Stellingen

behorende bij het proefschrift

A Fuzzy Approach to Model-Based Control

 João Miguel da Costa Sousa
Delft, 20 april 1998
1. Zowel “Impression, soleil levant” van Monet als de toepassing van vage logica laten zien dat het gebruik van vaagheid de perceptie kan verrijken.

2. Vage model-gebaseerde regeling heeft voordelen ten op zichte van een op vage logica gebaseerde regeling die uitgaat van de kennis van experts, indien wordt gekeken naar de prestatie en de flexibiliteit van de gekozen regeling (dit proefschrift).

3. Kacprzyk stelt “We consider (...) open-loop control. Unfortunately, not much is know about closed-loop (feedback) control in a fuzzy environment in the optimal-control-type Bellman and Zadeh setting [1].” Een grote vooruitgang in prestatie kan slechts worden bereikt indien in toepassingen wordt uitgegaan van een gesloten-lus regeling (dit proefschrift).

4. Voorspellend regelen met gebruik making van vage doelfuncties kan de prestatie van het geregeld systeem verbeteren ten op zichte van het gebruik van klassieke doelfuncties. Dit geldt zelfs voor lineaire systemen (dit proefschrift).

5. Conceptueel is er geen verschil tussen vage doelen en vage begrenzingen bij vage besluitvorming. Bij vage model-gebaseerde regeling moeten de vage begrenzingen echter ook de harde begrenzingen bevatten van het geregeld systeem (dit proefschrift).

6. In de meeste regelproblemen worden “goede” oplossingen geprefereerd boven “optimale” oplossingen vanwege de sterk toenemende rekenkosten die verbonden zijn aan het vinden van deze optimale oplossing.

7. Heden ten dage, wordt het academische belang vaak beoordeeld aan de hand van contacten en contracten en wordt minder gekeken naar de kwaliteit van het wetenschappelijk werk.

8. De belangrijkste vraag die momenteel onder onderzoekers wordt gesteld luidt eerder “Hoeveel artikelen heb je onderhanden?” dan “Met welke onderzoek ben je momenteel bezig?”


10. Een nederlands verjaardagsfeest is eenigzins saai en kan een wat levendiger karakter verkrijgen door de aanwezigheid van een Zuid-Europeaan.

11. Ofschoon zij onderzoek doen op het gebied van intelligent regelen worden onderzoekers in het gebouw van Elektrotechniek nog steeds geconfronteerd met een zogenaamde temperatuur regeling.

12. De belangrijkste reden waarom het Portugees een wereldtaal is en het Nederlands niet, wordt veroorzaakt door het feit dat de Portugezen ook hun cultuur en taal naar hun ex-colonies hebben overgebracht, terwijl de Nederlanders daar meestal waren om economische redenen.

1. “Impression, soleil levant” from Monet and fuzzy logic show that a certain vagueness enriches your perception.

2. Fuzzy model-based control presents advantages over the fuzzy logic control based on expert knowledge, in terms of performance and flexibility of the chosen controller (this thesis).

3. Kacprzyk states “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 setting [1].” A major progress in terms of performance can only be achieved if the applications will be on close-loop control (this thesis).

4. Predictive control using fuzzy objective functions can improve the control performance when compared to classical objective functions, even for linear systems (this thesis).

5. Fuzzy goals and fuzzy constraints are conceptually equal in fuzzy decision making. In fuzzy model-based control, however, the fuzzy constraints should also express the 'hard' constraints of the system under control (this thesis).

6. In control problems “good” solutions are preferred over “optimal” solutions due to the disproportionate computational cost associated with the process of finding the latter.

7. Nowadays, academic importance is many times judged by contacts and contracts, and not by the quality of the scientific work.

8. The present-day key question among researchers is not “What are you investigating at the moment?” , but instead, “How many papers do you have to finish?”

9. Small countries in Europe, especially with old borders and one main language, have some remarkable features like tolerance, the use of subtitles, and lack of extreme nationalistic feelings, that hardly can be found in large countries.

10. A Dutch birthday party needs some excitement, obtained e.g. by the attendance of a South European, to become more lively.

11. Although working in intelligent control, researchers still have to suffer the so-called control of the temperature in the Electrical Engineering building.

12. The main reason why the Portuguese is a world language and the Dutch not, is because the Portuguese exported also their culture and language to the ex-colonies, while the Dutch were there mainly for economical reasons.
3371
015-213 7521
18:00 uur Dinsdag ET W12.230
A Fuzzy Approach to Model-Based Control

João Miguel da Costa Sousa
A Fuzzy Approach to Model-Based Control

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,
op maandag 20 april 1998 te 16:00 uur
door

João Miguel da COSTA SOUSA

Mestre em Engenharia Mecânica pela Universidade Técnica de Lisboa,
geboren te Lissabon, Portugal
Dit proefschrift is goedgekeurd door de promotor:
Prof. ir. H.B. Verbruggen

Samenstelling promotiecommissie:
Rector Magnificus,
Prof. ir. H.B. Verbruggen,
Prof. dr. J. Sá da Costa,
Prof. ir. H. van der Ree,
Prof. dr. ir. A.C.P.M. Backx,
Prof. dr. ir. G. van Straten,
Prof. dr. H. Koppelaar,
Ir. P.M. Bruijn,
Technische Universiteit Delft, voorzitter
Technische Universiteit Delft, promotor
Technical University of Lisbon
Technische Universiteit Delft
Technische Universiteit Eindhoven
Landbouwuniversiteit Wageningen
Technische Universiteit Delft
Technische Universiteit Delft

Ir. P.M. Bruijn heeft als begeleider in belangrijke mate aan het totstandkomen van het proefschrift bijgedragen.

ISBN 90-9011506-4

Front cover: Detail of 34 (tile panel). Stocklein. Printed by permission of Galerias Ratton Cerámiccas, Lisbon, Portugal.


No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.
## Contents

1 Introduction .......................... 1
   1.1 Classical Design of a Control System ............. 1
       1.1.1 Regulation .................................. 3
       1.1.2 Tracking .................................. 3
       1.1.3 Steps of control design .................. 4
   1.2 Conventional Fuzzy Control .................. 5
   1.3 Fuzzy model-based control ................. 7
       1.3.1 Modeling and identification ............ 8
       1.3.2 Design specifications .................. 9
       1.3.3 Types of controllers .................... 9
       1.3.4 Research topics developed in this thesis .... 10
   1.4 Outline of the thesis ..................... 11

2 Modeling and Identification ............. 13
   2.1 Formulation of the modeling problem .......... 14
   2.2 Fuzzy modeling ................................ 15
       2.2.1 Linguistic fuzzy models .................. 17
       2.2.2 Singleton fuzzy model ................... 19
       2.2.3 Takagi-Sugeno fuzzy models ............... 19
   2.3 Fuzzy identification .......................... 21
   2.4 Identification by product-space fuzzy clustering .... 21
       2.4.1 Structure identification .................. 22
       2.4.2 Parameter estimation ..................... 23
   2.5 Summary and concluding remarks ............ 25

3 Performance Criteria .................. 27
   3.1 Design specifications ..................... 28
3.1.1 Design specifications for linear systems \hspace{2cm} 28
3.1.2 Design specifications for nonlinear systems \hspace{2cm} 31
3.2 Classical performance specifications \hspace{2cm} 32
  3.2.1 I/O specifications \hspace{2cm} 32
  3.2.2 Regulation specifications \hspace{2cm} 34
  3.2.3 Actuator effort \hspace{2cm} 35
3.3 Classical performance criteria \hspace{2cm} 35
  3.3.1 Norms and semi-norms of signals \hspace{2cm} 35
  3.3.2 Norms of systems \hspace{2cm} 38
3.4 Fuzzy performance criteria \hspace{2cm} 39
3.5 Summary and concluding remarks \hspace{2cm} 41

4 Types of controllers \hspace{2cm} 43
  4.1 Model-Based Predictive Control \hspace{2cm} 44
    4.1.1 Basic definitions \hspace{2cm} 45
    4.1.2 Modeling in MBPC \hspace{2cm} 48
    4.1.3 Optimization problems \hspace{2cm} 48
    4.1.4 Compensation of model-plant mismatch and disturbances \hspace{2cm} 49
  4.2 Nonlinear Internal Model Control \hspace{2cm} 49
    4.2.1 Classical internal model control \hspace{2cm} 49
    4.2.2 MBPC in an internal model control scheme \hspace{2cm} 51
  4.3 Fuzzy compensation of steady-state errors \hspace{2cm} 52
    4.3.1 Fuzzy compensation \hspace{2cm} 53
    4.3.2 Application: system with a dead-zone \hspace{2cm} 55
  4.4 Summary and concluding remarks \hspace{2cm} 57

5 Fuzzy Model-Based Control Using Fuzzy Models \hspace{2cm} 59
  5.1 Inversion of fuzzy models \hspace{2cm} 60
    5.1.1 Problem definition \hspace{2cm} 61
    5.1.2 Inversion methods \hspace{2cm} 63
    5.1.3 Inversion of a singleton fuzzy model \hspace{2cm} 64
    5.1.4 Inversion of an affine Takagi–Sugeno fuzzy model \hspace{2cm} 71
    5.1.5 On-line adaptation of feedforward fuzzy models \hspace{2cm} 75
  5.2 Predictive control using the inversion of a fuzzy model \hspace{2cm} 77
  5.3 Example: pressure control of a fermentation tank \hspace{2cm} 79
    5.3.1 Fuzzy modeling \hspace{2cm} 80
    5.3.2 Predictive control based on the singleton fuzzy model \hspace{2cm} 83
    5.3.3 Predictive control based on the affine TS fuzzy model \hspace{2cm} 86
  5.4 Summary and concluding remarks \hspace{2cm} 88

6 Fuzzy Model-Based Control with Fuzzy Decision Functions \hspace{2cm} 89
  6.1 Fuzzy decision making \hspace{2cm} 90
    6.1.1 Fuzzy goal, fuzzy constraint and fuzzy decision \hspace{2cm} 91
6.1.2 Multiple fuzzy goals and/or fuzzy constraints .... 94
6.1.3 Fuzzy goals and fuzzy constraints defined in different spaces .... 96
6.2 Fuzzy decision making in control .... 101
  6.2.1 Fuzzy goals and fuzzy constraints in the control environment .... 104
  6.2.2 Aggregation of criteria in the control environment .... 105
  6.2.3 Fuzzy criteria in model-based predictive control .... 107
6.3 Fuzzy criteria for FDM in control .... 108
  6.3.1 Control criteria and decision functions .... 108
  6.3.2 Description of the simulated systems .... 112
  6.3.3 Obtained results .... 115
6.4 Choice of aggregation operators for FDM in control .... 119
  6.4.1 Example of a linear system .... 122
6.5 Summary and concluding remarks .... 124

7 Optimization Problems .... 127
  7.1 Branch-and-bound optimization for predictive control .... 128
    7.1.1 Application of B&G to predictive control .... 129
    7.1.2 Application of the B&G method to nonlinear control .... 132
    7.1.3 Some considerations on the B&G method applied to MBPC .... 134
  7.2 Genetic algorithms for optimization in predictive control .... 135
    7.2.1 Genetic algorithms .... 136
    7.2.2 Application of genetic algorithms to predictive control .... 139
    7.2.3 Application example .... 145
    7.2.4 Summary and discussion .... 146
  7.3 Convex optimization in fuzzy predictive control .... 148
    7.3.1 Algorithms for convex optimization .... 150
    7.3.2 Example of application .... 151
    7.3.3 Discussion .... 153
  7.4 Branch-and-bound optimization for predictive control with fuzzy decision functions .... 153
    7.4.1 Application example .... 157
    7.4.2 Discussion .... 160
  7.5 Summary and concluding Remarks .... 160

8 Application to an Air-Conditioning System .... 163
  8.1 Air-conditioning systems .... 164
  8.2 Description of the fan-coil system .... 165
  8.3 Fuzzy models of the air-conditioning system .... 167
    8.3.1 TS fuzzy model of the air-conditioning system .... 167
    8.3.2 Affine TS model of the air-conditioning system .... 171
  8.4 Controllers applied to the air-conditioning system .... 172
    8.4.1 PID control of the air-conditioning system .... 173
    8.4.2 Inverse control based on an affine TS fuzzy model .... 174
8.4.3 Predictive control based on classical cost functions ................................................ 174
8.4.4 Predictive control based on fuzzy cost functions ....................................................... 178
8.5 Concluding remarks ........................................................................................................ 180

9 Conclusions .......................................................................................................................... 181

9.1 Fuzzy model-based control .............................................................................................. 181
9.2 Predictive control using fuzzy models ............................................................................ 182
9.3 Predictive control with fuzzy objective functions ............................................................ 183
9.4 Optimization problems ...................................................................................................... 185
  9.4.1 Optimization algorithms for classical MBPC ............................................................... 185
  9.4.2 Optimization in MBPC with fuzzy objective functions ................................................ 186
9.5 Suggestions for future research ...................................................................................... 186

A Fuzzy sets – basic definitions ............................................................................................. 189

A.1 Basic definitions ............................................................................................................... 189
A.2 Operations on fuzzy sets .................................................................................................. 190
A.3 Fuzzy relations ................................................................................................................ 192
A.4 Projections and cylindrical extensions ............................................................................ 192

B Gustafson-Kessel algorithm ................................................................................................. 195

C Fuzzy models of the air-conditioning system ..................................................................... 197

  C.1 Takagi-Sugeno fuzzy model .......................................................................................... 197
     C.1.1 Sub-model for the supply temperature ................................................................... 197
     C.1.2 Sub-model for the mixed-air temperature ............................................................ 199
  C.2 Affine TS model of the air-conditioning system ............................................................. 200

Symbols and abbreviations .................................................................................................... 203

Summary .................................................................................................................................. 207

Samenvatting ............................................................................................................................ 211

Resumo .................................................................................................................................... 215

Curriculum Vitae ..................................................................................................................... 219

Acknowledgements .................................................................................................................. 221

References ................................................................................................................................. 223

Author index .............................................................................................................................. 235

Subject index ............................................................................................................................. 239
Introduction

The process of designing classical control systems is usually presented in a quite different way from the process of designing fuzzy control systems, which hinders the mutual influence between the two approaches. Therefore, this chapter begins by reviewing briefly classical control and fuzzy control design, as they are usually introduced in literature. The classical control design is frequently divided in two main design methods:

1. *signal-based control* – where measurements of various signals of the system are used to compute the control action directly without evaluating the process knowledge. A model of the system is only used during the design stage to determine the proper setting of the controller parameters.

2. *model-based control* – in which the parameters of the controller are optimized based on the following three items: a given desired response (or reference), a model of the process and performance criteria. Contrary to signal-based control, in model-based control the model of the system and the controller parameters are directly related, and the model is used at each sampling instant to predict the output of the system.

Conventional fuzzy controllers are based on expert knowledge extracted from experience in controlling the system, and are represented by If–Then rules. The next two sections present classical and fuzzy control design in more detail.

### 1.1 Classical Design of a Control System

A system, consisting of interconnected components, is generally built to achieve a desired purpose. The performance of this system can be examined, and methods to ameliorate its
performance can be proposed. In general, a system cannot achieve a specified level of performance without changing its static and dynamic character by adding control actions. Control systems are often based on the principle of feedback, whereby the signal to be controlled is compared to a desired reference signal and the difference is used to calculate corrective control actions.

![Block diagram of a feedback control scheme](image)

**Figure 1.1. Block diagram of a feedback control scheme**

Figure 1.1 shows a general feedback control scheme for multivariable systems. Various input signals (actions) influence the process $P$ resulting in output variables $y$. The input variables are usually divided in control actions or manipulated variables $u$ and system disturbances $d$, 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 $r$ can be seen as a goal to be achieved by the control system. The plant under control and the actuators manipulated by $u$ are included in the process. The sensors are represented by the operator $S$, having as inputs, the output variables from the process $y$ and the measurement disturbances $d_m$, generating the measured outputs $y_m$. The controller $C$ generates the control actions $u$ based on the received information: the measured outputs $y_m$, the references $r$ to be followed, the disturbances $d$ and $d_m$ if available, and the goals to be obtained.

The purpose of a control system is to keep the values of the output variables $y$ as close as possible to the respective target values $r$, by manipulating the control actions $u$, minimizing the effects of the disturbances $d$ and $d_m$ on the controlled system. Sometimes this goal is translated to a cost function, where a term related to the control effort can be added. The objective of a control system must be accomplished taking into account the dynamic behavior of the process $P$, and the static constraints, as e.g. the flow rate in a tube has its maximum value when a valve before that tube is fully open.

Control problems are usually divided in two main categories:

1. **Regulation** - In this type of problems, the controller, usually called a regulator, should keep the system on an operating point or setpoint.

2. **Tracking** - The controller is designed such that the system should follow a predefined and time-varying trajectory or reference.
The description of these types of control problems is given in the next two sections, following the definitions presented in Slotine and Li (1991) and Palm, et al. (1997). First, let the system under control be modeled by a nonlinear open loop discrete model of the form

\[ x(k + 1) = f(x(k), u(k), k), \]  

(1.1)

where \( f \) is a \( n \times 1 \) nonlinear function, \( x \) is the \( n \times 1 \) state vector, \( u \) is the \( m \times 1 \) input vector, and \( k \) denotes discrete time samples.

This general model is time-variant or non-autonomous. However, most of the models used in this thesis are autonomous, and are thus given by:

\[ x(k + 1) = f(x(k), u(k)). \]  

(1.2)

Note also that only discrete models are considered in this thesis. In fact, most of the present-day systems are described by a discrete or discretized model obtained by identification and parameter estimation techniques. Note however that the actual system is in most cases continuous, which impose a well-chosen sampling period.

### 1.1.1 Regulation

Consider that a nonlinear autonomous system is described by the open-loop equation given by (1.2) or by some approximation of the system under control given by some discrete model. The regulation control problem consists of finding a control law such that, starting anywhere in a region around 0, the state vector \( x \) of the closed loop system tends to 0, \( x \to 0 \), as \( k \) tends to infinity. When the control law depends directly on the measurement signals, it is said to be a static control law. If it depends on a dynamic function of the measured signals it is called a dynamic control law. If the objective of the control system is to stabilize the close loop system around some setpoint \( x_d \) different from 0, this problem can be easily transformed into the regulation of the system (1.2) around 0, by taking \( x - x_d \) as the new state vector.

### 1.1.2 Tracking

Consider now a nonlinear autonomous open loop system described by

\[ x(k + 1) = f(x(k), u(k)), \]

\[ y(k) = g(x(k)), \]  

(1.3)

and a desired output trajectory (reference) \( r \). The tracking problem consists of finding a control law such that starting from any initial state in a region around 0, the tracking error \( r - y \) tends to 0, while the state vector \( x(k) \) and the control actions \( u(k) \) remain bounded.
It is usual to assume that the desired reference \( r \) and its derivatives up to a sufficient order are continuous and bounded (Slotine and Li, 1991). It is also usually assumed that the desired reference and its derivatives are available beforehand. In this thesis these assumptions are not considered and the only requirement is to have the desired reference beforehand. Although more general, this procedure hampers the possibility of proving stability and robustness for the control system. However, by taking a more general approach, it is possible to tackle real problems that due to the assumptions usually made for tracking control, the classical approach cannot cope with.

Note that in general it is not possible to achieve perfect tracking, i.e., the system cannot follow the trajectory with zero error at all time. For step references, for instance, the objective is usually to keep the error as small as possible considering the constraints present in the system. A possible control strategy used to satisfy this specification is model-based predictive control, described in Section 4.1.

Usually, tracking problems are more difficult to solve. In fact, regulation problems can often be regarded as a special case of tracking problems where the desired trajectory remains constant.

### 1.1.3 Steps of control design

The process of designing a control system deals with the level of abstraction and the physics of the system (Bélanger, 1995). A possible procedure for the different steps involved in this design is the following:

1. Study of the system throughput, energy levels and desired accuracy in order to decide the types of sensors and actuators to be used and their proper insertion in the system.

2. Modeling of the resulting system (process including actuators, and sensors) and disturbances, if possible. The model can be obtained from first principles, or using an identification procedure. Modeling and identification of a system usually involves the simplification and validation of the obtained model.

3. Choice of the design specifications.

4. Selection and design of the controller to be used, meeting the desired specifications.

5. Simulation of the resulting controlled system and selection of possible steps to ameliorate the modeling and control.

6. Choice of the hardware and software to implement the control system.

7. Real-time experiments on the real process, and final tuning of some of the design and/or control parameters.
After step 5 it may be necessary to go back to step 1, and repeat the control design, due to poor performance of the resulting system. Moreover, the choice of hardware and software for the control system can be very critical for reasons concerning safety, costs, speed and performance, and it should preferably be done in an early stage, but at the moment no general, present-day tools are available to accomplish this task.

The design of a classical control system is generally quite different from the design of the usually called fuzzy controller (Driankov, et al., 1993; Zimmermann, 1996), which is called conventional fuzzy controller in this thesis. The next section presents conventional fuzzy control, showing the differences to the classical design of a control system.

1.2 Conventional Fuzzy Control

The objective of fuzzy logic control (FLC), or conventional fuzzy control, is to control complex processes using a knowledge-based control strategy based on 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 system is not available, not possible to obtain, or 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, such as driving a car, without any model, which cannot be easily controlled by conventional control systems. Thus, the control design problem in FLC is based on empirical knowledge regarding the behaviour of the process, instead of using a strictly analytic framework. This knowledge, cast into a linguistic or rule-based form, constitutes the base of a fuzzy logic control system.

FLC design 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 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 a lot”. The definition of all control rules, acquired from an expert, constitutes the first step in the design of a FLC.

The set of all control rules constitutes the rule base. Furthermore, data is 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. 1.2).

The fuzzification first maps the measured output variables of the process into the suitable range constituting the normalized universes of discourse used in the rule base, or, in other words, performs an input normalization. Sometimes a non-normalized domain is used and then this step is not necessary. After the crisp input values are normalized, the “real” fuzzification is performed, converting the values of process variables into linguistic values of fuzzy sets, making it compatible with the fuzzy set representation in the rule base.
The module rule base in Fig. 1.2 contains a relation described in fuzzy terms. The inference engine computes the appropriate control action according to the fuzzified inputs and the rule base. This operation is known as the compositional rule of inference (Zadeh, 1973).

The defuzzification translates the fuzzy outputs provided by the inference engine into a numerical (crisp) representation. Sometimes this values 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 defuzzification results in the control actions $u$. If the change in the control actions $\Delta u$ is generated instead, these signals must be integrated and a dynamic filter between the defuzzification block and the process must be included in the control scheme presented in Fig. 1.2.

The fuzzy rules contained in the conventional fuzzy controller do not include any dynamics. The dynamic behaviour 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 $r$ and the output $y$, 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. 1.2 does not include the disturbances explicitly, but they are usually present as shown in the general control scheme (Fig. 1.1). 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 contain implicitly 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 is usually difficult to express 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 "condition" of the operator. It is questionable whether the control actions performed by the operator are the most desirable ones, given the possible changes in the process behaviour.

1.3 Fuzzy model-based control

An alternative method for the design of fuzzy controllers is used in this thesis, where the design of fuzzy controllers follows closely 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 in this thesis. As the term fuzzy model-based control (FMBC) can have different meanings dependent on the context, in this thesis FMBC is defined 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 behaviour, design a feedback control law, such that the closed loop system behaves in the desired way, where the model, the design specifications and/or the developed controller can be classical or fuzzy.

