *Fuzzy Sets and Systems*, **3**, 253–259.

Su, C. Y., Stepanenko, Y., Tang, S. and Han, H. (1999). Adaptive robust controller design and implementation for electrically driven robots, *in* O. Kaynak, S. Tosunoglu and M. H. A. Jr. (eds), *Recent Advances in Mechatronics*, Springer, pp. 115–126.

Sugeno, M. (1974). *Theory of Fuzzy Integrals and Its Applications*, Ph.D. thesis, Tokyo Institute of Technology, Tokyo.

Sugeno, M. (ed.) (1985). *Industrial Applications of Fuzzy Control*, Amsterdam, The Netherlands, Elsevier Science Publishers B.V. (North-Holland).

Sugeno, M. and Kang, G. T. (1988). Structure identification of fuzzy model, *Fuzzy Sets and Systems*, **28**, 15–33.

Sugeno, M. and Takagi, T. (1983). A new approach to design of fuzzy controller, *in* P. P. Wang (ed.), *Advances in Fuzzy Sets, Possibility Theory and Applications*, Plenum Press, New York, pp. 325–334.

Sugeno, M. and Tanaka, K. (1991). Successive identification of a fuzzy model and its application to prediction of a complex system, *Fuzzy Sets and Systems*, **42**, 315–334.

Sugeno, M. and Yasukawa, T. (1994). Qualitative modelling based on numerical data and knowledge data and its application to control, *in* J. M. Zurada, R. J. M. II and C. J. Robinson (eds), *Computational Intelligence: Imitating Life*, IEEE Press, New York, pp. 304–315.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control, *IEEE Transactions on Systems, Man and Cybernetics*, **15**(1), 116–132.

Tanaka, K. and Sugeno, M. (1992). Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems*, **45**(2), 135–156.

Tang, Y., Tomizuka, M., Gerrerro, G. and Montemayor, G. (2000). Decentralized robust control of mechanical systems, *IEEE Transactions on Automatic Control*, **45**, 771–776.

Terano, T., Asai, K. and Sugeno, M. (1994). *Applied Fuzzy Systems*, London, Academic Press.

Thole, U., Zimmermann, H. J. and Zysno, P. (1979). On the suitability of minimum and product operators for the intersection of fuzzy sets, *Fuzzy Sets and Systems*, **2**, 167–180.

Tseng, T. Y. and Klein, C. M. (1989). New algorithm for the ranking procedure in fuzzy decision making, *IEEE Transactions on Systems, Man and Cybernetics*, **19**(5), 1289–1296.

Türkşen, I. B. (1999). Type I and type II fuzzy system modeling, *Fuzzy Sets and Systems*, **106**(1), 11–34.

van Can, H. J. L., te Braake, H. A. B., Hellinga, C., Krijgsman, A. J., Verbruggen, H. B., Luyben, K. C. A. M. and Heijnen, J. J. (1995). Design and real-time testing of a neural model predictive controller for a nonlinear system, *Chemical Engineering Science*, **50**, 2419–2430.

van Nauta Lemke, H. R., Dijkman, J. G., van Haeringen, H. and Pleeging, M. (1983). A characteristic optimism factor in fuzzy decision-making, *Proc. IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, Marseille, France, pp. 283–288.

Verbruggen, H. B. and Bruijn, P. M. (1999). Fuzzy systems in control engineering, *in* H. B. Verbruggen, H.-J. Zimmermann and R. Babuška (eds), *Fuzzy Algorithms for Control*, International Series in Intelligent Technologies, Kluwer Academic Publishers, Dordrecht, pp. 3–15.

Veres, S. M. and Sokolov, V. F. (1998). Adaptive robust control under unknown plant orders, *Automatica*, **34**, 723–730.

von Winterfeldt, D. and Edwards, W. (1986). *Decision Analysis and Behavioral Research*, London, Cambridge University Press.

Wang, L. X. (1992). Fuzzy systems are universal approximators, *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, USA, pp. 1163–1170.

Wang, L. X. and Mendel, J. M. (1993). Fuzzy adaptive filters with application to nonlinear channel equalization, *IEEE Transactions on Fuzzy Systems*, **1**(3), 161–170.

Weber, S. (1983). A general concept of fuzzy connectives, negations and implications based on t-norms and t-co-norms, *Fuzzy Sets and Systems*, **11**, 115–134.

Yager, R. R. (1978). Fuzzy decision making including unequal objectives, *Fuzzy Sets and Systems*, **1**, 87–95.

Yager, R. R. (1980). On a general class of fuzzy connectives, *Fuzzy Sets and Systems*, **4**, 235–242.

Yager, R. R. (1984). General multiple-objective decision functions and linguistically quantified statements, *International Journal of Man-Machine Studies*, **21**, 389–400.

Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking, *IEEE Transactions on Systems, Man and Cybernetics*, **18**(1), 183–190.