This definition is rather broad, and several combinations of the different design components can be made. Considering that the model, the specifications and the controller can be conventional or fuzzy, there are eight possible combinations of these design components in FMBC. The explicit utilization of the fuzzy sets theory can be included in three distinct parts of controller synthesis: by using fuzzy models, defining fuzzy performance criteria as a confluence of fuzzy design specifications, and/or by developing fuzzy controllers. Note that this approach is not based on a trial-and-error approach, which makes the development of a fuzzy model-based controller more systematic, and, in general, less time-consuming.

First, fuzzy models are considered in the design of FMBC systems. Several authors presented controllers based on fuzzy models. Johansen (1994b), for instance, presented a model-based predictive controller using fuzzy models. A nonlinear controller based on a fuzzy model of MIMO dynamic systems is described and analyzed in Johansen (1994a). The 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 (1997). Some of this work was developed together with the author of this thesis, and is presented in Chapter 5. Palm, et al. (1997) presented 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 as fuzzy design specifications or fuzzy criteria was first introduced by Bellman and Zadeh (1970). The most recent work on fuzzy design specifications applied to control can be found in Kacprzyk (1997). Chapter 6 presents a survey on this subject.

As FMBC follows closely 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 and air-conditioned room, being decomposed afterwards in several hierarchical levels. For the given example, comfort has to be translated in 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 thesis. The selection of the different elements in the control design problem results in a wide range of possible combinations, such as: the (nonlinear) type of the model; conventional or fuzzy performance criteria; and different control structures, as for instance, model-based predictive control or internal model control. Therefore, some choices concerning the direction of research to be followed should be made a priori. The choices of modeling and identification techniques, performance criteria and types of controllers are presented in the following sections.

1.3.1 Modeling and identification

The first aspect to be considered in the model-based fuzzy control design is the modeling and identification of the process. This thesis considers analytical, linear or nonlinear models, and fuzzy models. Linear models are used in some examples throughout the thesis in order to emphasize the advantages of using fuzzy design specifications leading to nonlinear controllers even when the system is linear. Better performances can be obtained, because the specifications are closer to the desired criteria defined by the control designer, due to the added flexibility. For some nonlinear processes, the system’s behaviour can be described by suitable mathematical laws, leading to a model adequate for control, i.e., the model is not too complex and does not involve heavy computational effort. This modeling paradigm is usually known as white-box modeling. These models are used as simulation models in this thesis. In contrast to white-box modeling methods, a nonlinear model based on the so-called artificial intelligence (AI) techniques, e.g., fuzzy, neural, or neuro-fuzzy methods, can represent also (highly) nonlinear processes in an effective way due to their function approximation properties. Fuzzy models are widely used throughout the thesis. Modeling and identification, presented in Chapter 2, is used as a tool in this thesis, and no additional research concerning these subjects has been carried out.
1.3.2 Design specifications

Another important aspect in the design of a control system is the choice of the design specifications, which influences not only the parameters but also the type of controller to be used. These specifications can be made for different measures of the signals present in the system. The design goals given by the design specifications can be partly contradictory, and a trade-off between them must be made in order to choose the desired performance criteria. Usually, the control goals are expressed in terms of the size or limits of various signals. The tracking errors, for instance, given by the difference between the references \( r \) and the outputs \( y \) must be “small”, while the actuator signals \( u \) should, normally, be “not too large” and should preferably not exceed a certain limit. The design goals and objectives for controller design can be expressed in performance criteria using different approaches and different configurations of the control system. Fuzzy performance criteria are given by the combination of fuzzy goals and fuzzy constraints, and can be defined in fuzzy terms. Fuzzy multicriteria decision making is an approach that transparently translates the objectives and constraints to the control design goals of a given system. 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). In the control design, the decision goals and the constraints are defined on relevant system variables.

Beyond the use of fuzzy multicriteria decision making in direct control, this approach is also suitable at supervisory and planning levels, where a human operator or manager is involved in a decision making process. Therefore, by using the same mathematical framework one can control the system at different hierarchical levels. Although fuzzy logic controllers, as defined in Section 1.2, are used in industry, that design method is perhaps a less proper way of using the knowledge-based approach inherent to fuzzy control. The purpose of using fuzzy performance criteria is to allow the control of a given system at higher control levels, as supervisory or planning levels, combined to low level control (control of a system as a servo-mechanism or tracking problem in a signal-based approach). Design specifications are discussed in Chapter 3.

1.3.3 Types of controllers

As the model of the system under control is available, the evaluation of the process behaviour should not be made only in the closest neighborhood, i.e. for one-step-ahead prediction, but also for more steps ahead. When the models of process and disturbances are perfect, the horizon could be extended to infinity, and one could compute the optimal control actions by an optimal control approach. The process and its model are, however, always different, time varying, and subjected to changes in tracking and disturbances correlations. Thus, this model-plant mismatch hampers the application of optimal control due to the propagation of errors through the future time steps. However, a few steps ahead prediction has the potential to give better control performances than one-step-ahead, because the controller takes into account the future behaviour of the system. This approach is usually known as model-based
predictive control (MBPC) (Soeterboek, 1992; Camacho and Bordons, 1995). MBPC is a control methodology that has become an important research area in automatic control, which is characterized by the use of a process model to predict the process output at future discrete time instants over a specified prediction horizon, given the process inputs and the desired reference output. MBPC is presented in Section 4.1, and it is widely used in this thesis.

The model and the system under control are never exactly the same; i.e. there is always a model–plant mismatch. A control structure that is used to cope with this problem is the internal model control (IMC) scheme. In Section 4.2 of this thesis model–plant mismatches are alleviated by using the IMC scheme. A different approach to cope with this problem is introduced in this thesis, where a fuzzy compensator can compensate for steady-state errors based on the information contained on the considered model of the system under control. This approach is presented in Section 4.3.

1.3.4 Research topics developed in this thesis

Comparing fuzzy model-based control with conventional fuzzy control, it becomes clear that FMBC can combine the advantages of introducing fuzzy sets in control, into the classical design of control systems. In FMBC the controller uses a model of the process to compute the proper control actions at each time step. When fuzzy goals and fuzzy constraints are considered, human knowledge is used to specify the design specifications translated via fuzzy decision making in fuzzy performance criteria. This approach uses explicitly a model of the process to evaluate the degree of satisfaction of the control goals in the decision making algorithm.

As the research area of FMBC is rather broad, it would be impossible to cover it completely, and thus some choices were made a priori. First, due to their well-known properties, especially in the modeling and identification of nonlinear systems (Babuška, 1997), fuzzy models are considered. Some of these models have special properties, which make them analytically invertible, and theoretically, this inverted model can be directly used as a controller. This controller is ‘fast’, i.e. uses few computational effort. When the model is not invertible, or the system has nonminimum phase behaviour, fuzzy models can still be used in a model-based predictive control scheme. Chapter 5 presents controllers which are based on fuzzy models.

The use of fuzzy goals and fuzzy constraints in FMBC allows for a more flexible aggregation of the control objectives than other classical criteria as the weighted sum of squared errors. In this approach, a multistage decision-making algorithm is applied to compute the optimal control action. Compared to the standard quadratic objective function, the designer has more freedom in specifying the desired process behavior. Few research has been done in this area, and the applications presented in literature are usually at higher levels of control (Kacprzyk, 1997). This thesis gives some clues on how to use fuzzy design specifications in low and medium levels of control. A detailed description of the approach and the obtained
results are presented in Chapter 6. A discussion over some implementational problems is given in Chapter 7. Summarizing, the two main areas of FMBC presented in this thesis are:

- Fuzzy modeling in control. An invertible fuzzy model is used as a one-step-ahead controller allowing for time optimal control. When the fuzzy model is not invertible, the model is used in a MBPC scheme. Optimization problems resulting from the use of nonlinear models are alleviated by the use of a supervisory scheme combining the inversion of a fuzzy model with MBPC using discrete optimization techniques.

- Predictive control using fuzzy objective functions. The cost function usually applied in MBPC is substituted by a fuzzy decision function allowing for more flexibility in the definition of design specifications. This generalization leads often to optimization problems, hampering real-time applications with small sampling times. Special conditions where the optimization problem remains a convex problem are presented. Moreover, extensions of discrete optimization algorithms, e.g. branch-and-bound and genetic algorithms to reduce the computational time are also presented.

1.4 Outline of the thesis

This thesis presents some new developments in the field of fuzzy model-based control. Chapters 2 to 4 are intended to briefly describe the three different parts of FMBC: modeling and identification, design specifications and types of controllers, respectively. Chapters 5 to 7 present the new features in FMBC, followed by a real application of the developed techniques.

Chapter 2 presents the modeling and identification techniques used in this thesis. Models based on first-principles or mathematical descriptions of a system are used for control simulations or test case purposes. Fuzzy modeling is briefly described. The identification of fuzzy models from different types of knowledge is usually known as fuzzy identification. Identification of Takagi-Sugeno fuzzy models using product-space fuzzy clustering, which is the type of models most widely used in this thesis, is described.

Chapter 3 presents design specifications and their translation to design criteria. Design specifications for linear and nonlinear systems are discussed. Classical performance specifications, considering input-output specifications, regulation specifications and actuator effort are presented. Classical performance criteria are described by using norms and seminorms of signal and systems. A generalization to fuzzy performance criteria is made when the criteria are defined using fuzzy sets.

Chapter 4 presents the types of model-based controllers used in this thesis. Model-based predictive control is used as the general control methodology. Problems with model-plant mismatches and disturbances are alleviated using internal model control and fuzzy compensation. These techniques are both described in this chapter.
Chapter 5 presents controllers based on fuzzy models. First, the inversion of two different types of fuzzy models: singleton fuzzy model and TS fuzzy model affine with respect to the control action, is presented. On-line adaptation of singleton fuzzy models is discussed afterwards. When the inversion is not possible to derive due to nonminimum phase behaviour of the system, or in the presence of constraints, the inversion of the fuzzy model is combined with a predictive control scheme. After the presentation of this scheme, the chapter concludes with an example of pressure control covering all the presented control schemes.

Chapter 6 presents fuzzy model-based control using fuzzy objective functions. Fuzzy decision making theory is applied to model-based predictive control. First, a brief survey of fuzzy decision making is presented. The application of fuzzy decision making in the control environment, requiring multistage decision making, is discussed. The application of fuzzy decision making to control has two main design problems: the choice of the fuzzy criteria and the aggregation operators to combine them. Both problems are addressed, and illustrative examples are presented.

Chapter 7 discusses the optimization problems present in fuzzy model-based control. First, MBPC using fuzzy models and classical objective functions is addressed. For this situation, the optimization problem is non-convex. Two methods are presented to cope with this optimization problem: branch-and-bound and genetic algorithms. Secondly, MBPC using fuzzy objective functions is discussed. Special conditions under which this optimization problem remains convex are presented. For the general case, i.e., for non-convex optimization problems, a branch-and-bound algorithm is introduced.

Chapter 8 applies the control techniques developed in Chapter 5 and Chapter 6, using the optimization tools described in Chapter 7, to an air-conditioning system. Inversion control based on TS fuzzy models, PID control, MBPC with classical objective functions and MBPC using fuzzy objective functions are compared.

Chapter 9 present the main conclusions of this thesis. Suggestions for further research are briefly discussed.

Appendix A presents the basic concepts of fuzzy set theory utilized in this thesis. Appendix B gives the Gustafson–Kessel algorithm used in fuzzy clustering. Appendix C presents the fuzzy models of the air-conditioning system used in Chapter 8.

The main contributions of this thesis are the use of fuzzy objective functions in MBPC, the algorithms developed for the nonconvex optimization problems in MBPC, and the inversion methods for singleton fuzzy model and affine TS fuzzy models. The introductory chapters provide also some new insights in different areas, such as in Chapter 3, which introduces the use of fuzzy performance criteria, and Chapter 4, which introduces fuzzy compensation. Several examples are given throughout the thesis in order to clarify the different control methods developed.
Modeling and Identification

This chapter presents some of the modeling and identification techniques suitable for FMBC, and that are used in this thesis. The goal of the chapter is not to be a survey of the area and no exhaustive description of all the different methods is presented. A major focus is made on fuzzy modeling and identification, because fuzzy models are widely used in the thesis.

The development of a model of the process is essential for fuzzy model-based control. Traditional first-principle models are based on a deep knowledge of the nature of the system, and on a suitable mathematical treatment. These type of models are usually known as "white-box" models. Some systems are almost linear and can be approximated by a linear model. In general, that is not the case, and linear models are derived around a working point. This approach has the disadvantage of restricting the control to a certain 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 first-principles of the system are not available, linear identification techniques can be used. One of the most recent techniques in linear identification is subspace model identification (Verhaegen and Dewilde, 1992), where a linear state-space model of the form presented in Eq. (2.1) is obtained.

\[ x(k + 1) = Ax(k) + Bu(k), \]
\[ y(k) = Cx(k). \]  \hspace{1cm} (2.1)

The subspace identification technique determines the parameters of matrices A, B and C.

For some nonlinear processes, the system’s behaviour can be described by suitable mathematical laws, leading to a model usable in control, 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, a large number of processes are complex and only partly understood systems, where a good mathematical description of the underlying physics of the system is usually not possible with present-day tools.

As this white-box approach is laborious and inefficient for complex, and partially known systems, a nonlinear model based on artificial intelligence 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 function approximation properties. Fuzzy modeling is an attractive modeling technique because it combines first principles, knowledge obtained from linguistic rules describing the system and/or measurements.

This thesis uses linear models mainly as test cases. Nonlinear white-box models are used as the ‘real’ process in some simulation tests. Fuzzy models are widely used throughout the thesis, including real-time implementations. The general formulation of the modeling problem is presented in Section 2.1. The basic principles of fuzzy modeling are described in Section 2.2. When only data of the system under control is available, fuzzy identification, as presented in Section 2.3, can be utilized. The identification of fuzzy models using productspace fuzzy clustering, which is the type of fuzzy identification used in this thesis, is briefly described in Section 2.4.

2.1 Formulation of the modeling problem

Modeling is a technique that derives a model of the system under control. In this thesis, discrete nonlinear autonomous open loop models are considered. The general form of these multi-input multi-output (MIMO) models is given by

\[ x(k + 1) = f(x(k), u(k)), \]
\[ y(k) = g(x(k)), \]  

(2.2)

where \( u(k) \in \mathcal{U} \subset \mathbb{R}^m \) are the control actions, \( y(k) \in \mathcal{Y} \subset \mathbb{R}^p \) are the outputs of the system, \( x(k) \in \mathcal{X} \subset \mathbb{R}^n \) are the states, and \( k \) denotes discrete time samples. The domains of \( u, y \) and \( x \) are respectively given by \( \mathcal{U}, \mathcal{Y} \) and \( \mathcal{X} \). The orders of control actions, outputs and states are denoted by \( m, p, n \in \mathbb{N} \), respectively. The function \( f \) relates the states at time \( k + 1 \) with the states and the control actions at time \( k \), and the function \( g \) relates the outputs with the states at time \( k \). If the model is linear, Eqs. (2.2) are reduced to (2.1). Let \( f' \) and \( g' \) be the functions describing exactly the system. The objective of modeling is to approximate the functions \( f' \) and \( g' \), by the functions \( f \) and \( g \), 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. The state vector \( x \) can be obtained from the several inputs and outputs of the system, joining them in
a vector:

\[
x(k) = [y_1(k), \ldots, y_1(k-p_1+1), \ldots, y_p(k), \ldots, y_p(k-p_p+1), \]
\[
u_1(k), \ldots, u_1(k-m_1+1), \ldots, u_m(k), \ldots, u_m(k-m_m+1)]^T.
\] (2.3)

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. Further, 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, Eqs. (2.2) can be reduced to

\[
\hat{y}(k + 1) = f(x(k)).
\] (2.4)

The state variables \(x\) are called the regressor and the predicted outputs \(\hat{y}\) the regressand. The static nonlinear regression in (2.4) is widely used for the modeling of nonlinear dynamic systems, either in input-output or state-space form. The general MIMO system in (2.4) is depicted in Fig. 2.1a. This model can be decomposed in a collection of MISO systems, if

![Figure 2.1. Generic MIMO model and one its i-th MISO component.](image)

each MISO system is represented by a MISO Nonlinear Auto Regressive with eXogenous input (NARX) model. Denote \(y_i, i = 1, \ldots, p\), an output considered for a particular \(i\) MISO NARX model. This MISO system, shown in Fig. 2.1b, is given by:

\[
\hat{y}_i(k + 1) = f_i(x(k)).
\] (2.5)

The total MIMO system is given by the collection of all the \(p\) MISO systems. Note that the definition of the state vector as in (2.3) allows that any previous input or output of the system can be a state for each particular MISO model \(i, i = 1, \ldots, p\). The collection of the MISO systems, as in (2.5), defines thus a general MIMO system as in (2.4).

### 2.2 Fuzzy modeling

From the several AI modeling techniques, 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 linguistic rules describing the system and/or measurements, obtaining an usually called gray-box modeling approach (Babuška, 1997). 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 (Kaymak and Babuška, 1995; Yoshinari, et al., 1993; Zhao, et al., 1994), neural learning methods or orthogonal least squares (Wang, 1994). As shown in (Babuška and Verbruggen, 1994a), for low-order nonlinear systems, fuzzy models provide a transparent, gray-box description of the process dynamics that reflects the nature of the process nonlinearities. The processes used in this thesis, and, in general, a large number of processes, are partly known systems, where first principles and measurements can be synergistically combined. Therefore, fuzzy modeling is the modeling technique most used in this thesis.

The theory of fuzzy sets can be employed in the modeling of systems in different ways. Normally, 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, have been recognized as universal function approximators (Wang, 1992; Kosko, 1994; Zeng and Singh, 1995). Thus, in fuzzy modeling the fuzzy If–Then rules take the 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 thesis:

1. Linguistic or Mamdani fuzzy model (Zadeh, 1973; Mamdani, 1977), where both the antecedent and consequent are fuzzy propositions (see Section 2.2.1).

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 Section 2.2.3.

The singleton fuzzy model, can be seen as a particular case of both linguistic and TS fuzzy models, and is presented in Section 2.2.2. Some basic definitions of fuzzy logic used in this chapter and throughout the thesis are given in Appendix A.
2.2.1 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:

\[ R_i: \text{If } x \text{ is } A_i \text{ then } y \text{ is } B_i, \quad i = 1, 2, \ldots, K, \]

where \( R_i \) denotes the \( i \)th rule and \( K \) is the number of rules. The antecedent variable is given by \( x \in \mathcal{X} \subseteq \mathbb{R}^n \) and represents the input of the fuzzy system. Similarly, \( y \in \mathcal{Y} \subseteq \mathbb{R}^p \) 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 Section 2.1. \( A_i \) and \( B_i \) are fuzzy sets described by the membership functions \( \mu_{A_i}(x): \mathcal{X} \rightarrow [0, 1] \) and \( \mu_{B_i}(y): \mathcal{Y} \rightarrow [0, 1] \), respectively. Fuzzy sets \( A_i \) define regions in the antecedent space \( \mathcal{X} \), and fuzzy sets \( B_i \) define regions in the consequent space \( \mathcal{Y} \). The antecedents are usually defined as a combination of simple fuzzy propositions for each \( x_j, j = 1, \ldots, n \) of the vector \( x \), instead of using multidimensional fuzzy sets. In a similar way, the consequents can also be divided in simple fuzzy propositions \( y_l, l = 1, \ldots, p \). For these fuzzy propositions it is usual to attribute linguistic meanings such as ‘high temperature’, ‘small velocity’, etc. 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”, etc. Different fuzzy logic operators, such as conjunction, disjunction and complement can be used to combine the antecedent propositions. The most common is however the conjunctive form given by:

\[ R_i: \text{If } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } \ldots \text{ and } x_n \text{ is } A_{in} \]
\[ \text{ then } y_1 \text{ is } B_{i1} \text{ and } y_2 \text{ is } B_{i2} \text{ and } \ldots \text{ and } y_p \text{ is } B_{ip}, \]

where one-dimensional fuzzy sets are defined for each component of the antecedent and consequent vectors. Note that the conjunctive form divide 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 (2.6) is a fuzzy relation \( R_i: (\mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]) \), which is computed by

\[ \mu_{R_i}(x, y) = I(\mu_{A_i}(x), \mu_{B_i}(y)), \]

where the operator I can be a fuzzy implication (Klir and Yuan, 1995) or a conjunction operator (\( t \)-norm, see Appendix A). The entire rule base is represented by combining the \( K \) relations \( R_i \) of the individual rules into a global relation \( R \). If \( I \) is an implication, \( R \) is obtained by making a conjunction of all the \( R_i \), and if \( I \) is a conjunction operator, \( R \) is computed as a disjunction of the individual relations \( R_i \) (Jager, 1995). For a given input “\( x \) is \( A' \)”, and the relation \( R \), the corresponding output fuzzy set \( B' \) is derived by:

\[ B' = A' \circ R, \]
where $\odot$ denotes the sup-$t$ composition (Klir and Yuan, 1995). The most used norms in this composition are the minimum for the $t$-norm and the maximum for the $s$-norm ($t$-conorm), leading to the following composition:

$$
\mu_{B'}(y) = \max_x \min_{x,y} (\mu_{A'}(x), \mu_R(x,y)).
$$

(2.10)

When the implication $I$ in (2.8) is chosen to be the minimum conjunction operator, $\mu_R$ becomes the minimum of $\mu_{A_i}$ and $\mu_{B_i}$, and the compositional 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 $\beta_i$ of the antecedent is computed for each rule $i$:

$$
\beta_i = \mu_{A_{1i}}(x_1) \land \mu_{A_{2i}}(x_2) \land \ldots \land \mu_{A_{ni}}(x_n), \quad 1 \leq i \leq K.
$$

(2.11)

2. For each rule derive the output fuzzy set $B'_i$ using the minimum $t$-norm:

$$
\mu_{B'_i}(y) = \beta_i \land \mu_{B_i}(y),
$$

(2.12)

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

$$
\mu_{B'}(y) = \max_{i=1,2,\ldots,K}(\mu_{B'_i}(y)).
$$

(2.13)

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 in a single numerical value. The two most common defuzzification methods are the center-of-gravity (COG) and the mean-of-maxima (MOM) (Jager, 1995). For discrete domains $\mathcal{Y}$, the COG method computes each coordinate $y_j, j = 1, \ldots, p$ of the center of gravity for the fuzzy set $B'$ as a weighted sum:

$$
cog_{y_j}(B') = \frac{\sum_{\ell=1}^{N_t} \mu_{B'}(y_{\ell}) y_{j,\ell}}{\sum_{\ell=1}^{N_t} \mu_{B'}(y_{\ell})},
$$

(2.14)

where $N_t$ is the number of linguistic labels used for the discretization of the $\mu_{B'}(y)$. The point $y_{\ell}$ is the $\ell$th linguistic label of the space $\mathcal{Y}$. The MOM method computes the mean value of the interval with the largest membership degree. The Mamdani (max-min) inference utilizes the COG method, because it provides interpolation between the consequents proportional to the height of the individual consequent sets. As the Mamdani inference method itself does not provide any interpolation, this defuzzification method is necessary to avoid the stepwise output resultant from the application of the MOM method (Jager, 1995). The MOM method is normally used with the inference based on fuzzy implications, to select the "most possible" output.
2.2.2 Singleton fuzzy model

A special case of the linguistic fuzzy model is obtained when the consequent sets $B_i$ 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 $\mathcal{Y} \subseteq \mathbb{R}$. Singleton sets can be represented as real numbers $c_i$, yielding the following rules:

$$R_i: \text{If } x \text{ is } A_i \text{ then } y = c_i, \quad i = 1, 2, \ldots, K.$$  \hspace{1cm} (2.15)

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

$$y = \frac{\sum_{i=1}^{K} \beta_i}{\sum_{j=1}^{K} \beta_j} c_i. \hspace{1cm} (2.16)$$

This defuzzification depends on the number of rules $K$, and not on the number of fuzzy sets for a certain output $y_j$, $j = 1, \ldots, p$, like in (2.14). The singleton model can also be seen as a special case of the Takagi–Sugeno fuzzy model, presented in Section 2.2.3.

Contrary to the linguistic fuzzy model, the consequent parameters $c_i$ of the singleton model can be easily estimated from data 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

$$y = \sum_{i=1}^{K} \Phi_i(x) c_i. \hspace{1cm} (2.17)$$

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 $\Phi_i(x)$ are given by the normalized degrees of fulfillment of the rule antecedents, and the constants $c_i$ in (2.17) are the consequents in (2.15).

Beyond the fact that singleton fuzzy models are relatively easy to identify, they have also 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 given in Section 5.1.3.

2.2.3 Takagi–Sugeno fuzzy models

Takagi and Sugeno (1985) introduced a fuzzy rule-based model that can approximate a large number of nonlinear systems. The Takagi–Sugeno (TS) fuzzy model consists of the
generalization of the singleton model, where the rule consequents are not constants, but crisp functions of the model input:

$$R_i: \text{If } x \text{ is } A_i \text{ then } y_i = f_i(x), \quad i = 1, 2, \ldots, K,$$  \hfill (2.18)

where $R_i$ denotes the $i$th rule, $K$ is the number of rules, $x$ is the antecedent variable, $y$ is the unidimensional consequent variable and $A_i$ is the antecedent fuzzy set of the $i$th rule, as for the linguistic model in Section 2.2.1. Each rule $i$ has a different function $f_i$ yielding a different value for the output $y_i$. This fuzzy model can be generalized for $p$ outputs, and it is denoted by

$$R_i: \text{If } x \text{ is } A_i \text{ then } y_i = f_i(x), \quad i = 1, 2, \ldots, K.$$  \hfill (2.19)

Note that the index in the outputs $y_i$ and the functions $f_i$ corresponds to the $i$th rule. For the sake of simplicity, the form in (2.18) is used in the sequel. In fact, (2.19) is the representation of a MIMO fuzzy model, which is decomposed in several MISO systems as in (2.18). When the states $x$ are defined as in (2.3), the MIMO fuzzy model in (2.19) can be decomposed into a collection of MISO fuzzy models as in (2.18), without loss of generality. The antecedent proposition $A_i$ can again be a combination, usually in a conjunctive form, of simple propositions for each $x_j$, $j = 1, \ldots, n$, as in (2.7). The consequent functions $f_i$ in (2.18) 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, yielding the rules:

$$R_i: \text{If } x \text{ is } A_i \text{ then } y_i = a_i^T x + b_i,$$  \hfill (2.20)

where $a_i$ is a parameter vector and $b_i$ 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 $\mathbb{R}^n \times \mathbb{R}$. Note that when $a_i = 0$, $i = 1, \ldots, K$, the consequents in model (2.20) are constant functions, and the model is a singleton model as presented in Section 2.2.2.

For the sake of simplicity let $\mu_i(x) \triangleq \mu_{A_i}(x)$ denote the membership function of the antecedent $A_i$. The inference mechanism proposed by Takagi and Sugen (1985) is reduced to a straightforward extension of the fuzzy-mean defuzzification presented in (2.16):

$$y = \sum_{i=1}^{K} \beta_i(x) y_i = \sum_{i=1}^{K} \beta_i(x)(a_i^T x + b_i)$$  \hfill (2.21)

where $\beta_i(x)$ is the normalized degree of fulfillment of the $i$th rule’s antecedent given by:

$$\beta_i = \frac{\mu_i(x)}{\sum_{j=1}^{K} \mu_j(x)}.$$  \hfill (2.22)

When the antecedent fuzzy sets define distinct, but overlapping regions in the antecedent space, the TS model can be regarded as a smoothed piece-wise linear approximation of a nonlinear function.
2.3 Fuzzy identification

The previous sections reviewed the structures and inference mechanisms of different rule-based fuzzy models. The construction of the fuzzy models is now discussed, and it is usually known as fuzzy identification. It is assumed that the structure of the system, i.e. the input and outputs variables, are determined beforehand. For dynamic systems as in (2.4), 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, user interfaces, etc. Note that this section do not pretend to do a complete description of fuzzy identification techniques. A survey of these identification techniques can be found in Babuška (1997).

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 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, using fuzzy clustering methods based on adaptive distance measures (Gustafson and Kessel, 1979). 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 (Babuška, 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 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 thesis for the identification of fuzzy models, and it is briefly presented in the next section.

2.4 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, and 2) parameter estimation. These two steps are briefly reviewed below, with attention focused on the parameter estimation problem. The identification procedure presented below is for affine TS models as in (2.20). The identification of singleton models, also used in this thesis and presented in Section 2.2.2,
is just a particular case of an affine TS model. Product-space clustering can also be advantageously used in the identification of linguistic and fuzzy relational models (Babuška, 1997).

### 2.4.1 Structure identification

Structure identification allows to transform the dynamic identification problem into a static nonlinear regression. Suppose that the structure of the model is given by (2.4). For the sake of simplicity, let each MISO system be identified separately. As described in Section 2.1, the total MIMO system can be derived as a collection of MISO systems. A MISO system can thus be described by

\[ \hat{y}(k + 1) = f(x(k)). \]  

(2.23)

Product-space fuzzy clustering is based on the data in the product space of the regressor and the regressand \( X \times Y \). Let \( N \) denote the number of data samples, selected from the input and output data sequences. This number must be much greater than the number of states in the system, \( N \gg n \). Let \( N_d \) be the number of points actually used in the identification. Let also \( h \) denote the highest order of the several inputs and outputs in (2.3). Then, \( N_d = N - h \). Let \( \Phi \) denote the regressand matrix in \( \mathbb{R}^{N_d \times n} \) having the state vectors \( x(i)^T \) in its rows, and \( Y \) denote the vector in \( \mathbb{R}^{N_d} \) containing the regressands \( y(i + 1) \), with \( i = h, \ldots, N - 1 \):

\[
\Phi = \begin{bmatrix}
    x(h)^T \\
    \vdots \\
    x(N - 1)^T
\end{bmatrix}, \quad
Y = \begin{bmatrix}
    y(h + 1) \\
    \vdots \\
    y(N)
\end{bmatrix}.
\]  

(2.24)

The matrix \( \Phi \) contains shifted versions of the input and output data, as in (2.3). An example is presented in Example 2.4.1. In this example \( h = 2 \), and \( N_d = N - 2 \).

**Example 2.4.1** Let an NARX model be given by \( \hat{y}(k + 1) = f(y_1(k), y_2(k), y_2(k - 1), u_1(k), u_2(k)) \), and let the considered output of the MISO model be \( y_1 \). Having \( N \) data samples for \( y_1, y_2, u_1 \) and \( u_2 \), the regressor matrix and the regressand vector are given by:

\[
\Phi = \begin{bmatrix}
    y_1(2) & y_2(2) & y_1(1) & u_1(2) & u_2(2) \\
    y_1(3) & y_2(3) & y_2(2) & u_1(3) & u_2(3) \\
    \vdots & \vdots & \vdots & \vdots & \vdots \\
    y_1(N - 1) & y(N - 1) & y_2(N - 2) & u_1(N - 1) & u_2(N - 1)
\end{bmatrix}, \quad
Y = \begin{bmatrix}
    y_1(3) \\
    y_1(4) \\
    \vdots \\
    y_1(N)
\end{bmatrix}.
\]

Assuming that the structure is correctly chosen, the unknown nonlinear mapping between \( Y \) and \( \Phi \) can be estimated from the data set. The structural parameters \( m_1, \ldots, m_m \) and \( p_1, \ldots, p_p \), as in (2.3), are chosen on the basis of prior knowledge or automatically by comparing different structures in terms of some suitable criteria (Sugeno and Kang, 1988).
2.4.2 Parameter estimation

At this step, the number of rules $K$, the antecedent fuzzy sets $A_i$, and the consequent parameters $a_i, b_i$ for $i = 1, \ldots, K$, as in (2.20), are determined. Fuzzy clustering in the Cartesian product space $\mathcal{X} \times \mathcal{Y}$ is applied to partition the data into subsets, which can be approximated by local linear models (Babuška and Verbruggen, 1995c). The data set $Z$ to be clustered is formed by appending $\mathcal{Y}$ to $\Phi$:

$$Z = [\Phi, \mathcal{Y}]^T. \quad (2.25)$$

The columns of $Z$ are denoted by $\{z_{\ell}, \ell = 1, \ldots, N_d\}$. Let $U = [\mu_{i\ell}] \in [0, 1]^{(K \times N_d)}$ denote a fuzzy partition matrix of $Z$. Let $V$ be a vector of cluster prototypes (centers) to be determined, defined by $V = [v_1, v_2, \ldots, v_K]$, and let $F$ be a set of cluster covariance matrices $F = [F_1, \ldots, F_K]$, where $F_i$ are positive definite matrices in $\mathbb{R}^{(n+1) \times (n+1)}$. Given $Z$ and the desired number of clusters $K$, the Gustafson and Kessel (1979) (GK) algorithm computes the fuzzy partition matrix, $U$, the prototype matrix of cluster means, $V$, and a set of cluster covariance matrices $F$:

$$\begin{array}{c}
\text{clustering} \\
(Z, K) \longrightarrow (U, V, F).
\end{array} \quad (2.26)$$

The GK algorithm is briefly presented in Appendix B. Given the triplet, $(U, V, F)$ and the antecedent membership functions $A_i$, the consequent parameters $a_i$ and $b_i$ can be computed, as explained in the following.

**Number of clusters.** The number of clusters determines the number of rules in the obtained fuzzy model. Two main approaches to determine the appropriate number of clusters in data can be distinguished:

- Cluster the data for different values of $K$ and then use validity measures to assess the goodness of the obtained partitions. 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 successively this number by merging clusters that are compatible with respect to some predefined criteria. This approach is called compatible cluster merging (Krishnapuram and Freg, 1992; Kaymak and Babuška, 1995).

**Antecedent membership functions.** Each cluster represents one TS fuzzy rule, as in (2.20). The multidimensional membership functions $A_i$ are given analytically by computing the distance of $x(k)$ from the projection of the cluster center $v_i$ onto $\mathcal{X}$, and then computing the membership degree in an inverse proportion to the distance. Denote $F_i^x = [f_{ij}], 1 \leq
$j, l \leq n$ the submatrix of $F_i$. This matrix describes the form of the cluster in the antecedent space $X$. From the GK algorithm, the corresponding norm-inducing matrix is given by:

$$M_i^* = [\det(P_i^*)]^{1/n}(P_i^*)^{-1}. \tag{2.27}$$

Let $v_i^* = [v_{i1}, \ldots, v_{im}]^T$ denote the projection of the cluster center onto the antecedent space $X$. The inner-product distance norm,

$$d_{il} = (x(k) - v_i^*)^T M_i^*(x(k) - v_i^*). \tag{2.28}$$

is converted into the membership degree by:

$$\mu_{A_i}(x(k)) = \frac{1}{\sum_{j=1}^{K} (d_{il}/d_{ij})^{2/(m-1)}}, \tag{2.29}$$

where $m$ is the fuzziness parameter of the GK algorithm given in Appendix B.

**Consequent parameters.** Optimal consequent parameters are estimated by the least-squares method. Let $\theta_i^T = [a_i^T, b_i]$, let $\Phi_e$ denote the matrix $[\Phi, 1]$, and let $\Gamma_i$ denote a diagonal matrix in $\mathbb{R}^{N_x \times N_x}$ having the membership degree $\mu_{A_i}(x(k))$ as its $i$th diagonal element. Denote $\Phi'$ the matrix in $\mathbb{R}^{N_x \times K(n+1)}$ composed from matrices $\Gamma_i$ and $\Phi_e$ as follows:

$$\Phi' = [(\Gamma_1 \Phi_e), (\Gamma_2 \Phi_e), \ldots, (\Gamma_K \Phi_e)]. \tag{2.30}$$

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

$$\theta' = [\theta_1^T, \theta_2^T, \ldots, \theta_K^T]^T. \tag{2.31}$$

The resulting least squares problem, $Y = \Phi' \theta' + \epsilon$, has the solution

$$\theta' = [(\Phi')^T \Phi']^{-1} \Phi' Y. \tag{2.32}$$

The optimal parameters $a_i$ and $b_i$ are given by:

$$a_i = [\theta_{s+1}^T, \theta_{s+2}^T, \ldots, \theta_{s+n}^T]^T,$$

$$b_i = [\theta_{s+n+1}^T], \text{ where } s = (i - 1)(n + 1). \tag{2.33}$$

With the determination of the parameters $a_i$ and $b_i$, the fuzzy model identification procedure is completed. If several outputs are considered, the procedure must be repeated for each output.
2.5 Summary and concluding remarks

This chapter presented the modeling and identification techniques used in this thesis. These techniques are used as tools, and no research was performed in these fields. This thesis uses models based on first-principles or mathematical descriptions of a system, when this is possible. This type of models, also known as white-box models, are used for control simulations or test case purposes. Unfortunately, this type of models are usually not possible to obtain, because of the poor knowledge of the system, or the known models are too complex to be used in control applications. For this type of problems, which is the mainly considered in this thesis, models extracted from data of several inputs and outputs of the system, possibly combined to 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 the several AI modeling techniques due to its interesting properties. In fact, fuzzy models can incorporate different types of knowledge, and derives a gray-box model, i.e., a model that is not described by mathematical principles, but that can be easily interpretable, because it is usually a rule-based linguistic model, at least in some of its parts. 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 an TS fuzzy model was described. It starts by defining the structure of the system. The parameters must be estimated 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 rest of the thesis.
Performance Criteria

In the design of control systems, design specifications are usually the translation of the main goal in control design, i.e., the outputs $y$ should be as close as possible to respective pre-specified references $r$, suppressing the influence of the disturbances. This goal must be accomplished despite the fact that $u$ 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: stability for the overall system, performance regarding the accuracy and the speed of the system’s response and robustness to disturbances and dynamics which 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 Section 3.1, 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 Section 3.2. 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 Section 3.3.

Classical design specifications are specified by using performance criteria, which are based on norms or semi-norms. Sometimes, it is however 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 type of control goals can be formally translated to performance criteria using fuzzy logic theory. Fuzzy performance criteria are presented in Section 3.4.

### 3.1 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 mutually well on all these goals. Several criteria can however 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 $\phi_{os}$ must be smaller than 5%” (see the definition of overshoot in Eq. (3.7), Section 3.2). This section presents design specifications and criteria. The possible combination of design criteria for the design of controllers is discussed. First, generally used approaches of control design specifications for linear systems are presented. A generalization for nonlinear systems is discussed afterwards.

#### 3.1.1 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 $J_i$, which is dependent on different variables of the system, such as the control actions $u$, the outputs of the system $y$, the states
x, the disturbances d, etc. 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 multicriterion 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 $J_i$ must be translated to functions of only one variable, usually the control actions $u$. Moreover, an ordering of the several criteria must be given, and a unique solution of the optimal problem is obtained. Let $v$ be a general variable under optimization. Each design criterion is thus translated to a function of this variable represented by $J_i(v)$. The combination of all the design criteria is given by the cost (objective) function $J(v)$:

$$J(v) = f(J_1(v), \ldots, J_L(v)),$$  \hspace{1cm} (3.1)

where $L$ is the number of criteria defined, and the function $f$ combining the several criteria defines an ordering between them. The two most used methods to combine criteria; weighted-sum and weighted-max, are presented after the formal definition of optimal control, briefly discussed in the following paragraph.

**Optimal control problem.** The general form of an optimization problem, usually known as nonlinear constrained optimization problem is defined as:

$$\min_{v \in V} J(v)$$ subject to $C_i(v) \leq 0$, \hspace{1cm} (3.2)

where the objective function $J(v)$ is defined as before, the constraint functions $C_i(v)$ are real-valued scalar functions, $v \in V$, and $N_c$ is the number of constraints. The constraints in a system can be present for the control actions $u$, state variables $x$, outputs of the system $y$, or changes in these variables. Note that in the optimal control formulation all the constraints must be expressed in the constraint functions $C_i$ depending on the chosen variable under optimization $v$.

As an example, let the variable under optimization be the control actions, i.e., $v \triangleq u$. Let the design criterion be given simply by the error between the desired reference and the predicted outputs using the model of the system: $J(u) = r - y$. Considering a regulation problem, the references are constant. Thus, in this case it is sufficient to have a function relating $y$ to $u$:

$$y = f(u),$$ \hspace{1cm} (3.3)

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 $C_i(v)$ are usually known as 'hard' constraints, contrary to the 'soft' constraints $J_i(v)$. This terminology of 'hard' and 'soft' constraints is generalized for fuzzy constraints in Section 3.4. Each criterion $J_i$ 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 $J(v)$ 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.

**Weighted-Sum Objective.** A common method to combine the individual goals translated in a design criterion $J_i$ is to add all of them, after they have been multiplied by non-negative weights $\lambda_i$:

$$J(v) = \lambda_1 J_1(v) + \ldots + \lambda_L J_L(v).$$  \hspace{1cm} (3.4)

The weights assign relative values among the functionals $J_i$. The objective function of the form (3.4) 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 Section 4.1).

**Weighted-Max Objective.** Another approach called minimax design, is to form the objective function as the maximum of the weighted functions:

$$J(v) = \max\{\lambda_1 J_1(v), \ldots, \lambda_L J_L(v)\},$$  \hspace{1cm} (3.5)

where $\lambda_i$ are non-negative weights as before. 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 $J(v)$ 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 choose the respective weights for the different criteria.
3.1.2 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, a 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 in 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, model-plant mismatches, etc., 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 thesis does not address explicitly 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 Section 4.2 or fuzzy compensation described in Section 4.3. Hence, only performance specifications are explicitly treated in this thesis. 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 it is usually done for linear systems. The next section presents classical, i.e., non-fuzzy, performance specifications usually defined for linear and nonlinear systems.

3.2 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 by themselves can also be contradictory as design specifications. 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 in the following groups:

1. input/output (I/O) specifications, related to the effect of the control actions $u$ into the system’s outputs $y$,

2. regulation specifications, measuring the effect of the disturbances $d$ and $d_m$ into $y$, and

3. actuator effort of the control actions $u$.

Sometimes the combined effect of disturbances and control actions on the output is also considered. The following sections describe each of these three most utilized groups of performance specifications in more detail.

3.2.1 I/O specifications

It is usual to express specifications on the system outputs $y$ 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 $P$. 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 setpoint). Let $h(k)$ denote the unit step response of the SISO mapping describing a system $P$. 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 must 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, that can be translated as

$$\phi_{at}(P, A) \overset{\Delta}{=} \lim_{k \to \infty} A h(k) = A,$$

(3.6)

where $A \in \mathbb{R}$ is the amplitude of the step.

Both the overshoot and the undershoot are defined as functions of $P$. The overshoot is defined as

$$\phi_{os}(P, A) \overset{\Delta}{=} \sup_{k \geq 0} A(h(k) - 1),$$

(3.7)

and the undershoot as,

$$\phi_{us}(P, A) \overset{\Delta}{=} \sup_{k \geq 0} -A h(k).$$

(3.8)

The rise time and settling time can be defined in different ways. In general, the rise time is defined as

$$\phi_{rise}(P) \overset{\Delta}{=} \inf\{K \mid Ah(k) > \lambda A, \quad k \geq K\},$$

(3.9)

where a common value for the parameter is $\lambda = 0.8$. The settling time is given by

$$\phi_{set}(P) \overset{\Delta}{=} \inf\{K \mid |Ah(k) - A| < \epsilon, \quad k \geq K\},$$

(3.10)

where the parameter $\epsilon$ is usually set to 0.05 or 0.02. Fig. 3.1 presents an example of a step with amplitude $A = 1$, where the overshoot, the undershoot, the rise time with $\lambda = 0.8$, and the settling time with $\epsilon = 0.05$ are illustrated. Other specifications normally used for the step response of linear systems are the general step response envelope specification, the general response-time functional or the step response interaction. The readers interested in these specifications are referred to, e.g., (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 $y$ should track the respective references $r$ 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:

$$e(k) = r(k) - y(k).$$

(3.11)

Some norms of these error signals, as their root-mean square values, the average-absolute norm or the $\infty$-norm (peak), are commonly used as performance criteria for control purposes. The definitions of these performance criteria are given in Section 3.3.
3.2.2 Regulation specifications

This type of specifications considers the effect of the disturbances \( d \) and \( d_m \) in the outputs of the system, assuming that the control signals \( u \) 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 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 requiring 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 Section 3.3.1) of the obtained outputs must be 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 can not 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, which falls out of the scope of this thesis.
3.2.3 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:

- *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 $\infty$-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 Section 3.4.

3.3 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).

3.3.1 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 $\mathbf{v}(t)$ denote a time signal in a vector space $\mathcal{V}$. A norm of $\mathbf{v}$, represented by $\|\mathbf{v}\|$ maps the space $\mathcal{V}$ to $\mathbb{R}$ and has the following four properties:
1. \[ \|v\| \geq 0 \text{ (Nonnegativity)} \]

2. \[ \|v\| = 0 \iff v = 0, \text{ (Positive definiteness)} \]

3. \[ \|\alpha v\| = |\alpha|\|v\|, \quad \forall \alpha \in \mathbb{R} \text{ (Homogeneity)} \]

4. \[ \|v + w\| \leq \|v\| + \|w\| \text{ (Triangle inequality)} \]

for any \( v, w \in V \). 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 time and frequency domain, where 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, 2-norm and \( \infty \)-norm. These norms can be derived as special cases of a \( p \)-norm defined as follows:

\[
\|v\|_p \triangleq \left( \int_0^\infty |v(t)|^p dt \right)^{1/p}.
\]

(3.12)

**1-norm.** This norm is the integral of the absolute value of a signal \( v(t) \):

\[
\|v\|_1 \triangleq \int_0^\infty |v(t)| dt,
\]

(3.13)

and can be seen as a measure of the total fuel or resource consumption.

**2-norm.** The 2-norm of a signal gives the square root of the total energy, and is given by:

\[
\|v\|_2 \triangleq \left( \int_0^\infty |v(t)|^2 dt \right)^{1/2}.
\]

(3.14)

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.
\( \infty \)-norm (Peak). 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 \( \infty \)-norm of \( v \) is thus the least upper bound (supreme) of the absolute value of a signal, given by

\[
\|v\|_{\infty} \triangleq \sup_{t \geq 0} |v(t)|.
\]  

The \( \infty \)-norm of a signal depends entirely on the extreme or large values the signal takes on. As the \( \infty \)-norm depends on occasionally large values of the signal, it is a worst case norm.