Yager, R. R. (1992). On the inclusion of importance in multi-criteria decision making in the fuzzy set framework, *International Journal of Expert Systems, Research and Applications*, **5**, 211–228.

Yager, R. R. (1994). Aggregation operators and fuzzy systems modeling, *Fuzzy Sets and Systems*, **67**, 129–145.

Yager, R. R. and Filev, D. P. (1993). SLIDE: a simple adaptive defuzzification method, *IEEE Transactions on Fuzzy Systems*, **1**(1), 69–78.

Yager, R. R. and Filev, D. P. (1994). *Essentials of Fuzzy Modelling and Control*, New York, John Wiley.

Yager, R. R. and Rybalov, A. (1995). Full reinforcement operators in aggregation techniques, *Technical report # mii-1502*, Machine Intelligence Institute, Iona College, New Rochelle, New York.

Yang, M. S. (1993). A survey of fuzzy clustering, *Mathematical and Computer Modelling*, **18**(11), 1–16.

Yasunobu, S. and Miyamoto, S. (1985). Automatic train operation by predictive fuzzy control, *in* M. Sugeno (ed.), *Industrial Application of Fuzzy Control*, North Holland, Amsterdam, pp. 1–18.

Yen, J., Langari, R. and Zadeh, L. A. (1995). *Industrial Applications of Fuzzy Logic and Intelligent Systems*, New York, USA, IEEE Press.

Yoshinari, Y., Pedrycz, W. and Hirota, K. (1993). Construction of fuzzy models through clustering techniques, *Fuzzy Sets and Systems*, **54**, 157–165.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, **3**, 28–44.

Zeng, X. J. and Singh, M. G. (1995). Approximation theory of fuzzy systems – MIMO case, *IEEE Transactions on Fuzzy Systems*, **3**(2), 219–235.

Zeng, X.-J. and Singh, M. G. (1996a). Approximation properties of fuzzy systems generation by the min inference, *IEEE Transactions on Systems, Man and Cybernetics*, **26**(1), 187–193.

Zeng, X.-J. and Singh, M. G. (1996b). Decomposition property of fuzzy systems and its applications, *IEEE Transactions on Fuzzy Systems*, **4**(2), 149–165.

Zhao, J., Gorez, R. and Wertz, V. (1997). Synthesis of fuzzy control systems based on linear Takagi–Sugeno fuzzy models, *in* R. Murray-Smith and T. A. Johansen (eds), *Multiple Model Approaches to Nonlinear Modeling and Control*, Taylor & Francis, London, UK, pp. 307–336.

Zhou, K., Doyle, J. C. and Glover, K. (1996). *Robust and Optimal Control*, New Jersey, USA, Prentice Hall.

Zimmermann, H. J. (1978). Results of empirical studies in fuzzy set theory, *in* G. J. Klir (ed.), *Applied General Systems Research: Recent Developments*, Plenum Press, New

York, pp. 303–312.

Zimmermann, H. J. (1987). *Fuzzy Sets, Decision Making and Expert Systems*, Boston, Kluwer Academic Publishers.

Zimmermann, H. J. (1996). *Fuzzy Set Theory and its Application*, third edn, Boston, Kluwer.

Zimmermann, H. J. and Zysno, P. (1980). Latent connectives in human decision making, *Fuzzy Sets and Systems*, **4**, 37–51.

Zurada, J. M., Marks II, R. J. and Robinson, C. J. (eds) (1994). *Computational Intelligence: imitating life*, IEEE Press, Piscataway, NJ, pp. v–xi.

Index

World Scientific Series in Robotics and Intelligent Systems – Vol. 27

FUZZY DECISION MAKING IN MODELING AND CONTROL

Decision making and control are two fields with distinct methods for solving problems, and yet they are closely related. This book bridges the gap between decision making and control in the field of fuzzy decisions and fuzzy control, and discusses various ways in which fuzzy decision making methods can be applied to systems modeling and control.

Fuzzy decision making is a powerful paradigm for dealing with human expert knowledge when one is designing fuzzy model-based controllers. The combination of fuzzy decision making and fuzzy control in this book can lead to novel control schemes that improve the existing controllers in various ways. The following applications of fuzzy decision making methods for designing control systems are considered:

- *Fuzzy decision making for enhancing fuzzy modeling.* The values of important parameters in fuzzy modeling algorithms are selected by using fuzzy decision making.
- *Fuzzy decision making for designing signal-based fuzzy controllers.* The controller mappings and the defuzzification steps can be obtained by decision making methods.
- *Fuzzy design and performance specifications in model-based control.* Fuzzy constraints and fuzzy goals are used.
- *Design of model-based controllers combined with fuzzy decision modules.* Human operator experience is incorporated for the performance specification in model-based control.

The advantages of bringing together fuzzy control and fuzzy decision making are shown with multiple examples from real and simulated control systems.

ISBN 981-02-4877-6

9 789810 248772

www.worldscientific.com
4900 hc