**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

\[
\|v\|_{\text{ITAE}} \triangleq \int_{0}^{\infty} t|v(t)|dt.
\]  

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-emphasize the signal at the beginning of the response. Thus, for this norm the steady-state behaviour of the signal is more important than the transient behaviour.

**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

\[
\|v\|_{\text{rms}} \triangleq \left( \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} v(t)^2 dt \right)^{1/2},
\]  

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.

**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

\[
\|v\|_{\text{aa}} \triangleq \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} |v(t)|dt ,
\]  

supposing that the limit in (3.18) exists. The \( \|v\|_{\text{aa}} \) semi-norm is useful in measuring the average resource used (like fuel), when the resource consumption is proportional to \( |v(t)| \).
The comparison of the three (semi-)norms: $\infty$-norm, $\|v\|_{\text{rms}}$ and $\|v\|_{\text{aa}}$, shows that they simply put different emphasis on large and small signal values. The $\infty$-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 be extended to the norm of a system.

### 3.3.2 Norms of systems

Let $H$ be a mapping from a given input $w$ to an output $z$ as in Fig. 3.2. The input can be, e.g., a control action $u$, a disturbance $d$, etc. The output can be a system's output $y$, for instance. Note that $H$ can be a subsystem of the total considered system $P$.

![Figure 3.2. Input-output mapping of a subsystem.](image)

The notion of norm can be used for the mapping $H$ 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 $H$ is defined as

$$
\|H\|_{ip} \triangleq \max_{\|w\|_p < \infty} \frac{\|z\|_p}{\|w\|_p},
$$

(3.19)

using the definition of $p$-norm as in (3.12). A norm for a particular mapping is used when an input signal, e.g., a step, sinusoidal, impulse, etc. is applied to the system, and the output $z$ 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 behaviour is readily available.

A completely different situation occurs when the system under control is linear. The norms defined in Section 3.3.1 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 $H$ to the given input signals.

Classical performance criteria pretend to translate the performance specifications in formal terms such that the behaviour of the system is as close as possible to the desired behaviour. Performance specifications are sometimes contradictory, and a compromise between them is then necessary. This is also the case in decision making problems, in which also a compromise between competing criteria and requirements should be obtained. For this type of problems it is useful to fuzzify the criteria and requirements, leading usually 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 then classical performance criteria due to the inherent fuzziness present in the performance specifications and the flexibility introduced in the definition of the performance criteria. Moreover, the combination of different criteria can be made using different and more general operators than the ones presented in Section 3.1.1, by using decision functions to combine the several criteria.

### 3.4 Fuzzy performance criteria

Classical performance criteria can only be defined using norms or semi-norms. However, sometimes informal design goals translate better 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, generalizing 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, classically known as performance specifications, 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 6. Each performance criterion is described by a membership function. As an example, let each error in (3.11) be defined by a triangular membership function, as depicted in Fig. 3.3. Note that this definition is for a certain time step \( k \), and a confluence of the different errors for the all response should be made to obtain a measure that is a norm or semi-norm.

\[ \mu \]

\[ e_{\min} \quad 0 \quad e_{\max} \quad e \]

**Figure 3.3. Example of a performance criteria defined for an error, as defined in (3.11).**

This procedure is however time consuming, and normally not necessary, because 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 (3.10). In fact, the I/O specifications defined in Section 3.2.1 can be almost straightforward used 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 can not 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 Section 3.3 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 \( \infty \)-norm as in (3.15). 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 (3.13) and (3.14).

Fuzzy decision making applied to control is presented in Section 6.2, and no further discussion over this subject is made at this point. 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 Section 6.2.3, showing the advantages of generalizing the objective function, usually at the cost of increasing computational time, to derive the optimal control actions.
3.5 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. However, this formulation is not possible when the system is nonlinear and/or fuzzy performance criteria are considered. 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 thesis do 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 the controlled system.

Classical performance specifications are usually divided in I/O specifications, regulation specifications and actuator effort. The I/O specifications are defined by some measures on a transient response. For the step, for instance, overshoot, rise time or settling time can be considered. Regulation specifications are defined for the disturbances $d$ and $d_m$, 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, and, in the classical approach, are usually built by using different norms or semi–norms of the signals of interest. Several norms, such as 1-norm, 2-norm, or $\infty$-norm are used to measure the size of different signals in the transient regime. The norms more 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, because it is an approach that 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.
Types of controllers

This chapter presents the different types of controllers used in this thesis. As expected, all control schemes are based on a model of the system under control. Predictive control is generally considered a good control strategy, because it can incorporate the predicted future behaviour of the system in the determination of the control actions, leading in general to improved performance.

Model-based predictive control is a general control methodology that has become an important research area in automatic control theory. It has been widely applied in industry (Richalet, 1993) and the theory of model-based predictive control has been studied extensively in the literature (Camacho and Bordons, 1995; Soeterbock, 1992). Reasons for this success are the ability of MBPC to control multivariable (nonlinear) systems under various constraints in an optimal way (with respect to the specified objective function). This control technique is described in Section 4.1. MBPC has revealed some problems in coping with model-plant mismatches and disturbances. However, using an additional control loop, the control performance can be improved.

A possible solution for the problems stated above is to incorporate MBPC in an internal model control scheme, as presented in Section 4.2. In this control scheme, the errors between the outputs of the model and the outputs of the process are used to shift the amplitude of the reference values. A linear feedback filter is usually included in the feedback loop before the signal is added to the reference. This filter aims to filter the process disturbances, reducing the gain and improving the robustness characteristics (Economou, et al., 1986). However, this linear filter deteriorates the closed loop dynamics.

These problems can be overcome by using a technique to compensate steady-state errors called fuzzy compensation, introduced by Sousa, et al. (1996a), and presented in Section 4.3. This control scheme uses information contained in the model and a fuzzy set defining the steady-state of the system to add compensation control actions.
4.1 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, et al. (1987) introducing Generalized Predictive Control and Soeterboek (1992) defining 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 behaviour 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 instants 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 or processes with a long time delay, and can efficiently deal with constraints. 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 Mahfonf, 1994).

Next section describes first the basic principles usually found in classical MBPC. The 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 Section 4.1.2. When the model is nonlinear and in the presence of constraints the optimization problem becomes more complex. Possible solutions for this problem are presented in Section 4.1.3. Section 4.1.4 addresses problems related to output errors caused e.g. by model-plant mismatches or disturbances.
4.1.1 Basic definitions

Control and prediction horizons. The future plant outputs for a determined prediction horizon $H_p$ are predicted at each time instant $k$ using a model of the process. The predicted output values $\hat{y}(k+j), j = 1, \ldots, H_p$, depend on the states of process at the current time $k$ (given, for instance, by the past input and outputs) and on the future control signals $u(k+j), j = 1, \ldots, H_c$, where $H_c$ is the control horizon. The control signals change only inside the control horizon, remaining constant afterwards,

$$u(k+j) = u(k+H_c-1), \quad j = H_c, \ldots, H_p - 1.$$ (4.1)

The basic principle of model-based predictive control is depicted in Fig. 4.1. 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 6), this number can be slightly reduced, alleviating 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, this time can be slightly reduced due to the flexibility introduced by the fuzzy goals, see Section 6.3.

**Figure 4.1. Basic principle of predictive control.**

Objective function. The sequence of future control signals is obtained by the optimization of a determined objective function, which describes the control goal. In classical MBPC, the objective functions are usually given by the following quadratic form:

$$J(u) = \sum_{i=1}^{H_p} \alpha_i(w(k+i) - \hat{y}(k+i))^2 + \beta_i(\Delta u(k+i-1))^2,$$ (4.2)
or some small modifications of it, where \( \hat{y} \) are the predicted process outputs, \( w \) is the shaped reference trajectory (see paragraph below), and \( \Delta u \) is the change in the control signal, weighted by the parameters \( \beta_i \). Thus, the first term in (4.2) 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 \( u \), 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 presented in Eq. (4.2). The parameters \( \alpha_i \) and \( \beta_i \) 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 over a general form of classical objective functions in predictive control is presented in Section 6.3.1. Other terms, commonly called 'soft constraints', can also be considered in the classical objective function (4.2). Note that for systems with input time delays, only outputs from the time instant \( k \) plus the considered delay until \( H_p \) must be considered in (4.2), because only these outputs can be influenced by the actual control actions \( u(k) \). For nonminimum phase systems, the first steps including the nonminimum phase behaviour 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, leading still to controllers with good performance. An example of a linear nonminimum phase system using fuzzy criteria is presented in Section 6.3; this section 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 the various terms by different ways than just by the sum of the terms, as in (4.2).

**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-based predictive control scheme. In the optimization of the objective function as in Eq. (4.2), the reference trajectory utilized \( w(k + i) \) is sometimes different from the real reference \( r(k + i) \). Normally, a smooth approximation from the actual value of the outputs towards the known reference is considered. This shaped reference \( w(k + i) \) is usually approximated by means of a first order system:

\[
\begin{align*}
w(k) &= y(k) \\
w(k + i) &= \lambda w(k + i - 1) + (1 - \lambda) r(k + i), & i = 1, \ldots, H_p. 
\end{align*}
\]

(4.3)

The parameter \( \lambda \) is in the interval \([0, 1]\), where the closer to one, the smoother is the shaped reference, and it influences the dynamic response of the system. Other forms of shaping the reference can be used (Clarke and Mohdadi, 1989). This shaped reference avoid sudden changes in the control actions and local unstability of the system, at the cost of slower responses.

**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 \( \hat{y}(k+1) \) are different from the process outputs \( y(k+1) \). For this reason, only the control signals \( u(k) \) are applied to the process. The control signals at the next sampling periods \( u(k+1), \ldots, u(k+H_p - 1) \) are discarded, because at the next sampling instant the process outputs \( y(k+1) \) are known and the optimization can be repeated using the updated data. The new \( u(k+1) \) 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 Section 4.1.4.

**Classical MBPC scheme** As the process outputs are feedback 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. 4.2. 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.

![Control Algorithm](image)

**Figure 4.2. Classical model-based 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 Section 6.2.
4.1.2 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 decreases also. 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 as in (4.2) are used.

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 can not derive reliable models for complex or partly known systems. For these types of problems, fuzzy models, as presented in Section 2.2 can be used advantageously. This thesis uses this type of modeling technique to derive nonlinear models. Linear models are used sometimes in the thesis, such as in Section 6.3.2, in order to emphasize the advantages of MBPC with fuzzy objective functions on some situations, even if a linear system is considered.

4.1.3 Optimization problems

If a linear model of the system is used in the control algorithm, the objective function is described by (4.2), or a similar quadratic equation, and no constraints are active, the optimization problem has an analytical solution. The different parameters in (4.2), such as control and prediction horizons $H_c$ and $H_p$, and the weight factors $\alpha_i$ and $\beta_i$, must still be tuned. When any constraint is violated, but the other two conditions remain, no analytical solution is available. The optimization problem results then in a quadratic problem (QP) to be solved at each time instant (Camacho and Bordons, 1995). This nonlinear optimization problem is convex and can be solved using fast gradient-descent methods with a guaranteed global solution.

However, in the most general case both nonlinear models and constraints are present, and the optimization problem results in a non-convex problem. 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 hamper usually the application of MBPC to fast systems because these iterative methods have generally 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 results usually 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 in a discrete optimization problem, where techniques such as branch-and-bound or genetic algorithms can be applied. Chapter 7 discusses the implementational problems of classical and fuzzy MBPC.
4.1.4 Compensation of model-plant mismatch and disturbances

The MBPC scheme, as presented in Fig. 4.2, can present problems to deal with model-plant mismatches 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, as e.g. generalized predictive control (Clarke, et al., 1987), which can reduce significantly their influence. 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 more 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 presented in Section 4.2. Fuzzy compensation introduced in Section 4.3 is a new compensation 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.

4.2 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, can however induce steady-state errors due to model-plant mismatches or disturbances, depending on the number of integrators in the process, the type of disturbances, e.g., off-set or process disturbances, and the required reference following accuracy. Beyond shaping the reference that can cope with this problem to some extend, a scheme is needed to compensate for these errors.

4.2.1 Classical internal model control

A classical approach 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 undesired oscillations, and if it is too small the system converges quite slowly to the setpoint. 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 and 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 was first
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. 4.3, where $P$ is a mapping describing 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.

![Figure 4.3. General internal model control scheme.](image)

The disturbances are separated in process disturbances $d$ and the measurement (additive) disturbances $d_m$, according to the general block diagram presented in Fig. 1.1. The output is simply given by $y = P u$. Supposing that there is no filter ($F = 1$) the following relationships can be derived from Fig. 4.3:

\[
\begin{align*}
  e &= r - y + y_m \quad (4.4a) \\
  y_m &= M C e \quad (4.4b) \\
  e_m &= (P - M) C e \quad (4.4c)
\end{align*}
\]

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 (4.4) (Economou, et al., 1986):

**Proposition 4.2.1 (Stability)** *If $P$ and $C$ are input-output stable, and if a perfect model of the plant is available, i.e., $M = P$, then the closed-loop system is input-output stable too.*
4.2 Nonlinear Internal Model Control

Proposition 4.2.2 (Perfect control) If the right inverse of the model operator $M^r$ exists, $C = M^r$, and the closed-loop system is input-output stable with this controller, then the control is perfect, i.e., $y = r$.

Proposition 4.2.3 (Zero offset) If the right inverse of the steady-state model operator $M^r_\infty$ exists, $C = M^r_\infty$, and the closed-loop system is stable with this controller, then for asymptotically constant inputs offset free control is achieved.

Proposition 4.2.1 can be proved easily, since the feedback loop has no influence when $M = P$, $e_m = 0$, and the system is in open loop. As both controller and process are open loop stable, the global system is also stable. Proposition 4.2.2 is a direct consequence of equations (4.4a) and (4.4b), when $C = M^r$. Finally, Proposition 4.2.3 is also a result from the direct application of equations (4.4a) and (4.4b), 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 alleviates the design problems for systems that are input-output stable or stabilizable by output feedback. For systems with these characteristics, and when a good model of the plant is available, Proposition 4.2.2 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 unmeasured plant disturbances, as suggested by Properties 4.2.2 and 4.2.3.

The filter $F$ in Fig. 4.3 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 present undesired overshoots or even oscillations.

4.2.2 MBPC in an internal model control scheme

The model-based predictive controller can be incorporated in the internal model control scheme, as presented in Fig. 4.4. 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 was first presented for inverse control, as in Section 4.2.1, but note that predictive control can be regarded as a generalization of inverse control. In fact, when \( H_c = H_p = 1, \alpha_i = 1, \beta_i = 0 \) in (4.2), and a control command exists such that \( y(k+1) = r(k+1) \), a global optimum of the objective function (4.2), 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 Section 4.1.3. 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. A different way of coping with model-plant mismatches and disturbances is presented in the next section.

![Predictive controller diagram](image)

**Figure 4.4. MBPC in an internal model control scheme.**

### 4.3 Fuzzy compensation of steady-state errors

In the previous section it was seen that the linear feedback used in the IMC scheme can deteriorate the closed loop dynamics in the presence of highly nonlinear systems. A new solution called **fuzzy compensation** is introduced in Sousa, et al. (1996a), and it can compensate for steady-state errors, based on the information contained in the model of the system. A fuzzy set is defined for the steady-state error which 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,
4.3 Fuzzy compensation of steady-state errors

Fuzzy compensation makes explicit use of a nonlinear model of the process. Next section describes the process in detail, and an application example is presented afterwards.

4.3.1 Fuzzy compensation

For the sake of simplicity, the method is presented for nonlinear discrete-time SISO systems, but it can be extended for MIMO systems. The general model of a system is given by (2.4).

In this section it is convenient to delay one step the model for notation simplicity. The discrete-time SISO regression model of the system under control is then given by:

\[ y(k) = f(x(k-1)) \]

(4.5)

where \( x(k-1) = [y(k-1), \ldots, y(k-p), u(k-1), \ldots, u(k-m)] \) is the state containing the lagged model outputs and inputs given by \( y(k-1), \ldots, y(k-p) \) and \( u(k-1), \ldots, u(k-m) \), respectively.

Fuzzy compensation uses a correction action called \( u_c(k) \) which is added to the action derived from a (model-based) controller, \( u_m(k) \), as shown in Fig. 4.5. The total control signal applied to the process is thus given by,

\[ u(k) = u_m(k) + u_c(k) \]

(4.6)

The controller in Fig. 4.5 can be any controller that can control the system, such as a predictive controller, but still yields small steady-state errors at the output. Taking into account the noise and a (small) offset error, a fuzzy set \( C \) defines the region where the compensation is active, see Fig. 4.6. The error is defined as \( e(k) = r(k) - y(k) \), and the membership function \( \mu_C(e(k)) \) is designed to allow for steady-state error compensation whenever the support of \( \mu_C(e(k)) \) is not zero. The value of \( B \) should be a superior limit of the absolute value of the possible steady-state errors. Fuzzy compensation is fully active in the interval \([-B, B]\). The support of \( \mu_C(e(k)) \) should be chosen such that it allows
Figure 4.6. Definition of the fuzzy boundary $C$ where fuzzy compensation is active.

for a smooth transition from enabled to disabled compensation. This smoothness of $C$ induces smoothness on the fuzzy compensation action $u_c(k)$, and avoids abrupt changes in the control action $u(k)$.

The compensation action $u_c(k)$ at time $k$ is given by

$$u_c(k) = \mu_C(e(k)) \left( \sum_{i=0}^{k-1} u_c(i) + K e(k) f_u^{-1} \right), \quad (4.7)$$

where $\mu_C(e(k))$ is the error membership degree at time $k$, $K$ is a constant and

$$f_u = \left[ \frac{\partial f}{\partial u(k-1)} \right]_{x(k-1)} \quad (4.8)$$

is the partial derivative of the function $f$ in (4.5) with respect to the control action $u(k-1)$, for the present state of the system $x(k-1)$. Comparing (4.7) with a classical integral action, there are two new terms: $\mu_C(e(k))$, whose effect was already described, and the term (4.8), 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 $K$, but for highly nonlinear systems, the compensation can be largely improved taking this factor into account. As the partial derivative in (4.8) 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 (4.8) must be considered in the compensation action (4.7). For linear systems, the term (4.8) is constant and the parameter $K$ multiplied by the term $f_u^{-1}$ can be seen as the integral gain in a PID controller. The parameter $K$ 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 $e(k+1)$ of opposite sign from the previous $e(k)$.

When the model of the system $f$ is available, the partial derivative (4.8) 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
4.3 Fuzzy compensation of steady-state errors

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 points can be computed as,

\[
\left[ \frac{\partial f}{\partial u(k-1)} \right]_{x(k-1)} = \frac{\left[ \frac{\partial f}{\partial u(k-1)} \right]_{x(k-1)+} + \left[ \frac{\partial f}{\partial u(k-1)} \right]_{x(k-1)-}}{2}
\]

This approximation does not deteriorate significantly the control performance, as it can be seen from the simple example of a nonlinear system with dead-zone presented in the next section.

4.3.2 Application: system with a 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 \( \hat{y}(k+1) = f(y(k), u(k)) \). The model of the system is illustrated in Fig. 4.7. 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. 4.7 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.

![Figure 4.7. Model of the system.](image)

The process is simulated to generate input-output data and a fuzzy model with singleton consequents, as described in Section 2.2.2, is derived for the system. The model of the process is described by the lookup table presented in Table 4.1. The values for the output \( y(k) \) and the control input \( u(k) \) 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 \( \hat{y}(k+1) \). The range of the output of the system is \( y(k) \in [-7, 3] \), and the range of the control actions is \( u(k) \in [-1, 1] \). Only two values are needed for the process output \( y(k) \) because the system is linear with respect to this variable, and thus, the linear interpolation between the points in the table describes completely the system. The
cores of the input $u(k)$ coincide with the nonlinearities of the process. The simulated system use a simple inverted model control technique, as presented in Section 5.1.3. 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 $u(k) = 0.2$ is divided in $0.199$ and $0.201$.

### Table 4.1. Fuzzy model with singleton consequents.

<table>
<thead>
<tr>
<th>$y(k)$</th>
<th>-1</th>
<th>-0.401</th>
<th>-0.399</th>
<th>0.199</th>
<th>0.201</th>
<th>0.5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7</td>
<td>-7.01</td>
<td>-5.60</td>
<td>-5.40</td>
<td>4.60</td>
<td>1.80</td>
<td>2.39</td>
<td>2.90</td>
</tr>
<tr>
<td>3</td>
<td>-6.20</td>
<td>-5.59</td>
<td>-5.10</td>
<td>0.90</td>
<td>2.40</td>
<td>2.60</td>
<td>3.00</td>
</tr>
</tbody>
</table>

In order to eliminate steady-state errors, nonlinear IMC and fuzzy compensation were added to the controller based on the fuzzy model inversion, following the schemes presented in Fig. 4.3 and Fig. 4.5. The parameter $K$ is chosen equal to 5 and the membership function $C$ is given by

$$
\mu_C(e(k)) = \begin{cases} 
0 & \text{when } |e(k)| > 1, \\
1 & \text{when } |e(k)| < 0.5, \\
2e(k) + 2 & \text{when } -1 < e(k) < -0.5, \\
-2e(k) + 2 & \text{when } 0.5 < e(k) < 1.
\end{cases}
$$

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 $C$, allowing for a smooth transition in the compensation control actions $u_c(k)$ as desired. The obtained results are shown in Fig. 4.8.

The advantage of using fuzzy compensation is clearly seen at the region from 350s to 400s, 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 both 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 $K$ 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. Therefore, the IMC scheme, though the referred disadvantage concerning the linear filter, is the control technique used in this thesis to cope with model-plant mismatches and disturbances.
4.4 Summary and concluding remarks

Model-based control techniques present some advantages over other control techniques, leading in general to increased robustness and performance, when a good and reasonably simple model of the process under control is available. MBPC is the control technique most widely used in this thesis. MBPC is characterized by the use of a process model to predict the process output at future discrete time instants over a specified prediction horizon, given the process inputs and the desired reference. The sequence of future control actions is computed by minimizing an objective function while satisfying a set of operation constraints. The optimization is repeated at every sample instant with the new available process data, and each time only the first control action in the calculated sequence is applied to the process (receding horizon principle). Sometimes, MBPC has problems to deal with model-plant mismatch and disturbances. A possible solution for this problem is to include MBPC in an internal model control scheme.

The IMC scheme uses the model information to compute the model-based control action. Contrary to MBPC, the current outputs of the process are not directly feedback to the controller. The error between the outputs of the model and the outputs of the system are used to shift the reference. Note that this error can be caused by either the referred model-plant mismatches or disturbances. This scheme 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 is to use fuzzy compensation, which was developed for steady-state errors resulting from model-plant mismatches and disturbances. Although the problems concerning the speed of the response and overshoots are usually overcome, fuzzy compensation can introduce undesired oscillations. Experiments showed that this control scheme is,
in general, less robust then IMC. Taking this fact into account, the IMC scheme is further used in this thesis to cope with model-plant mismatches and disturbances.
Fuzzy Model-Based Control Using Fuzzy Models

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 directly deal 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 thesis, because fuzzy relational models are computationally more complex than linguistic or TS fuzzy models, implying larger computational times, and loss of linguistic meaning for the fuzzy rules in the model. In (Graham and Newell, 1988) an adaptive fuzzy controller based on fuzzy relational models is applied to a laboratory-scale liquid level rig. Driankov et al. (1993) present another example of application of model-based control methodologies. Local design techniques derived from linear control theory have been applied to Takagi-Sugeno models with linear consequents (Kuijpers and Åström, 1994; Sugeno and Takagi, 1983; Tanaka and Sugeno, 1992; Palm et al., 1997).

The simplest way to control a process using a fuzzy model is, however, to invert the model and use it in an open-loop (feed-forward) 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 must be added to the controller in order to choose the best control action at a given moment. This "ideal" control configuration cannot 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. Summarizing, 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 Section 5.1. 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 5.2. A simulation example of a fermenter covering all the proposed control schemes is presented in Section 5.3. 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 the air-conditioning system presented in Chapter 8.

### 5.1 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 the model $M$ mapping the control actions $u$ to the system's outputs $y$, the control actions are simply given by $u = M^{-1}r$, where $r$ are the references to be followed (see Fig. 5.1).

![Figure 5.1. Mapping and perfect inversion of a system.](image)

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 situation of perfect control is impossible to achieve because 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, and disturbances
acting on the process are present and not taking into account in the controller. Further, when the system has a delay of $d$ steps, the inverse 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 were treated in Chapter 4.

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 2. 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 Section 5.1.1. The most common methods to invert fuzzy models are presented in Section 5.1.2. Fuzzy models with certain structures can be exactly inverted, and this inversion can be used for control purposes. This thesis 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 Section 5.1.3. 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 $u(k)$, which inversion is described in Section 5.1.4. Constraining the model to be affine on $u(k)$, causes usually the effect of reducing the model accuracy.

Both these inversions are computationally very fast allowing for its use 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 Section 5.1.5, 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.

### 5.1.1 Problem definition

Let a global MIMO fuzzy model be considered. For the structure presented in Section 2.1, any MIMO model can be decomposed in $p$ MISO models without lack of generality. The MISO fuzzy system with $n$ states given by (2.4), can be represented as in Fig. 5.2. Note that the state variables $x_i$ in (2.4) are for input-output models, and they can be, in general, different inputs or outputs at delayed times of the process.

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. 5.3a. 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. 5.3b).

The partial inversion has usually an 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 an input \( u_i, i = 1, \ldots, m \) of the original system. Let this input be called \( u_i \Delta x_i = u \), 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, like, 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. The partial inversion is the only one used in this thesis. For the sake of simplicity, partial inversion is often simply called inversion in the rest of the thesis.

Note that partial inversion can only be applied if 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

\[ y = f(u, x_2, \ldots, x_n), \]  

(5.1)
and the inverted function exists, such that
\[ u = f^{-1}(y, x_2, \ldots, x_n). \] (5.2)

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 Sections 5.1.3 and 5.1.4, respectively.

### 5.1.2 Inversion methods

Several methods can be applied to obtain the inverse model of a given process (Boullart, et al., 1992; Hunt, et al., 1992), but the following two are the most utilized:

1. Identification of the inverse model from input-output data.
2. Inversion of the original model.

The first method is maybe 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 and Isermann, 1996).

![Diagram](image)

(a) Direct inverse learning.  
(b) Specialized inverse learning.

**Figure 5.4. Inverse learning.**

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. 5.4a. Different identification algorithms can be used to derive the inverse model, as the one developed by Nelles and Fischer (1996). Following Hunt, et al. (1992), two major drawbacks can be found in this approach. 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, as the excitation of the system is introduced as the activation of $u$, 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. 5.4b, or with a forward plant model. The parameters of the inverse model $M^{-1}$ 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 and Isermann (1996) 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 can not 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 Section 4.2.

5.1.3 Inversion of a singleton fuzzy model

The inversion of singleton fuzzy models was introduced by Babuška, et al. (1995a), and it is developed in (Babuška, 1997). A special structure of the singleton fuzzy model, which is presented in this section, is necessary to perform this inversion.

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
5.1 Inversion of fuzzy models

measurements. A general fuzzy rule $R_i$ has the following form:

$$
R_i : \text{If } y(k) \text{ is } A_{i1} \text{ and } \ldots \text{ and } y(k-p+1) \text{ is } A_{ip} \text{ and } u(k) \text{ is } B_{i1} \text{ and } \ldots \text{ and } u(k-m+1) \text{ is } B_{im} \text{ then } \dot{y}(k+1) = c_i, \quad i = 1, 2, \ldots, K,
$$

(5.3)

where $A_{i1}, \ldots, A_{ip}$ and $B_{i1}, \ldots, B_{im}$ are fuzzy sets and $c_i$ are singletons. Note that this model can represent a nonlinear system as a collection of local linear models. To simplify the notation, the rule index will be omitted in the sequel. The considered fuzzy rule is then given by the following expression:

$$
\text{If } y(k) \text{ is } A_1 \text{ and } y(k-1) \text{ is } A_2 \text{ and } \ldots \text{ and } y(k-p+1) \text{ is } A_p \text{ and } u(k) \text{ is } B_1 \text{ and } u(k-1) \text{ is } B_2 \text{ and } \ldots \text{ and } u(k-m+1) \text{ is } B_m \text{ then } \dot{y}(k+1) = c.
$$

(5.4)

Let a state vector $\mathbf{x}(k)$ containing the $m-1$ past inputs, the $p-1$ past outputs and the current output, i.e., all the antecedent variables in (5.4) except $u(k)$, be defined as:

$$
\mathbf{x}(k) = [y(k), \ldots, y(k-p+1), u(k-1), \ldots, u(k-m+1)]^T.
$$

(5.5)

The fuzzy sets of $\mathbf{x}(k)$ are aggregated into a multidimensional fuzzy set $X$, by applying the Cartesian product:

$$
X = A_1 \times \ldots \times A_p \times B_2 \times \ldots \times B_m.
$$

By introducing the formal substitution of $B_1$ by $U$ in order to simplify the notation, the fuzzy rule (5.4) can be written as:

$$
\text{If } \mathbf{x}(k) \text{ is } X \text{ and } u(k) \text{ is } U \text{ then } \dot{y}(k+1) \text{ is } c.
$$

(5.6)

Note that the rule base (5.4) is equivalent to the rule base (5.6) since the order of the model dynamics is the same, taking into account that $\mathbf{x}(k)$ is a vector and $X$ is a multidimensional fuzzy set. Let $N$ denote the number of different fuzzy sets $X_i$ defined for the state $\mathbf{x}(k)$ and $M$ the number of different fuzzy sets $U_i$ defined for the input $u(k)$. If the rule base consists of all possible combination of $X_i$ and $U_j$ (the rule base is complete), the total number of rules is $K = N \cdot M$. The entire rule base can be represented as a table:

<table>
<thead>
<tr>
<th>$\mathbf{x}(k)$</th>
<th>$U_1$</th>
<th>$U_2$</th>
<th>$\ldots$</th>
<th>$U_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$c_{11}$</td>
<td>$c_{12}$</td>
<td>$\ldots$</td>
<td>$c_{1M}$</td>
</tr>
<tr>
<td>$X_2$</td>
<td>$c_{21}$</td>
<td>$c_{22}$</td>
<td>$\ldots$</td>
<td>$c_{2M}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$X_N$</td>
<td>$c_{N1}$</td>
<td>$c_{N2}$</td>
<td>$\ldots$</td>
<td>$c_{NM}$</td>
</tr>
</tbody>
</table>

(5.7)

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 $\beta_{ij}(k)$ is calculated as:

$$
\beta_{ij}(k) = \mu_{X_i}(\mathbf{x}(k)) \cdot \mu_{U_j}(u(k)),
$$

(5.8)
where \( \mu_{X_i}(x(k)) \) is the membership degree of a particular state \( x(k) \) in the fuzzy set \( X_i \) and \( \mu_{U_j}(u(k)) \) is the membership degree of an input \( u(k) \) in the fuzzy set \( U_j \).

The predicted output \( \hat{y}(k+1) \) of the model is computed by the fuzzy-mean defuzzification, where an average of the consequents \( c_{ij} \) is weighted by the normalized degrees of fulfillment \( \beta_{ij} \):

\[
\hat{y}(k+1) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \beta_{ij}(k) c_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \beta_{ij}(k)} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mu_{X_i}(x(k)) \mu_{U_j}(u(k)) c_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \mu_{X_i}(x(k)) \mu_{U_j}(u(k))}.
\] (5.9)

If this type of singleton fuzzy models has also triangular membership functions in the antecedents, and form a partition, i.e., \( \sum_{i=1}^{N} \mu_{X_i}(x) = 1, \forall x \), and \( \sum_{j=1}^{M} \mu_{U_j}(u) = 1, \forall u \), the above singleton model provides piece-wise linear interpolation between the rule consequents (Jager, 1995).

**Inversion of the singleton model.** The rule-based model (5.6) corresponds to a nonlinear regression model

\[
\hat{y}(k+1) = f(x(k), u(k)),
\] (5.10)

shown schematically in Fig 5.5a. The model inputs are the current state \( x \) and the current input \( u(k) \) and the output is the system’s predicted output at the next sampling instant \( \hat{y}(k+1) \).

\[\text{(a) Model of the system.} \quad \text{(b) Derived controller.} \]

**Figure 5.5. Fuzzy model and a controller based on the model inverse.**

Given the current system state \( x(k) \) and the desired system output (reference) at the next sampling time \( r(k+1) \), the objective of the control algorithm is to find \( u(k) \), such that the system output \( y(k+1) \) is as close as possible to the desired output \( r(k+1) \). This can be achieved by inverting the plant model, as indicated in Fig 5.5b, substituting the reference \( r(k+1) \) for \( \hat{y}(k+1) \) in the static function:

\[
u(k) = f^{-1}(x(k), r(k+1)).
\] (5.11)
The multivariate mapping of the fuzzy model (5.10) can be reduced to the univariate mapping \( \hat{y}(k+1) = f_x(u(k)) \) by making use of the model structure. The subscript \( x \) denotes that \( f_x \) is obtained for the particular state \( x(k) \). If the model is invertible, the inverse mapping \( u(k) = f_x^{-1}(r(k+1)) \) 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 (5.6) and the defuzzification method (5.9) is invertible if \( \forall x \) and \( \forall y \), a unique \( u \) exists such that \( y = f(x, u) \). In terms of the parameters of the model the monotonicity is translated into the following conditions:

\[
\text{card(} \text{core}(U_j) \text{)} = 1, \quad \forall j = 1, 2, \ldots, M, \text{ and } \tag{5.12a}
\text{core}(U_1) < \ldots < \text{core}(U_M) \rightarrow c_{i1} < c_{i2} < \ldots < c_{iM}, \text{ or }
\text{core}(U_1) < \ldots < \text{core}(U_M) \rightarrow c_{i1} > c_{i2} > \ldots > c_{iM}, \quad i = 1, 2, \ldots, N. \tag{5.12b}
\]

Here \text{card} denotes the cardinal of a set. The definition of \text{core} is given in Appendix A.

**Example 5.1.1** Figure 5.6 presents an example where both the above conditions are violated. Fuzzy set \( U_3 \) does not meet the condition \text{card(} \text{core}(U_3) \text{)} = 1; and for \text{core}(U_1) < \text{core}(U_2) \rightarrow c_{i1} < c_{i2}, \text{ while for } \text{core}(U_3) < \text{core}(U_4) \rightarrow c_{i3} > c_{i4}. \]

\[\begin{array}{c}
\text{Figure 5.6. Example where both condition for the singleton model invertibility fail.}
\end{array}\]

The inverse of the singleton fuzzy model can be formulated in the following theorem (Babuška, 1997).

**Theorem 5.1.1 (Inversion of the singleton fuzzy model)** Let the process be represented by the singleton fuzzy model (5.6) with the weighted-mean defuzzification method (5.9). Further, let the antecedent membership functions form a partition, i.e., let \( \sum_{i=1}^{N} \mu_{X_i}(x) = \)
1, \forall x, and \sum_{j=1}^{M} \mu_{U_j}(u) = 1, \forall u. At a certain time \( k \) the system is at the state \( x(k) \), and the inverse of the singleton model is given by the fuzzy rules:

\[
\text{If } r(k+1) \text{ is } C_j(k) \text{ then } u(k) \text{ is } U_j, \quad j = 1, 2, \ldots, M, \tag{5.13}
\]

where \( C_j \) are fuzzy sets that form a partition as in Figure 5.7.

The cores \( c_j \) of the fuzzy sets \( C_j \) are given by:

\[
c_j = \sum_{i=1}^{N} \mu_{X_i}(x(k)) c_{ij}, \quad j = 1, \ldots, M. \tag{5.14}
\]

The inference and defuzzification of the rules (5.13) is accomplished by the fuzzy-mean method:

\[
u(k) = \sum_{j=1}^{M} \mu_{C_j}(r(k+1)) \cdot \text{core}(U_j), \tag{5.15}\]

The open loop connecting the control action resultant from the inversion and the singleton fuzzy model gives an identity mapping (perfect control):

\[
y(k+1) = f_x(u(k)) = f_x(f_x^{-1}(r(k+1))) = r(k+1), \tag{5.16}\]

when \( u(k) \) exists, and thus \( r(k+1) = f(x(k), u(k)) \).

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure5_7.png}
\caption{Partition of fuzzy sets \( C_j \) built using the cores \( c_j \).}
\end{figure}

**Proof:** As the product \( t \)-norm is used both for the and connective in the rule antecedent and for the Mamdani inference, the rule (5.6) can be rewritten in the following way:

\[
\text{If } x(k) \text{ is } X_i \text{ then (if } u(k) \text{ is } U_j \text{ then } y(k+1) \text{ is } c_{ij}). \tag{5.17}
\]

The inversion is made for a given state \( x(k) \). The degree of fulfillment for this state given in the proposition, "\( x(k) \) is \( X_i \)," is denoted by \( \mu_{X_i}(x(k)) \). Then the \( N \) consequents of the rules containing a particular \( U_j \) (columns in (5.7)) can be aggregated. This aggregated
consequents $c_j(k)$ are given by eq. (5.14). As a result, the following set of $M$ rules is obtained:

$$\textbf{If } u(k) \text{ is } U_j \text{ then } \hat{y}(k + 1) = c_j(k), \quad j = 1, \ldots, M. \quad (5.18)$$

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

$$\textbf{If } \hat{y}(k + 1) = c_j(k) \text{ then } u(k) \text{ is } U_j \quad j = 1, \ldots, M \quad (5.19)$$

As the consequents $c_j(k)$ are singletons, an interpolation method must be applied to obtain $u(k)$. This interpolation is accomplished by the fuzzy sets $C_j$ defined as in Figure 5.7. The serial connection of the inverse and the model gives an identity mapping if the desired reference $r(k + 1)$ is in the range of the reached states from the actual state $x(k)$, i.e. $c_1 \leq r(k + 1) \leq c_M$. Figure 5.8 shows that the inversion of the reference $r(k + 1)$ is given by one and only one point for the described conditions. Moreover, the direct model giving $u(k)$ as input reaches $y(k + 1) = r(k + 1)$ for the same state $x(k)$.

**Figure 5.8.** Unique mapping between $u(k)$ and $r(k + 1)$ for the singleton fuzzy model considered.

If the desired output cannot be reached from the current state in one time step, i.e. $r(k + 1) < c_1$ or $r(k + 1) > c_M$, the control action is still the mapping with the minimal error. In the case that $r(k + 1) > c_M$, $\mu_{c_M}(r(k + 1)) = 1$, and the control input is $u(k) = \text{core}(U_M)$. The degree of fulfillment for the control action is given by $\mu_{U_M}(u(k)) = 1$, which yields
the model output $y(k+1) = c_M$. As $c_M > c_j$, $1 \leq j \leq M - 1$, the difference $|r(k+1) - y(k+1)|$ is the minimum possible. A similar situation occurs for $r < c_1$, where the control action $u(k) = \text{core}(U_1)$ yields the output $y(k+1) = c_2$. This is the best control action, since $c_1 < c_j$, $2 \leq j \leq M$. \hfill \Box

A simple example of the inversion of a singleton model is presented in Example 5.1.2. For fuzzy models with input delays $\hat{y}(k+1) = f(x(k), u(k-d))$, the inversion cannot be applied directly since in that case, the control law (5.13) would compute a control action $u(k-d)$ which is $d$ steps delayed. To generate the appropriate control action $u(k)$, the inverse model must be applied to a state $x(k+d)$, $d$ samples ahead, i.e., $u(k) = f^{-1}(r(k+1+d), x(k+d))$, with

$$x(k+d) = [\hat{y}(k+1), \ldots, y(k), \ldots, y(k-p+1+d), u(k-1), \ldots, u(k-m+1)].$$  

(5.20)

The unknown values $\hat{y}(k+1), \ldots, \hat{y}(k+d)$ are predicted using the fuzzy model: $\hat{y}(k+i) = f(x(k+i-1), u(k+i-1-d))$, for $i = 1, \ldots, d$. 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 $x$ are used in the subsequent steps.

**Example 5.1.2** Consider a model of the form $\hat{y}(k+1) = f(y(k), u(k), u(k-1))$ where two linguistic terms $\{\text{low, high}\}$ are used for $y(k)$, and three terms $\{\text{small, medium, large}\}$ for $u(k)$ and $u(k-1)$. Therefore the model rule base consists of $2 \times 3 \times 3 = 18$ rules in total:

If $y(k)$ is low and $u(k)$ is small and $u(k-1)$ is small then $\hat{y}(k+1)$ is $c_{11}$

If $y(k)$ is low and $u(k)$ is small and $u(k-1)$ is medium then $\hat{y}(k+1)$ is $c_{21}$

\ldots

If $y(k)$ is high and $u(k)$ is large and $u(k-1)$ is large then $\hat{y}(k+1)$ is $c_{63}$

In this example $x(k) = [y(k), u(k-1)]$, $N = 6$ and $M = 3$. The rule base can be represented by the following table:

<table>
<thead>
<tr>
<th>$x(k)$</th>
<th>$u(k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>small</td>
</tr>
<tr>
<td>$X_1$ (low × small)</td>
<td>$c_{11}$</td>
</tr>
<tr>
<td>$X_2$ (low × medium)</td>
<td>$c_{21}$</td>
</tr>
<tr>
<td>$X_3$ (low × large)</td>
<td>$c_{31}$</td>
</tr>
<tr>
<td>$X_4$ (high × small)</td>
<td>$c_{41}$</td>
</tr>
<tr>
<td>$X_5$ (high × medium)</td>
<td>$c_{51}$</td>
</tr>
<tr>
<td>$X_6$ (high × large)</td>
<td>$c_{61}$</td>
</tr>
</tbody>
</table>

For a given state $x(k) = [y(k), u(k-1)]$, the degree of fulfillment of the first antecedent proposition "$x(k)$ is $X$", is calculated as $\mu_{x_i}(x(k))$. Using (5.14), the consequents $c_j(k)$ are given by:

$$c_j(k) = \frac{\sum_{i=1}^{6} \mu_{x_i}(x(k)) c_{ij}}{\sum_{i=1}^{6} \mu_{x_i}(x(k))}, \quad j = 1, 2, 3,$$

(5.21)
resulting in the following three rules:

- If \( u(k) \) is \( U_1 \) then \( \dot{y}(k + 1) \) is \( c_1(k) \),
- If \( u(k) \) is \( U_2 \) then \( \dot{y}(k + 1) \) is \( c_2(k) \),
- If \( u(k) \) is \( U_3 \) then \( \dot{y}(k + 1) \) is \( c_3(k) \),

An example of membership functions \( C_j(k), j = 1, 2, 3 \), of the fuzzy partition created using the consequent singletons \( c_1(k), c_2(k), c_3(k) \) is shown in Fig. 5.9. Assuming that fuzzy rule base is monotonic, the rules can be inverted resulting in:

- If \( r(k + 1) \) is \( C_1(k) \) then \( u(k) \) is \( U_1 \),
- If \( r(k + 1) \) is \( C_2(k) \) then \( u(k) \) is \( U_2 \),
- If \( r(k + 1) \) is \( C_3(k) \) then \( u(k) \) is \( U_3 \).

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 done 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 Section 5.2. Therefore, in these situations it is preferable to use the predictive control scheme as in Section 5.2, which gives an optimal control solution directly, and the inversion of non-monotonous systems is not considered further in this thesis.

### 5.1.4 Inversion of an affine Takagi–Sugeno fuzzy model

This section presents the inversion of a TS fuzzy model affine with respect to a control action. This section is an extension of this inversion presented in Sousa, et al. (1997c). Let a MIMO fuzzy system be given as in (2.4). 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 Section 5.1.1. Denote $y \triangleq y_1$ and $u \triangleq u_1$ for the ease of notation. Let the control action $u(k) = u_1(k)$ not be considered in the state vector (2.3), and thus the vector of states is now given by

$$x(k) = [y_1(k), \ldots, y_1(k - p_1 + 1), \ldots, y_p(k), \ldots, y_p(k - p_p + 1),$$

$$u_1(k - 1), \ldots, u_1(k - m_1 + 1), u_2(k), \ldots, u_2(k - m_2 + 1), \ldots,$$

$$u_m(k), \ldots, u_m(k - m_m + 1)]^T.$$  \hspace{1cm} (5.22)

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:

$$\hat{y}(k + 1) = f(x(k), u(k)).$$ \hspace{1cm} (5.23)

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 (2.3). The dimension of the state vector is now given by $n = \sum_{j=1}^m m_j + \sum_{j=1}^p p_j - 1$.

**TS fuzzy model.** The unknown function $f$ in (5.23) is parameterized by the Takagi–Sugeno (TS) fuzzy model (Takagi and Sugeno, 1985), which can approximate a large class of nonlinear systems, see Section 2.2.3. In these 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:

$$R_i: \text{If } y_1(k) \text{ is } A_{i1} \text{ and } \ldots \text{ and } y_1(k - p_1 + 1) \text{ is } A_{ip_1} \text{ and}$$

$$y_2(k) \text{ is } A_{i(p_1+1)} \text{ and } \ldots \text{ and } y_2(k - p_2 + 1) \text{ is } A_{i(p_1+p_2)} \text{ and } \ldots \text{ and}$$

$$y_p(k) \text{ is } A_{i(p_1+\ldots+p_{p-1}+1)} \text{ and } \ldots \text{ and } y_p(k - p_p + 1) \text{ is } A_{i(p_1+\ldots+p_p)} \text{ and}$$

$$u_1(k) \text{ is } B_{i1} \text{ and } \ldots \text{ and } u(k - m_1 + 1) \text{ is } B_{im_1} \text{ and } \ldots \text{ and}$$

$$u_m(k) \text{ is } B_{i(m_1+\ldots+m_{m-1}+1)} \text{ and } \ldots \text{ and } u_m(k - m_m + 1) \text{ is } B_{i(m_1+\ldots+m_m)} \text{ then}$$

$$\hat{y}_i(k+1) = \sum_{j=1}^{p_1} a_{ij} y_1(k - j + 1) + \ldots + \sum_{j=1}^{p_p} a_{ij}^p y_p(k - j + 1) +$$

$$\sum_{j=1}^{m_1} b_{ij}^1 u_1(k - j + 1) + \ldots + \sum_{j=1}^{m_m} b_{ij}^m u_m(k - j + 1) + c_i,$$

$$i = 1, \ldots, K,$$ \hspace{1cm} (5.24)

where $A_{ij}$, $B_{ij}$ are fuzzy sets, $a_{ij}^1, \ldots, a_{ij}^p, b_{ij}^1, \ldots, b_{ij}^m$ and $c_i$ 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:

$$\hat{y}_i(k+1) = \sum_{l=1}^{p} \left( \sum_{j=1}^{p_l} a_{ij}^l y_l(k - j + 1) \right) + \sum_{l=1}^{m} \left( \sum_{j=1}^{m_l} b_{ij}^l u_l(k - j + 1) \right) + c_i.$$ \hspace{1cm} (5.25)
The fuzzy rule base given in (5.24) can be expressed in a compact way by using the state vector \( \mathbf{x}(k) \) and the control action \( u(k) \):

\[
R_i: \quad \text{If } [\mathbf{x}(k), u(k)] \text{ is } A_i \text{ then } \dot{y}_i(k + 1) = a_i^T \mathbf{x}(k) + b_i u(k) + c_i, \quad i = 1, 2, \ldots, K, \tag{5.26}
\]

where \( A_i \) is the antecedent multidimensional fuzzy set, defined by its membership function:

\[
\mu_{A_i}(\mathbf{x}(k), u(k)) : \mathbb{R}^{n+1} \rightarrow [0, 1], \tag{5.27}
\]

resultant from the Cartesian product of all the \( A_{ij} \), for \( j = 1, 2, \ldots, \sum_{l=1}^{P} p_{l} \), and \( B_{il} \), for \( l = 1, 2, \ldots, \sum_{l=1}^{m} m_{l} \). Thus, the antecedent fuzzy set \( A_i \) is defined as

\[
A_i = A_{i1} \times \ldots \times A_{ip_i} \times \ldots \times A_{i(p_i+p_{i+1})} \times B_{i1} \times \ldots \times B_{im_{1}} \times \ldots \times B_{i(m_{1}+\ldots+m_{m})}. \tag{5.28}
\]

The consequent parameters of the \( i \)th rule, \( a_i \in \mathbb{R}^n \), and \( b_i, c_i \in \mathbb{R} \) are related to the consequents in (5.24) by:

\[
a_i = [a_{i1}^1, a_{i2}^1, \ldots, a_{ip_i}^p, b_{i1}, \ldots, b_{im_m}^m]^T, \]

\[
b_i = b_{i1}, \quad c_i = c_i. \tag{5.29}
\]

The output of the model \( \dot{y}(k + 1) \) is computed by:

\[
\dot{y}(k + 1) = \sum_{i=1}^{K} \beta_i(k) (a_i^T \mathbf{x}(k) + b_i u(k) + c_i), \tag{5.30}
\]

where \( \beta_i(k) \) is the normalized degree of fulfillment of the \( i \)th rule’s antecedent:

\[
\beta_i(k) = \frac{\mu_{A_i}(\mathbf{x}(k), u(k))}{\sum_{j=1}^{K} \mu_{A_j}(\mathbf{x}(k), u(k))}. \tag{5.31}
\]

**Inversion of the TS fuzzy model.** In order to invert the fuzzy model, i.e., to compute \( u(k) \) based on the current state \( \mathbf{x}(k) \) and on the desired future output \( r(k + 1) \), the general fuzzy model (5.26) can be described by the simplified affine fuzzy model of the form:

\[
\dot{y}(k + 1) = f(\mathbf{x}(k)) + g(\mathbf{x}(k)) u(k). \tag{5.32}
\]

If the term in \( u(k) \) is not considered in the input, i.e. the input of the fuzzy model is just given by the state vector \( \mathbf{x}(k) \), this model is parameterized by the Takagi–Sugeno (TS) structure as follows:

\[
R_i: \quad \text{If } \mathbf{x}(k) \text{ is } A_i' \text{ then } \dot{y}_i(k + 1) = a_i'^T \mathbf{x}(k) + b_i u(k) + c_i, \quad i = 1, 2, \ldots, K, \tag{5.33}
\]

where \( \mu_{A_i'}(\mathbf{x}) : \mathbb{R}^n \rightarrow [0, 1] \) is the projection of the antecedent fuzzy set \( A_i \in \mathbb{R}^{n+1} \) in the space of the state vector \( \mathbf{x}(k) \in \mathcal{X} \subset \mathbb{R}^n \). This projection can be obtained by using the
Gustafson-Kessel algorithm, presented in Appendix B. Thus, the projected membership functions are given by

\[ \mu_{A'_i}(x(k)) = \frac{1}{\sum_{j=1}^{K} (d_{ik}/d_{jk})^{2/(m-1)}}, \tag{5.34} \]

with

\[ d_{ik} = (x(k) - v^x_i)^T M^x_i (x(k) - v^x_i). \tag{5.35} \]

Here \( v^x_i = [v_{i1}, \ldots, v_{in}]^T \) denotes the projection of the cluster center onto \( \mathcal{X} \). The norm-inducing matrix is given by

\[ M^x_i = \left[ \det(F^x_i) \right]^{1/n}(F^x_i)^{-1}, \]

where each \( F^x_i = [f_{ij}] \), \( 1 \leq i, j \leq q \) is the submatrix of \( F_i \). Since the antecedent partition of (5.33) is different from the one in (5.26), the optimal consequent parameters \( a_i, b_i \) and \( c_i \) must be re-estimated utilizing the least squares algorithm, as presented in Section 2.4.2, by using eq. (2.30) to (2.33), where the matrices \( \Gamma_i \) contain the membership degrees given by (5.34) and (5.35). The normalized degrees of fulfillment \( \beta'_i(k) \) for the affine TS fuzzy model are defined as

\[ \beta'_i(k) = \frac{\mu_{A'_i}(x(k))}{\sum_{j=1}^{K} \mu_{A'_j}(x(k))}. \tag{5.36} \]

The predicted output of the model \( \hat{y}(k + 1) \) is recalculated by:

\[ \hat{y}(k + 1) = \sum_{i=1}^{K} \beta'_i(k)(a^T_i x(k) + b_i u(k) + c_i), \tag{5.37} \]

As the antecedent of (5.33) does not include the input term \( u(k) \), the model output \( \hat{y}(k + 1) \) is affine in the input \( u(k) \). Thus, Eq. (5.37) can be easily divided in two terms:

\[ \hat{y}(k + 1) = \sum_{i=1}^{K} \beta'_i(k) [a^T_i x(k) + c_i] + \sum_{i=1}^{K} \beta'_i(k) b_i u(k) \tag{5.38} \]

This expression can be translated in the nonlinear affine form given in (5.32), with

\[ f(x(k)) = \sum_{i=1}^{K} \beta'_i(k) [a^T_i x(k) + c_i] \]
\[ g(x(k)) = \sum_{i=1}^{K} \beta'_i(k) b_i. \tag{5.39} \]

If the goal is that the model at time step \( k + 1 \) equals the reference output, i.e. \( \hat{y}(k + 1) = r(k + 1) \), the corresponding control input \( u(k) \) is computed by a simple algebraic manipulation on Eq. (5.32),

\[ u(k) = \frac{r(k + 1) - f(x(k))}{g(x(k))}. \tag{5.40} \]
5.1 Inversion of fuzzy models

In terms of Eq. (5.38) one obtains:

$$u(k) = \frac{\tau(k + 1) - \sum_{i=1}^{K} \beta_i(k)[a_i x(k) + e_i]}{\sum_{i=1}^{K} \beta_i(k) b_i}.$$  \hspace{1cm} (5.41)

Thus, similarly to a singleton fuzzy model, the TS fuzzy model affine on the control input, as in (5.33), can be exactly inverted, provided that the function $g(x(k))$ 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 Section 5.1.3, where for $d$ samples of delay, the inverse model must be applied $d$-samples ahead.

5.1.5 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, changing product mix, etc. 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 (Proczyk and Mamdani, 1979), a neuro-fuzzy controller with temporal backpropagation learning (Jang, 1992) or a self-learning fuzzy controller based on reinforcement learning (Berenji and Khedar, 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. 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-based 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 using some nonlinear optimization technique. Various approaches have been suggested in literature, such as error backpropagation (Jang, 1992), nonlinear programming (Babuska and Verbruggen, 1994b) or genetic algorithms (Klawonn, et al., 1994). Moreover, the identification of fuzzy models using product-space fuzzy clustering, as presented in Section 2.4, can be performed using new data from the system.
To cope with model-plant mismatch and disturbances when present, the on-line adaptation of the fuzzy model is done in the IMC scheme shown in Fig. 5.10, where the consequents of the fuzzy model are adapted.

**Figure 5.10. Adaptive internal model control scheme.**

The adaptation of singleton fuzzy models presented in this thesis was introduced by Sousa, et al. (1995a). The predicted output for the singleton model, as in Eq. (5.3), is given by Eq. (5.9), 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 (5.24) are also linear in the consequent parameters, the number of parameters to be tuned is very high \((K \cdot (n + 1))\), 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 in this thesis. The consequent parameters can be indexed by the rule number, and arranged in a column vector denoted \(\mathbf{c}(k) = [c_1(k), c_2(k), \ldots, c_K(k)]^T\), where \(K\) is the number of rules. Similarly, the normalized degrees of fulfillment of the rule antecedents are arranged in a column vector \(\varphi(k) = [\varphi_1(k), \varphi_2(k), \ldots, \varphi_K(k)]\). This normalized degrees of fulfillment are computed as:

\[
\varphi_i(k) = \frac{\beta_i(k)}{\sum_{j=1}^K \beta_j(k)} .
\]  

(5.42)

The consequent vector \(\mathbf{c}(k)\) is updated recursively using the standard least squares equation:

\[
\mathbf{c}(k) = \mathbf{c}(k-1) + \frac{R(k-1)\varphi(k)}{\lambda + \varphi^T(k)R(k-1)\varphi(k)} \left[ y(k) - \varphi^T(k)\mathbf{c}(k-1) \right] .
\]

(5.43)

where \(\lambda\) is a constant (forgetting factor) and \(R(k)\) is a covariance matrix updated as follows:

\[
R(k) = \frac{1}{\lambda} \left[ R(k-1) - \frac{R(k-1)\varphi(k)\varphi^T(k)R(k-1)}{\lambda + \varphi^T(k)R(k-1)\varphi(k)} \right]
\]

(5.44)

The forgetting factor \(\lambda\) influences the tracking capabilities of the adaptation algorithm. The smaller the \(\lambda\), the faster the consequent parameters adapt, not only to the process changes but also to disturbances and noise. Therefore, the choice of \(\lambda\) is problem-dependent. The
initial covariance is usually set to $R(0) = \alpha \cdot I$, where $I$ is a $K \times K$ identity matrix, and $\alpha$ is a large positive constant. Another possibility is to calculate the initial covariance $R(0)$ 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-based 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 readapt the parameters as the process state changes.

## 5.2 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 $u(k)$ due to physical or safety reasons. Constraints in state variables are also often found. Model-based predictive control, presented in Section 4.1, is a general control method which can deal with constraints of the system, allowing to find the optimal sequence of control actions $u(k), \ldots, u(k + H_p - 1)$ 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, given by Eq. (4.2), 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 to 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, as described in Section 5.1. The combination of both control schemes was introduced in Babuška, et al. (1995a) and Sousa, et al. (1995b). This 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 5.2.1 summarizes the described control strategy.

Algorithm 5.2.1 (Combination of predictive and inverse-model control)

**Step 1:** Apply inverse model control. Compute the control actions $u(k), \ldots, u(k + H_p - 1)$ over the prediction horizon and the predicted process outputs using the inverse fuzzy model for calculating the control output, and the fuzzy model for calculating 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 $u(k)$ computed in Step 1 to the process.

**Step 3:** Use predictive control. By using a given objective function, a suboptimal solution for the control actions is found. Apply $u(k)$ to the process.

The optimization performed in Step 3 is usually non-convex, as stated before. 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 moreover converge frequently to local minima, hampering the application of the combined control scheme described in Algorithm 5.2.1. 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-based predictive control is described in Chapter 7. The control scheme proposed in Algorithm 5.2.1 avoids the oscillations that occur usually due to the discretization of the control space, see Section 7.1. With constant or slowly varying refer-
ences, 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 here.

5.3 Example: pressure control of a fermentation tank

Inverse model control and the combination of inverse control with predictive control described in this chapter, are applied to a highly nonlinear pressure dynamics in a laboratory fermenter*. 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 45s, which allows for a sample time of $T = 5s$. The following nonlinear differential equation is used as the simulation model which describes the pressure dynamics:

$$\frac{dP}{dt} = \frac{1000\, RT}{22,4\, V_h} \cdot \left[ \Phi_g - (\pi R_H^2) \sqrt{\frac{2P_0}{\rho_0 K_f}} \ln\left(\frac{P}{P_0}\right) \right]. \quad (5.45)$$

The symbols represent the following quantities:

- $R$: the gas constant (8.134 J mol$^{-1}$K$^{-1}$),
- $T$: temperature (305 K),
- $V_h$: gas volume (0.015 m$^3$),
- $\Phi_g$: gas flow-rate ($3.75 \times 10^{-4}$ m$^3$s$^{-1}$),
- $R_H$: radius of the outlet pipe (0.0178 m),
- $P_0$: reference pressure ($1.013 \times 10^5$ N m$^{-2}$),
- $\rho_0$: 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 $\Phi_g$. The maximum changes in the valve position are $\Delta u(k) = -\Delta u(k) = 10\%$ of the total range per sample, and the level constraints are $u_{\text{min}} = 0\%$ and $u_{\text{max}} = 90\%$ of the valve position. More detailed descriptions of the process can be found in (van Can, et al., 1995).

*This application was made using data collected from the fermenter at the Kluyver Laboratory for Biotechnology, Delft University of Technology.
5.3.1 Fuzzy modeling

The inversion of singleton models presented in Section 5.1.3 and the inversion of a TS fuzzy model affine on the control action \( u(k) \) explained in Section 5.1.4, as well as the predictive control scheme based on the inversion of a fuzzy model, described in Section 5.2, 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.

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 Section 2.4. 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 \( u(k) \), the current pressure \( y(k) \), and the pressure at the next sampling instant \( y(k+1) \). The identified model is as follows:

1. If \( y(k) \) is LOW and \( u(k) \) is OPEN
   then \( \dot{y}(k + 1) = 0.67y(k) + 0.0007u(k) + 0.35 \)

2. If \( y(k) \) is MEDIUM and \( u(k - 1) \) is HALF CLOSED
   then \( \dot{y}(k + 1) = 0.80y(k) + 0.0028u(k) + 0.07 \)

3. If \( y(k) \) is HIGH and \( u(k - 1) \) is CLOSED
   then \( \dot{y}(k + 1) = 0.90y(k) + 0.0071u(k) - 0.39 \).

This rule base represents a nonlinear first-order regression model

\[
\dot{y}(k + 1) = f(y(k), u(k)).
\]

(5.46)

Figure 5.11a shows the membership functions for 'Open', 'Half Closed' and 'Closed' for the valve position, and Fig. 5.11b the membership functions found for the pressure, which are 'Low', 'Medium' and 'High'.

![Membership functions for the TS fuzzy model premise variables.](image)

Figure 5.11. 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 is in good agreement with the prior knowledge about the system.

![Membership functions for inputs](image)

**Figure 5.12. 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 and Verbruggen, 1995d). The membership functions obtained for the inputs with this method are presented in Fig. 5.12.

This 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 using the least-squares method. The rules obtained are presented in Table 5.1. In this table, the first rule, e.g., reads "If $y(k)$ is Low and $u(k)$ is Open then $\hat{y}(k+1) = 1.06$". The model is validated using the nonlinear output error, where only the real input of the system is used,
i.e. $\hat{y}(k + 1) = f(u(k), \hat{y}(k))$. Figure 5.13 presents the validation made on a different data set from the one used for identification.

![Figure 5.13. Validation of the singleton model (solid line – process, dash-dotted line – model).](image)

**Affine Takagi-Sugeno model.** First, a second-order TS fuzzy model in $u(k)$ is developed for the pressure system. This model is a straightforward extension of the model as in (5.46), and the process dynamics is represented by: $\hat{y}(k + 1) = f(y(k), u(k), u(k - 1))$. The additional term $u(k - 1)$ in the model is important to derive the affine TS fuzzy model, as it will be explained in the sequel. Figure 5.14a shows the membership functions fitted to the projection of the fuzzy partition onto the pressure antecedent variable. Figures 5.14b and 5.14c 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 $u(k)$ in the input, the TS fuzzy model remains affine in the input $u(k)$, as proved in Section 5.1.4. The consequent parameters are re-estimated by using the least-squares technique, and the following three rules are obtained:

1. **If** $y(k)$ is LOW and $u(k - 1)$ is OPEN
   **then** $\hat{y}(k + 1) = 0.76y(k) + 2.4 \cdot 10^{-3}u(k) + 2.7 \cdot 10^{-4}u(k - 1) + 0.15$

2. **If** $y(k)$ is MEDIUM and $u(k - 1)$ is HALF CLOSED
   **then** $\hat{y}(k + 1) = 0.75y(k) + 2.4 \cdot 10^{-4}u(k) + 3.1 \cdot 10^{-4}u(k - 1) + 0.27$

3. **If** $y(k)$ is HIGH and $u(k - 1)$ is CLOSED
   **then** $\hat{y}(k + 1) = 0.93y(k) + 8.2 \cdot 10^{-3}u(k) + 9.2 \cdot 10^{-4}u(k - 1) - 0.45$

Figure 5.14d shows the validation of the model by simulation from the inputs only, as it was done for the singleton fuzzy model. This model is slightly inferior to the singleton model presented in Fig. 5.13. Therefore, it is expected that the control based on this model presents also inferior performance. Next sections will apply both models in control.
5.3 Example: pressure control of a fermentation tank

5.3.2 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-principle model of the process given by the nonlinear differential equation (5.45) is used for simulations purposes. This simulation can be called realistic (te Braake, et al., 1994), since the model is highly nonlinear (as the process itself), including the rate and level constraints on u(k), 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 controller’s capability to cope with the process nonlinearities. 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 5.15 shows the simulation results with the controller based on the inverse model. The inverse model-based control can not 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 5.16
Figure 5.15. Fuzzy controller based on inverted fuzzy model.

Figure 5.16. Combination of predictive and inverse model control.
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 Section 7.1. 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: \( \Delta u(k) \in [-10, 0, 10] \). 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 way of control.

**Adaptive control.** The on-line adaptation mechanism presented in Section 5.1.5 is tested using an external disturbance. The flow-rate is increased from \( 3.75 \cdot 10^{-4} \) to \( 5 \cdot 10^{-4} \) [m³s⁻¹] at time \( t = 400 \) s. Figure 5.17 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 follows again 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 of the plant parameters. Let the consequent parameters

**Figure 5.17. Adaptive control of the pressure system.**

of the rules presented in Table 5.1 be denoted by \( c_i(k) \), for the \( i \)th rule, \( i = 1, \ldots, 16 \). The evolution of these consequent parameters is shown in Fig. 5.18, 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. 5.18b,
some parameters were adjusted only after a sudden step-like change of the reference, which occurs at time $t = 600s$. The forgetting factor was set to $\lambda = 0.98$ and the covariance matrix was initialized at $R = 100 \cdot I$, where $I$ is the identity matrix. The covariance matrix $R$ was automatically reset each 100 samples to guarantee permanent adaptation of the fuzzy model.

5.3.3 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 were 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 Section 5.3.2. The nonlinear internal control scheme is again used to cope with model-plant mismatch and noise disturbances. Figure 5.19 shows the simulation results with the controller based on the inverse model. The behaviour 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. 5.14d). The predictive control scheme based on inverted affine TS models uses the branch-and-bound algorithm for optimization with $\Delta u(k) \in [-10, 0, 10]$, as it was done for the singleton fuzzy model. The results of this control scheme are shown in Fig. 5.20. 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 it was when the inversion of the singleton model was combined with a predictive control scheme. However, the sum squared error between the pressure output and the desired references is decreased again, this
5.3 Example: pressure control of a fermentation tank

Figure 5.19. Fuzzy controller based on inverse fuzzy model.

Figure 5.20. Combination of predictive and inverse model control.
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 confirms the importance of having an accurate model to be used in the control scheme.

5.4 Summary and concluding remarks

Methods to derive 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 5.1.3 presented the inversion of singleton fuzzy models and Section 5.1.4 described the inversion of Takagi-Sugeno fuzzy models affine in the control action $u(k)$. 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 $u(k)$ 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 can not 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 presented in Section 4.2. 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 this phenomena. An adaptation algorithm based on recursive least-squares is presented in Section 5.1.5, 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 Section 5.2. The resultant optimization problem is usually non-convex, and algorithms to reduce the computational time by using discrete optimization techniques are proposed in Chapter 7. Compared to conventional fuzzy logic control, the controllers developed by using fuzzy models demand much less tuning effort. Though, some experimentation and iterative tuning may be required in the modeling phase. But once a fuzzy model of the process is available, it can be directly used in the control scheme. The application example presented in Section 5.3 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.
Fuzzy 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 Section 1.2, 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 (Sugeno, 1985; Terano, et al., 1994; Yen, et al., 1995). 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 are 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 particular approach to fuzzy model-based control, following closely the classical model-based control design approach, but making use of the fuzzy set theory in a higher level than usually made in FLC, where the fuzzy rules to control the system are given directly from expert knowledge. By this perspective approach the appropriate control actions are obtained by means of a multistage fuzzy decision making algorithm, as introduced by Bellman and Zadeh (1970). The first application in this field was presented by Yasunobu and Miyamoto (1985) in automatic train operation, using a linguistic description of the system. In fact,
this application is a mixture of both descriptive and perspective approaches to fuzzy control. A different approach called fuzzy multiobjective optimal control is presented by Jia and Zhang (1993b), but it is quite complex and difficult to implement in real-time. A good survey on this model-based approach to fuzzy control and decision making is presented by Kacprzyk (1997). However, in this book only open-loop control applications are reported. In fact, this approach is generally computationally intensive, which hampers its application in real-time control. This thesis pretends to present the first step in generalizing the fuzzy decision making approach to control in a simple but effective way, in order to be applied to a large number of industrial processes.

This chapter begins by describing fuzzy decision making (FDM) in Section 6.1. Fuzzy goals, fuzzy constraints and fuzzy decision are presented, and an approach to solve the optimization problem for fuzzy criteria defined in different sets is proposed. Section 6.2 presents FDM applied to control, where multistage fuzzy decision making is considered. Section 6.3 discusses the possible types of fuzzy criteria used for FDM in predictive control, where two illustrative examples are given. Finally, a discussion on the choice of aggregation operators for FDM, in general, and applied to control, is presented in Section 6.4.

### 6.1 Fuzzy decision making

Decision making is present in many human activities. Most of decision making in the real world usually takes place in an environment where the goals, the constraints and the consequences of the possible actions are fuzzy. Therefore, Bellman and Zadeh (1970) developed a framework called decision making in a fuzzy environment, where the goals and the constraints are fuzzy, but not necessarily the considered system. This approach called for brevity fuzzy decision making (FDM) was first applied to control, but several other applications were considered afterwards (Zimmermann, 1987). This thesis considers only the application of FDM to control, and especially to predictive control.

The paper of Bellman and Zadeh (1970) on decision making in a fuzzy environment has been the base for many research activities in different fields. Some additional classic references on fuzzy decision making are the works of Baas and Kwakernaak (1977) and Yager (1978). Several surveys on fuzzy decision making have been presented, and the most relevant ones can be found in Dubois and Prade (1980a), Zimmermann (1987) and Chen and Hwang (1992). These last authors provide a thorough systematic review of the existing fuzzy decision making methods. The basic concepts of fuzzy goal, fuzzy constraint and fuzzy decision are presented in Section 6.1.1. In general, a FDM problem has several goals and several constraints. The extension to multiple fuzzy goals and constraints is presented in Section 6.1.2. In the general case, goals and constraints can be defined in different sets or spaces, which makes the optimization problem quite complex. The formulation of this problem, and possible solutions for it are presented in Section 6.1.3.
6.1.1 Fuzzy goal, fuzzy constraint and fuzzy decision

The principal ingredients of a classic decision process are: a set of alternatives, a set of constraints on a certain choice between different alternatives, and a performance function associating with each alternative the cost resulting from the choice of that alternative. This scenario can be generalized for fuzzy decision making in a straightforward way, where the set of alternatives is maintained, the constraints are generalized to fuzzy constraints and the performance function to fuzzy goals. Both fuzzy goals and fuzzy constraints can be defined by fuzzy sets, which simplifies the computation of the fuzzy decision, as is shown in the sequel.

More formally, let \( \Omega = \{ \omega \} \) be a given set (space) of alternatives (or choices), where the set \( \Omega \) can be discrete or continuous. This set contains all the possible values for the variable \( \omega \) under consideration. The set \( \Omega \) must be a good representation of the possible alternatives, in order to lead to a significant decision, i.e., a decision that constitutes a good (close to the optimal) solution for the given problem.

A fuzzy goal \( G \) contained in the set of alternatives \( \Omega \) is identified by a given fuzzy set \( G \) contained in \( \Omega \). This fuzzy goal is characterized by its membership function \( \mu_G: \Omega \rightarrow [0, 1] \), and \( \mu_G(\omega) \in [0, 1] \) specifies the grade of membership of an alternative \( \omega \in \Omega \) for the fuzzy goal \( G \). An example of a fuzzy goal is "\( \omega \) should be substantially higher than 4", defined for the set of alternatives \( \Omega = \mathbb{R} \). This goal can be described by different fuzzy sets, as e.g. the membership function shown in Fig. 6.1a.

![Fuzzy sets](image)

(a) Fuzzy set representing the fuzzy goal "\( \omega \) should be substantially higher than 4".
(b) Fuzzy set representing the fuzzy constraint "\( \omega \) should be around 3".

**Figure 6.1. Example of a fuzzy goal (a) and a fuzzy constraint (b).**

Similarly, a fuzzy constraint \( C \) contained in the set of alternatives \( \Omega \) is described by a fuzzy set \( C \) contained in \( \Omega \), and it is characterized by its membership function \( \mu_C: \Omega \rightarrow [0, 1] \), where \( \mu_C(\omega) \in [0, 1] \) specifies the grade of membership of an alternative \( \omega \in \Omega \) for the fuzzy constraint \( C \). An example when \( \Omega = \mathbb{R} \) can be, for instance, "\( \omega \) should be around 3".
This fuzzy constraint can be represented, for instance, by the membership function depicted in Fig. 6.1b.

In the conventional decision making approach, the performance function defines an ordering of the set of alternatives. Clearly, the essence of the membership function $\mu_G(\omega)$ is the same, and a fuzzy goal can be derived from a performance function. In other words, suppose that:

$$J : \Omega \rightarrow \mathbb{R},$$

(6.1)

is a conventional performance function associating each alternative $\omega \in \Omega$ to a real number $J(\omega) \in \mathbb{R}$. This function must be bounded, i.e. $J(\omega) \leq L < \infty$, for all $\omega \in \Omega$. The limit $L$ is defined as

$$L = \max_{\omega \in \Omega} J(\omega).$$

(6.2)

In this case, the membership function of the fuzzy goal $G$ can be defined as a normalized performance function $J$ (Kacprzyk, 1997):

$$\mu_G(\omega) = \frac{J(\omega)}{L}, \quad \text{with } \omega \in \Omega,$$

(6.3)

in order to derive the fuzzy set representing the fuzzy goal in the interval $[0, 1]$. This normalization can also be performed for fuzzy constraints because they are virtually defined in the same way as fuzzy goals. Therefore, Bellman and Zadeh (1970) treated them in an analogous way. Summarizing, fuzzy goals and fuzzy constraints are defined as fuzzy sets in the set of alternatives, and thus, they can be treated alike in the formulation of a decision. Such analogy can be found also intrinsically in classical approaches such as the Lagrange multipliers or penalty functions used in optimization. However, the formulation of fuzzy decision making as in Bellman and Zadeh (1970) makes this similarity explicit.

Having a fuzzy goal $G$ and a fuzzy constraint $C$, the most straightforward way to combine them is to say that

$$G \text{ must be accomplished and } C \text{ must be satisfied.}$$

(6.4)

Therefore, the decision to be taken from the set of alternatives must simultaneously fulfill the goal and the constraint. As the goal and the constraint are fuzzy, the decision is called a fuzzy decision, and is also a fuzzy set defined in the set of alternatives $\Omega$. Thus, the fuzzy decision $D$ is a fuzzy set resulting from the aggregation of the goal and the constraint.

Formally, let $G$ be a fuzzy goal and $C$ a fuzzy constraint, both defined as fuzzy sets in the set of alternatives $\Omega$. The fuzzy decision $D$ is a fuzzy set defined also in the set $\Omega$ resulting from the aggregation $\boxplus: [0, 1] \times [0, 1] \rightarrow [0, 1]$ of the membership functions representing the fuzzy goal $\mu_G(\omega)$ and the fuzzy constraint $\mu_C(\omega)$:

$$\mu_D(\omega) = \mu_G(\omega) \boxplus \mu_C(\omega), \quad \text{with } \omega \in \Omega.$$  

(6.5)

The choice of the operator $\boxplus$ is an important issue because it influences largely the decision. If the decision making problem is formulated as in (6.4), this operation is clearly translated by
the "and" connective given by a $t$–norm (Kacprzyk, 1997). In the approach of Bellman and Zadeh (1970), the and was translated as the min operator of the two fuzzy sets representing the fuzzy goal and the fuzzy constraint. However, this operator does not allow for any interaction or trade-off between the goal and the constraint under consideration. Other operators as the $s$-norm can then be used. In many fuzzy decision making applications, the lack of compensation may not be appropriate, but, on the other hand, the union operator providing full compensation or trade-off, or any other $s$-norm, may be inappropriate as well. Operators between these two (conjunction and disjunction), such as the generalized mean (Kaymak and van Nauta Lemke, 1993), might be used as well, providing a degree of compensation among the goal and the constraint.

The definition of fuzzy decision $D$ as in (6.5), suggests that the goal and the constraint have equal importance. There are some cases, however, in which the goal has more importance than the constraint or vice-versa. A possible fuzzy decision that can deal with these type of situations is the weighted-sum type, which is defined as

$$
\mu_D(\omega) = \lambda \mu_G(\omega) + (1 - \lambda) \mu_C(\omega), \quad \text{with } \omega \in \Omega,
$$

where $\lambda \in [0, 1]$ is a weighting parameter, and $+$ stands for the algebraic sum. The parameter $\lambda$ ranges from 0, where only the constraint counts, to 1, where only the goal is considered, passing through all the intermediate values. Other possible weights for goals and constraints can be found, e.g., in Yager (1992). The use of aggregator operators using weights is avoided in this thesis, because it introduces more degrees of freedom in the FDM problem. Therefore, more parameters must be chosen by the engineer or designer, which makes the FDM problem more complex. Although, the weighting factors should still be used, when a clear hierarchy between goals and constraints is present.

**Example 6.1.1** Suppose that the fuzzy goal is given by "$\omega$ should be substantially higher than 4", as shown in Fig. 6.1a, and the fuzzy constraint is "$\omega$ should be around 3", depicted in Fig. 6.1b. Both goal and constraint are also shown in Fig. 6.2. If the min operator is chosen for the aggregation operator $\odot$, the membership function $\mu_D(\omega)$ shown in Fig. 6.2 (bold line) is obtained.

It can be seen that the set of possible alternatives is in the interval $[3, 5]$, because $\mu_D(\omega) = 0$ outside this interval. When the membership function $\mu_D(\omega)$ is not zero, it is clear that some alternatives lead to better solutions than others. Thus, the higher the value of $\mu_D(\omega)$ is, the higher is the degree of satisfaction from the choice of an alternative $\omega \in \Omega$. In this case $\mu_D(\omega) < 1, \forall \omega \in \Omega$, which means that no alternative satisfies the fuzzy goal and the fuzzy constraint completely. This is generally the case in FDM problems.

The fuzzy decision $D$ is the solution for the decision making problem. Usually, this fuzzy set has a maximum degree of membership smaller than 1. A fuzzy decision with membership functions of this type can not completely satisfy both goal and constraint. The implementation of a solution obtained from $D$ requires the defuzzification of the membership
function $\mu_D(\omega)$ in order to obtain a crisp solution. Example 6.1.1 suggests that $\mu_D(\omega)$ should be interpreted as the degree of satisfaction for a particular alternative $\omega$. This assumption can be generalized and the best alternative is thus the one that induces the highest values of $\mu_D(\omega)$.

More precisely, let $D$ be a fuzzy decision represented by its membership function $\mu_D$. Let also $\Omega^*$ be the set of points in $\Omega$ for which $\mu_D$ achieves a maximum value, if it exists. A maximizing or optimal decision is defined as one of the alternatives $\omega^* \in \Omega^*$ defined by

$$\mu_D(\omega^*) = \max \mu_D(\omega), \quad \text{or,}$$

$$\omega^* = \arg \max \mu_D(\omega). \quad (6.7)$$

For Example 6.1.1, this value is unique and given by $\omega^* = 4.1$, see Fig. 6.2. Other types of defuzzification can be used such as e.g. the center of gravity (Jager, 1995). Note, however, that the maximizing decision in (6.7) has strong similarities to the optimal solution of a classical optimization problem, presented in Section 3.1.1, Eq. (3.2).

### 6.1.2 Multiple fuzzy goals and/or fuzzy constraints

Most of the problems in fuzzy decision making are not so trivial as in the previous section where only one goal and one constraint are considered, and face a more general decision making problem, where multiple fuzzy goals and constraints are present. Fortunately, the extension to the case of multiple goals and constraints is rather straightforward, if the criteria, i.e. goals and constraints are still defined in the same space of alternatives $\Omega$. In control, this assumption is sometimes difficult to achieve, and therefore a general formulation of the decision making problem for goals and constraint in different sets is presented in Section 6.1.3.

Assume that one considers $q$ fuzzy goals $G_1, \ldots, G_q$ and $r$ fuzzy constraints $C_1, \ldots, C_r$, which are all defined as fuzzy sets in the set of alternatives $\Omega$. The resultant fuzzy decision
can be defined analogously to (6.5), for one goal and one constraint. Hence, the fuzzy
decision for this case is given by

$$
\mu_D(\omega) = \mu_{G_1}(\omega) \oplus_g \cdots \oplus_g \mu_{G_q}(\omega) \oplus_c \mu_{C_1}(\omega) \oplus_c \cdots \oplus_c \mu_{C_c}(\omega), \quad \text{with } \omega \in \Omega.
$$

(6.8)

where $\oplus$, $\oplus_g$, and $\oplus_c$ are the aggregation operators for combining a goal with a constraint, a goal with a goal and a constraint with a constraint, respectively. An even more general
definition is given by Zimmermann (1996), where a different aggregation operator is used for
each term in (6.8). Different aggregation operators can be used for different applications.
The discussion about the proper operators $\oplus$ to be chosen, presented in Section 6.1.1,
remains valid for multiple goals and constraints, as well as the possibility of using weighting
factors between different goals and constraints. A simple example of a weighted decision
function is the weighted-sum type or convex combination type of decision making (Kacprzyk,
1997), where the weighting parameters reflect the relative importance of the terms. This
fuzzy decision is the generalization of the weighted-sum aggregation presented in Eq. (6.6).
This convex combination fuzzy decision is thus defined as

$$
\mu_D(\omega) = \sum_{i=1}^{q} \alpha_i \mu_{G_i}(\omega) + \sum_{j=1}^{r} \beta_j \mu_{C_j}(\omega), \quad \text{with } \omega \in \Omega,
$$

(6.9)

with the weights $\alpha_i, \beta_j \in [0, 1]$, and these parameters are combined such that $\sum_{i=1}^{q} \alpha_i
+ \sum_{j=1}^{r} \beta_j = 1$. Thus, the values of $\alpha_i$ and $\beta_j$ reflect the importance of the different goals $G_i$
and constraints $C_j$. Note that Eq. (6.9) is similar to the combination of different vector criteria
in a scalar function using the weighted-sum objective, which is presented in Section 3.1.1,
Eq. (3.4).

The maximizing decision for multiple fuzzy goals and constraints is also obtained by
maximizing the membership function obtained for the decision making $\mu_D(\omega)$ over $\omega \in \Omega$,
as in (6.7). The fuzzy decision for multiple criteria is defined in such a way that it wants to
attain all the fuzzy goals and satisfy all the fuzzy constraints simultaneously. This condition
can be difficult to obtain, and the most common approach is to relax this requirement, by
attaining most of the goals and satisfying almost all constraints (Kacprzyk, 1997). In this
thesis a different approach is proposed to deal with this problem, when control applications
of fuzzy decision making are considered. This approach attains to fulfill all the goals
and all the constraints to a certain degree, if some pre-requirements are satisfied. One of
this requirements is the formulation of the FDM problem in different spaces, presented in
the next section, which allows for different and more general definitions of fuzzy goals
and constraints. Another difference proposed in this thesis is the definition of goals and
constraints in such a way that some special requirements, sometimes present in predictive
control, as, e.g., the presence of 'hard' constraints, can be fulfilled. A complete discussion
of this problem is presented in Section 6.2.
6.1.3 Fuzzy goals and fuzzy constraints defined in different spaces

The discussion presented until now concerns only fuzzy goals and fuzzy constraints defined in the same space of alternatives \( \Omega \). However, for most applications of decision making, and especially the application of decision making in control, an extension to fuzzy goals and fuzzy constraints defined in different sets is needed.

Bellman and Zadeh’s (1970) approach. Bellman and Zadeh (1970) presented a formulation where a mapping from the set (space) of alternatives \( \Omega = \{ \omega \} \) to a set (space) of effects of these alternatives \( \Phi = \{ \phi \} \) is considered. In this formulation \( \omega \) represents an input and \( \phi = f(\omega) \) represents the corresponding output.

More formally, let the \( r \) fuzzy constraints \( C_1, \ldots, C_r \) be defined as fuzzy sets in \( \Omega = \{ \omega \} \), and let the \( q \) fuzzy goals \( G_1, \ldots, G_q \) be defined as fuzzy sets in \( \Phi = \{ \phi \} \). Suppose that a given mapping \( f: \Omega \to \Phi \) exists and is known. Typically, \( \Omega \) is the input set (cause), and \( \Phi \) is the output set (effect). Given a fuzzy goal \( G_i \) in \( \Phi \), an induced fuzzy goal \( G'_i \) in \( \Omega \) is derived from the fuzzy goal \( G_i \) in \( \Phi \) by

\[
\mu_{G'_i}(\omega) = \mu_{G_i}(f(\omega)), \quad i = 1, \ldots, q. \tag{6.10}
\]

The fuzzy decision can be expressed as the confluence of goals and constraints, similarly to the aggregations performed in (6.8). Using (6.10), \( \mu_D(\omega) \) is explicitly expressed by

\[
\mu_D(\omega) = \mu_{C_1}(\omega) \circ_2 \cdots \circ_2 \mu_{G'_q}(\omega) \circ_1 \mu_{G_1}(\omega) \circ_1 \cdots \circ_1 \mu_{G_r}(\omega). \tag{6.11}
\]

The introduction of the induced fuzzy goal allows for the definition of fuzzy goals \( G'_i \) and fuzzy constraints \( C_j, j = 1, \ldots, r \) in the same set \( \Omega \). Hence, the aggregation operators \( \circ, \circ_2, \circ_1 \) can still be utilized for the aggregation when this transformation is used. This approach can also be quite useful in the analysis of multistage decision making (Bellman and Zadeh, 1970; Kacprzyk, 1997), which is presented in Section 6.2.

The presented approach implies that the mapping between \( \Omega \) and \( \Phi \) must be known, and the definition of goals and constraints is restricted to \( \Phi \) and \( \Omega \), respectively. However, it is sometimes difficult to restrict the definition of goals and constraints to these sets. And, in a more general approach, each goal, as well as each constraint, can be defined in different sets. The performance criteria of a given system under control, for instance, can be defined using different variables, as, e.g., the error between the reference and the system’s output, the system’s output itself, the overshoot, etc. Moreover, in general these goals can be defined at a higher level, where the definition of the space is more vague.

Goals and constraints defined in different spaces. Due to the lack of generalization in Eq. (6.11), a different approach is suggested in this thesis, where each goal and each constraint can be defined in a different universe of discourse. Note that this approach has
still the advantages of using fuzzy sets and their intrinsic normalization. The optimization
to find the optimal decision is made in a \( T \)-dimensional space, where \( T \) stands for the sum
of all goals and all constraints.

Formally, let \( T = q + r \) be the total number of fuzzy criteria (sum of fuzzy goals and fuzzy
constraints). A criterion \( \zeta_i, i = 1, \ldots, T \), can correspond to a fuzzy goal \( G_j, j = 1, \ldots, q \),
or a fuzzy constraint \( C_k, k = 1, \ldots, r \). Let also \( \Phi_i, i = 1, \ldots, T \) be a space defined for
a given criterion. Note that goals and constraints can be treated alike, and a space \( \Phi_i \) can
be correspond to the spaces previously defined \( \Omega, \Phi \), or any other space. Moreover, for a
given problem different goals and different constraints can still be defined in the same space.
The fuzzy criteria \( \zeta_i, i = 1, \ldots, T \) must now be combined. As the criteria are defined in
different spaces, the confluence of goals and constraints can not be done by using the fuzzy
operators \( \odot, \oplus, \ominus \) directly, because these operators can only aggregate fuzzy sets
that are defined in the same space, e.g. the space of alternatives \( \Omega \) defined in Section 6.1.1.
The confluence of criteria must be done in such a way that each criterion can be defined in
a different space. Therefore instead of fuzzy operators, a fuzzy relation must be used. The
general definition of a fuzzy relation and the cylindrical extension used in the sequel are given in Appendix A.

Let \( \mathcal{R}_D \) be the fuzzy relation of all the criteria. This relation is defined in the Cartesian
product of all the spaces defined for each criteria \( \zeta_i \), resulting in the space \( \Phi = \Phi_1 \times \Phi_2 \times
\ldots \times \Phi_T \).

\[
\mathcal{R}_D : \Phi_1 \times \Phi_2 \times \ldots \times \Phi_n \rightarrow [0, 1].
\] (6.12)

In order to define the relation \( \mathcal{R}_D \), let each fuzzy criterion \( \zeta_i \) be cylindrical extended to the
space \( \Phi \):

\[
\text{C}_\zeta_i ((\phi_1, \ldots, \phi_T)) = \text{cext}_i^{\Phi} (\mu_{\zeta_i}),
\] (6.13)

where \([\phi_1, \ldots, \phi_T]\) is a point in the \( T \)-dimensional space \( \Phi \). Note that each of this cylindrical
extensions is a fuzzy set in the multidimensional space \( \Phi \). As these fuzzy sets are in the
same multidimensional space, they can be combined by using fuzzy aggregations deriving
the relation \( \mathcal{R}_D \) between the criteria, which is described by

\[
\mu_{\mathcal{R}_D} (\phi_1, \ldots, \phi_T) = \text{C}_\zeta_1 \odot_g \ldots \odot_g \text{C}_\zeta_\xi \odot \text{C}_{\pi_{T+1}} \odot_c \ldots \odot_c \text{C}_{\pi_T}.
\] (6.14)

Note that the aggregation operators \( \odot, \odot_g \) and \( \odot_c \) in (6.14) are generalizations of the
operators in (6.11), which are unidimensional operators \([0, 1] \times [0, 1] \rightarrow [0, 1] \), to \( T \-
dimensional operators \([0, 1]^T \times [0, 1]^T \rightarrow [0, 1]^T \).

As the goals and the constraints are defined in different spaces, it is necessary to obtain the
values \( \phi_i \in \Phi_i \), given the corresponding alternative \( \omega \). In more formal terms, this means
that for each criterion \( \zeta_i \) defined in \( \Phi_i \), a mapping \( f_i \) from the space of alternatives \( \Omega \) to the
particular space \( \Phi_i \) must exist:

\[
f_i : \Omega \rightarrow \Phi_i.
\] (6.15)

Thus, the total mapping \( f \) from the space of alternatives to the \( T \)-dimensional space of all
criteria \( \Phi = \Phi_1 \times \Phi_2 \times \ldots \times \Phi_T \) is defined as

\[
f : \Omega \rightarrow \Phi.
\] (6.16)
This mapping usually restricts the space of the decision resulting in a new decision space \( \Phi_r \). The fuzzy relation \( R_D \) must also be restricted to this space. The obtained relation \( R_{Dr} \) is now defined by

\[
R_{Dr} : \Phi_r \rightarrow [0, 1].
\]

(6.17)

Note that this relation can still be derived using (6.14), if the possible points representing several criteria are inside the new space, i.e., \([\phi_1, \ldots, \phi_T] \in \Phi_r\). Note also that the \( T \)-dimensional aggregation operators \( \oplus \), \( \odot_g \), and \( \odot_c \) must be chosen in such a way that the fuzzy relation \( \mu_{R_{Dr}}(\{\phi_1, \ldots, \phi_T\}) \) is not empty for all points in the space \( \Phi_r \). The optimal decision \([\phi_1, \ldots, \phi_T]^*\) is given by the point in \( \Phi_r \) for which the relation \( R_{Dr} \) attains its maximum:

\[
\mu_{R_{Dr}}([\phi_1, \ldots, \phi_T]^*) = \max_{(\phi_1, \ldots, \phi_T) \in \Phi_r} R_{Dr}
\]

\[
[\phi_1, \ldots, \phi_T]^* = \arg \max_{[\phi_1, \ldots, \phi_T] \in \Phi_r} R_{Dr}
\]

(6.18)

This point is derived from a certain optimum alternative \( \omega^* \). A simple example of a two-dimensional space for the criteria is given in Example 6.1.2.

**Example 6.1.2** Let a fuzzy constraint \( C \supseteq \zeta_1 \), corresponding to the criterion \( \zeta_1 \), be defined by a triangular membership function in the space \( \Phi_1 \), as in Fig. 6.3. Let also a fuzzy goal \( G \supseteq \zeta_2 \) be defined by a trapezoidal membership function in the space \( \Phi_2 \), see also Fig. 6.3. Further, this figure shows the relation \( \mu_{R_D} \) in the space \( \Phi_1 \times \Phi_2 \) when the internal product

\[\text{Figure 6.3. Space obtained by the combination of a fuzzy constraint in } \Phi_1 \text{ and a fuzzy goal in } \Phi_2.\]

(see Appendix A) of fuzzy sets \( \zeta_1 \) and \( \zeta_2 \) is used as aggregation operator, i.e. the cylindrical
6.1 Fuzzy decision making

Figure 6.4. Transformation of the space of the alternatives in $\Phi_f$ and optimal decision $[\phi_1, \phi_2]^*$ obtained from the application of $f$ to goals and constraints.

extensions of $\zeta_1$ and $\zeta_2$, $C_{E_1}$, and $C_{E_2}$ respectively, are combined using the min operator. The mappings from $\Omega$ to $\Phi_1$ and $\Phi_2$ are given respectively by:

$$ f_1: \Omega \rightarrow \Phi_1, \quad f_2: \Omega \rightarrow \Phi_2. $$

Thus, an alternative $\omega$ in $\Omega$ is mapped to a point $[\phi_1, \phi_2]$ in the space $\Phi_1 \times \Phi_2$. This point is transformed afterwards to a certain degree of fulfillment by using the relation $\mathcal{R}_{D_F}$. Figure 6.3 presents the example of one point $[\phi_1, \phi_2]$ obtained from the transformation of a certain $\omega$ by using $f_1$ and $f_2$. Thus, $\phi_1 = f_1(\omega)$ and $\phi_2 = f_2(\omega)$. The membership degree corresponding to that point: $\mu_{\mathcal{R}_{D_F}}(\phi_1, \phi_2)$ is also shown in this figure.

The mapping $f(\omega)$ in $\Phi_1 \times \Phi_2$ for all the possible values of $\omega$ is depicted in Fig. 6.4. The space $\Phi_1 \times \Phi_2$ is transformed in the curve $\Phi_f$ by using the functions $f_1$ and $f_2$. Thus, each point of $\Phi_f$ is obtained by using a given alternative $\omega$. This line is transformed by the relation $\mathcal{R}_{D_F}$ resulting in the curve $\mu_{\mathcal{R}_{D_F}}$ also depicted in Fig. 6.4. The optimal decision is given by the point with the maximum membership degree, as in (6.18), and it is given by the point $[\phi_1, \phi_2]^*$ in $\Phi_f$, for which the relation $\mathcal{R}_{D_F}$ attains its maximum, see Fig. 6.4. Note that even for this simple two-dimensional case it is clear that the optimization problem can be a difficult non-convex optimization problem. $\square$

**Multidimensional space of alternatives.** Until now no detailed discussion over the space of alternatives $\Omega$ was made. It was implicitly assumed that the space of alternatives is unidimensional. However, this space can be multidimensional, as e.g. for MISO or MIMO control systems, where the states of the system are usually in different spaces. For the
multidimensional case, \( \Omega \) is a composition of several unidimensional sets \( \Omega_i, i = 1, \ldots, S \), and the space of alternatives is given by

\[
\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_S.
\] (6.19)

**Solution of multidimensional FDM using discrete alternatives.** The definition of criteria in different spaces allows for a generalization of fuzzy decision making defined by Bellman and Zadeh (1970). Although more general, this approach has a serious drawback. In fact, working with relations in the \( T \)-dimensional space is readily more difficult than working with operators between fuzzy sets in the same unidimensional space. Moreover, if the number of goals and constraints is rather large, the computational time to derive the global relation is usually also large. Further, the process of finding the optimal is rather difficult because there is no guarantee of convex surfaces, except for special cases as in Section 7.3. These problems can be circumvented if the set of alternatives is discrete instead of continuous.

Let \( \omega_i \) be a discrete alternative. If the space of alternatives is \( S \)-dimensional, as in (6.19), an alternative is thus represented by the vector

\[
\omega_i = [\omega_{i1} \ \omega_{i2} \ \ldots \ \omega_{iS}]^T.
\] (6.20)

Let the total number of discrete alternatives be given by \( M \). With these alternatives it is possible to construct a matrix \( D \) representing the membership degrees of the discrete alternatives \( \omega_i \) for all the criteria \( \zeta_j \):

\[
D = \begin{pmatrix}
\zeta_1 & \zeta_2 & \ldots & \zeta_T \\
\omega_1 & \begin{pmatrix} \mu_{\zeta_1}^1 & \mu_{\zeta_2}^1 & \ldots & \mu_{\zeta_T}^1 \end{pmatrix} \\
\omega_2 & \begin{pmatrix} \mu_{\zeta_1}^2 & \mu_{\zeta_2}^2 & \ldots & \mu_{\zeta_T}^2 \end{pmatrix} \\
\vdots & \vdots & \ddots & \vdots \\
\omega_M & \begin{pmatrix} \mu_{\zeta_1}^M & \mu_{\zeta_2}^M & \ldots & \mu_{\zeta_T}^M \end{pmatrix}
\end{pmatrix},
\] (6.21)

where \( \mu_{\zeta_i}^j, i = 1, \ldots, M, j = 1, \ldots, T \) is the membership degree for the alternative \( \omega_i \) and the criterion (goal or constraint) \( \zeta_j \). As the membership degrees \( \mu_{\zeta_i}^j \) are normalized values in the interval \([0, 1]\), it is possible to aggregate them using fuzzy operators, as before in (6.8). For each alternative \( \omega_i \), the corresponding membership degree is given by:

\[
\mu_D(\omega_i) = \mu_{\zeta_1}^i \odot_y \ldots \odot_y \mu_{\zeta_4}^i \odot_e \mu_{\zeta_5}^{i+1} \odot_e \ldots \odot_e \mu_{\zeta_T}^i.
\] (6.22)

The alternative with the maximal decision value is chosen as the optimal decision, as in (6.7), following the usual approach in fuzzy decision making. Note that this method can be directly used for discrete problems. Two examples of the application of discrete alternatives in a control problem are given in Section 6.3.

If the FDM problem has a continuous space of alternatives, it is necessary to discretize this space first, and obtain the discretized solution afterwards. The optimal solution is obtained
using a discrete optimization method. Chapter 7 presents some optimization methods for discretized spaces. The efficiency of this optimization is strongly related to the number of points chosen for the discretization. If this is too coarse, the solution of the optimization problem can be quite poor. On the other hand, if the number of discretizations is large, the computation time is also large, and a readily good solution can not be found in a reasonable amount of time. Some clues on how to choose the proper number of discretizations for fuzzy decision making problems in control are also presented in Chapter 7.

The core of fuzzy decision making presented in this section can be applied to problems where time is not involved. However, many problems have dynamic characteristics, where multistage decision making must be considered. In the scope of this thesis, FDM must be treated in the control environment, which is usually a dynamical problem, especially if a multistep predictive control scheme is considered. The problem of multistage decision making is considered in the next section.

6.2 Fuzzy decision making in 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 also multistage decision making can be applied to other fields. In this thesis, multistage decision making is presented as FDM in control, as done before by several authors (Bellman and Zadeh, 1970; Kacprzyk, 1997). Note that the notation used in this section is convenient for control approaches, but multistage FDM maintains its generality. When multistage decision making is translated to the control environment, the set of alternatives constitute the different control actions, the system under control is a relationship between inputs and outputs (or causes and effects), and the mapping relating the inputs to the outputs of the system under control in referred to as the model. Moreover, fuzzy constraints are constraints defined for several variables presented 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, as defined in Chapter 3.

The systems and models considered in this thesis for FDM applications follow the definitions presented in Section 2.1, Eq. (2.2). However, in the general case the systems can be time-variant, and not only time-invariant as in (2.2). Moreover, the system considered in (2.2) is a deterministic one, while in general the system can be of other types. The most utilized types of systems are the following:

- deterministic,
- stochastic,
- fuzzy.
Another 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 Section 4.1. In this thesis 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 resume 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** - The solution of this type of termination time for deterministic systems using dynamic programming is given in (Bellman and Zadeh, 1970). Different techniques were proposed to solve this problem, as e.g. branch-and-bound (Kacprzyk, 1978a; Kacprzyk, 1979; Esogbue, et al., 1992), a genetic algorithm (Kacprzyk, 1995a; Kacprzyk, 1995c), 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 (1980a), 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 (Kacprzyk, 1979), interpolative reasoning (Kacprzyk, 1993b; Kacprzyk, 1993a) and a genetic algorithm (Kacprzyk, 1995a) were proposed.

- **implicitly specified termination time** - In these systems the process terminates when the outputs reach some prespecified value. An iterative solution for deterministic systems was introduced by Bellman and Zadeh (1970). A graph-theoretic analysis was also used to tackle the same problem, but a simpler solution can be derived by using the branch-and-bound approach (Kacprzyk, 1978a).

- **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 (1977b). For deterministic and stochastic systems, solutions using dynamic programming (Kacprzyk, 1977; Kacprzyk, 1978b; Stein, 1980) or branch-and-bound (Kacprzyk, 1983) are possible. An extension of these methods 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. Fundamental references are the works of (Howard, 1960; Howard, 1971), which introduce a policy iteration technique, and solve infinite termination time problems using a finite sequence of iterations. A solution for deterministic, stochastic and fuzzy systems was introduced in Kacprzyk and Staniewsky (1980b) and extended 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.
This thesis restricts the termination time to the prediction horizon, which is shifted when time evolves, but on the other hand allows for the application of multistage FDM for low level of control in real-time. Note that even recently 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 (...)."

His book is dedicated to multistage decision making (control) in a fuzzy environment considering time-variant, stochastic and/or fuzzy systems, but just for open-loop control.

The application of fuzzy decision making to close-loop control was first developed by Yasunobu and Miyamoto (1985). The fuzzy control criteria including safety, comfort, energy consumption and stopping accuracy is included in a linguistic fuzzy model derived from expert knowledge and implemented in a predictive control scheme. This control system presents the disadvantages detected in using linguistic models derived from expert rules, i.e., it requires the trial-and-error tuning of the rules. However, the predictive fuzzy controller is being used on the Sendai city subway in Japan, presenting better performance than the previously used controllers. Jia and Zhang (1991a) presented a different approach for MIMO systems, but their approach is quite complex and computationally demanding, hampering its application in real-time. The papers developing this approach presented by the same authors confirms these disadvantages (Jia and Zhang, 1993c; Jia and Zhang, 1993b), as well as the work of Chang, et al. (1996). To our best knowledge, and excluding the references where the author of this thesis is included, no other applications have been reported. Moreover, the application presented in Chapter 8 is maybe the first real-time application of FDM applied to low level control.

This thesis considers only deterministic systems. In fact, most of the physical system are still modeled using deterministic, time-invariant reasoning. Moreover, nonlinear systems are modeled by using nonlinear modeling techniques. In this thesis fuzzy modeling is used to derive nonlinear models, see Section 2.2. However, other modeling techniques, such as standard nonlinear regression (Seber and Wild, 1989), neural networks (Hunt, et al., 1992), etc. 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 in the future. 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, et al. (1997). 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. Moreover, 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 will possibly be able to model uncertain processes, for which no present-day modeling paradigm can be used, allowing the generalization of FDM in control as presented in this thesis, for fuzzy systems. The definition of fuzzy goals and constraints
in the control environment is presented in Section 6.2.1. The aggregation of the different criteria for control applications is presented in Section 6.2.2, where the set of alternatives is discretized in order to find the optimal control actions. The application of FDM to predictive control becomes then easy to understand, and is presented in Section 6.2.3.

6.2.1 Fuzzy goals and fuzzy constraints in the control environment

The fuzzy goals $G_i$ and the fuzzy constraints $C_i$ can be defined for any of the spaces $U$, $V$ or $X$, or even in any other convenient space. Note that usually fuzzy constraints are defined in $U$, and fuzzy goals in $X$ (Bellman and Zadeh, 1970; Kacprzyk, 1997). The approach was applied initially to systems with discrete states and a finite number of possible transitions between the states, and subsequently it has been extended to systems with continuous states (Gluss, 1973). The reasoning applied in this thesis for FDM in control, and presented in the sequel, 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 Section 6.1.3 for fuzzy goals and constraints in different spaces.

As before, 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 $G_i$ 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 $E$. An example is the fuzzy goal "small output error", defined for a SISO system and shown in Fig. 6.5a. Fuzzy constraints can be defined on the universe of discourse $U$ of the control variables $u$. An example, also for 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. 6.5b. Note that the two 'hard' constraints $0 \leq u \leq 1$ are also included in the given membership function. The given examples of a goal and a constraint show that it is sometimes advantageous to treat them in different ways, contrary to the Bellman and Zadeh (1970) approach.

The qualitative distinction between goals and constraints can be settled when the membership functions for a fuzzy criterion are defined. In this thesis, a fuzzy goal is defined in such a way that the membership grade is never zero, unless this is strictly necessary. Therefore, the example in Fig. 6.5a 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. 6.5b 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. 6.5b is a valve opening, where 1 stands for completely open and 0 for completely closed. Therefore, the definition of the fuzzy constraint as is given in Fig. 6.5b, takes this physical limitations into account. It is suggested that a fuzzy criterion should be defined with 'hard' constraints when these are present in the system, and in this case it is seen as a fuzzy constraint. A fuzzy goal should be defined so that the membership function can be very low, but never
becomes zero. This procedure distinguishes goals and constraints in the form of the defined membership functions, but clearly does not affect the confluence of criteria.

\begin{figure}[h]
\centering
\subfloat[Goal: 'small output error'.]{
\includegraphics[width=0.4\textwidth]{goal.png}
\label{fig:goal}
}
\hfill
\subfloat[Constraint: 'u not substantially larger than 0.8'.]{
\includegraphics[width=0.4\textwidth]{constraint.png}
\label{fig:constraint}
}
\caption{Example of a fuzzy goal and a fuzzy constraint for FDM in control.}
\end{figure}

Assume that one considers \( q \) goals \( G_1, \ldots, G_q \) and \( r \) constraints \( C_1, \ldots, C_r \). Each fuzzy goal \( G_i \) and each fuzzy constraint \( C_i \) constitute a decision criterion \( \zeta_{ij}, i = 1, \ldots, T \), where \( T' = q + r \) is the total number of goals and constraints. Each criterion is defined in the space \( \Phi_i, i = 1, \ldots, T \), which can be any of the various spaces used in control: \( U, Y, X \) or \( E \). The confluence of goals and constraints could be obtained by generalizing the reasoning presented for goals and constraints in different spaces, presented in Section 6.1.3 from Eq. (6.12) to (6.18). It is simpler, however, to describe the problem directly for discrete alternatives, which are the only ones used in this thesis, to solve control problems using fuzzy objective functions.

### 6.2.2 Aggregation of criteria in the control environment

Assume that a policy \( \pi \) is defined as a sequence of control actions for the entire prediction horizon

\[
\pi = u(k), \ldots, u(k + H_p - 1) \in \Omega,
\]

(6.23)

where the control actions belong to the set of alternatives \( \Omega \). In the general case, all the criteria must be applied at each time step \( i \), with \( i = 1, \ldots, H_p \). Thus a criterion \( \zeta_{ij} \) denotes that the criterion \( j \) is considered at time step \( k + i \), with \( i = 1, \ldots, H_p \) and \( j = 1, \ldots, T \). Further, let \( \mu_{\zeta_{ij}} \) denote the membership value that represents the satisfaction of the decision criteria after applying the control actions \( u(k + i) \). The total number of decision criteria in the problem is thus given by \( \hat{T} = T \cdot H_p \). The confluence of goals and constraints can be
done by aggregating the membership values $\mu_{\zeta_i}$. The membership value $\mu_n$ for the control sequence $\pi$ is obtained using the aggregation operators $\boxdot$, $\boxtimes$ and $\boxplus$ to combine the decision criteria:

$$
\mu_n = \mu_{\zeta_{11}} \boxtimes \ldots \boxtimes \mu_{\zeta_{1s}} \boxtimes \mu_{\zeta_{1(s+1)}} \boxtimes \ldots \boxtimes \mu_{\zeta_{1T}} \boxplus \mu_{\zeta_{21}} \boxtimes \ldots \boxtimes \mu_{\zeta_{2s}} \boxtimes \mu_{\zeta_{2(s+1)}} \boxtimes \ldots \boxtimes \mu_{\zeta_{2T}} \boxplus \ldots \mu_{\zeta_{Hp,1}} \boxtimes \ldots \boxtimes \mu_{\zeta_{Hp,s}} \boxtimes \mu_{\zeta_{Hp,(s+1)}} \boxtimes \ldots \boxtimes \mu_{\zeta_{Hp,T}} .
$$

(6.24)

Note that the aggregation operator to combine a goal and a constraint at different time steps, i.e., $\mu_{\zeta_i}$ to $\mu_{\zeta_i(t+1)}$, $i = 1, \ldots, H_p - 1$, is the same as the one to combine a goal and a constraint at the same time step, i.e., $\mu_{\zeta_i}$ to $\mu_{\zeta_i(t+1)}$, $i = 1, \ldots, H_p$. For the purpose of this thesis the aggregation as in (6.24) 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 $\boxplus_{\ell}$, $\ell = 1, \ldots, T - 1$ can be used for each aggregation. This possibility is clearly not applicable, 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 using the well-known properties of these operators (Klir and Yuan, 1995). The aggregation operators can be used to 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 detailed discussion over the influence of aggregation operators in FDM applied to control is given in Section 6.4.

The translation of each goal and constraint given a certain policy $\pi$ to a membership value as in (6.24) avoids the combination of the criteria in a large dimensional space, as presented in Section 6.1.3. Note that the problem would be $T$-dimensional in this case. Instead, the combination of criteria in different spaces by discretizing the sets of alternatives and resulting in different policies $\pi$, can be applied to find the optimal control policy. Using this approach, the computation of the optimal policy $\pi^*$ becomes simpler. The decision criteria in (6.24) should be satisfied as much as possible, which corresponds to the maximal value of the overall decision. Thus, the optimal sequence of control actions $\pi^*$ is found by the maximization of $\mu_n$:

$$
\pi^* = \arg \max_{u(k), \ldots, u(k+H_p-1)} \mu_x .
$$

(6.25)

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 (6.25) is usually non-convex. Problems concerning optimization are discussed in Chapter 7. Sometimes, only some criteria are considered in (6.24). The criteria which are not considered can be simply removed from this equation. An example is to consider the minimization of the error of the $i$-th output of the system $y_i(k)$ only close to the prediction horizon $H_p$. Hence, every criterion can be simply removed beforehand from (6.24). This simplification alleviates the computation of the optimal control sequence, and sometimes it leads even to better results.
6.2.3 Fuzzy criteria in model-based predictive control

The definition of fuzzy goals and constraints must be given by an operator or design engineer. Therefore, human knowledge is involved in specifying the control objectives and constraints, rather than the control protocol itself, when FDM in control is considered (Meiritz, et al., 1995). Using a process model, a fuzzy decision making algorithm selects the control actions that best meet the specifications, see Fig. 6.6. Hence, a control strategy can be obtained that is able to push the process operation closer to the constraints and to force the process to a better performance based on goals and constraints set by the operator and by known conditions provided by the system’s designers.

![Controller diagram](image)

*Figure 6.6. Controller based on objective evaluation and fuzzy decision making.*

This prescriptive approach is closely related to model-based predictive control, presented in Section 4.1. 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 model-based predictive control (Sousa, et al., 1996b; Sousa, et al., 1996c; Babuška, et al., 1995b). 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 among the 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 $\pi$ with the possible control actions $u(k), \ldots, u(k + H_p - 1)$ can be defined as in Eq. (6.23). The objective function using fuzzy criteria is defined in (6.24). The closed-loop control configuration presented in this thesis is now discussed in more detail, in aspects concerning the criteria and the aggregation operator used to combine the criteria. Thus, the first problem in MBPC with a fuzzy objective function is to define the several performance criteria. Section 6.3 presents a discussion about the possible criteria to be used in fuzzy multicriteria decision making applied to control. Using fuzzy goals and fuzzy constraints it is possible to aggregate them using fuzzy operators, choosing the operator that best fits the desired combination of goals and constraints. Section 6.4 presents a discussion on the possible aggregation operators for FDM in control.
6.3 Fuzzy criteria for FDM 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 translate 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-based predictive control framework (te Braake, 1997). This section investigates the use of fuzzy performance criteria in predictive control and compares the results to those obtained from conventional model-based predictive control. This study is the development of previous work carried out by the author together with others (Kaymak, et al., 1997b). First, control criteria and decision functions are discussed in Section 6.3.1. The predictive controllers are applied to two simulated systems: (i) a nonminimum phase, unstable linear plant and (ii) an air-conditioning system with nonlinear dynamics. The description of the simulated systems is presented in Section 6.3.2. Section 6.3.3 presents a discussion over the obtained results. Another applications of the presented approach can be found in Sousa, et al. (1996b) and Kaymak and Sousa (1997).

6.3.1 Control criteria and decision functions

When a control system is designed, performance criteria to be accomplished are specified. In the time domain, these criteria are usually defined in terms of the desired steady-state errors between the references and the outputs, rise time, overshoot, settling time, etc. – see Section 3.3 – presenting the goals of the control system. In MBPC, these goals representing the performance criteria must be translated and represented in an objective function. This function is optimized over the prediction horizon, given the desired control actions $u(k)$. The translation of the (fuzzy) goals to an objective function can be done in two different ways.

- The control goals are explicitly expressed in the objective function. This method leads usually to long term predictions of the behaviour 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.

- 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. For this case, the overall control goals must 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, the 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 it will be shown in this section.

Classical objective functions. Conventional MBPC, mainly utilizes sum-quadratic func-
tions like in (4.2) as the objective function. 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 nonconv-
ex constraints and/or the presence of nonlinearities in the system often lead to nonconvex
optimization problems. In these cases, the sum-quadratic objective function does not have
any advantages over other more complex objective functions, which can describe possibly
better 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 can be represented in the
objective function as in (4.2). An extension of this objective function is used in this section
by including the changes in the outputs in the objective function, resulting in the following:

\[ J = \sum_{i=m_1}^{n_1} \alpha_i (\hat{e}(k+i))^2 + \sum_{i=m_2}^{n_2} \beta_i (\Delta u(k+i-1))^2 + \sum_{i=m_3}^{n_3} \gamma_i (\Delta \hat{y}(k+i))^2, \]

(6.26)

where \( \hat{e}(k+i) \) denotes the predicted errors given by

\[ \hat{e}(k+i) = r(k+i) - \hat{y}(k+i). \]

(6.27)

The change in the predicted outputs \( \Delta \hat{y} \) is defined as

\[ \Delta \hat{y}(k+i) = \hat{y}(k+i) - \hat{y}(k+i-1), \]

(6.28)

and is equal to the change in the errors \( \Delta \hat{e}(k+i) \), when the references to be followed are constant. The change in the control actions is defined in a similar way:

\[ \Delta u(k+i) = u(k+i) - u(k+i-1). \]

(6.29)

The parameter vectors \( \alpha_i, \beta_i \) and \( \gamma_i \), which are \( p \), \( m \) and \( p \)-dimensional respectively, are
also weighting terms that are application dependent. The parameters \( m_1, n_1, m_2, n_2, m_3 \)
and \( n_3 \) must be selected appropriately depending on the application, and they must satisfy
\( 1 \leq m_i \leq n_i \leq H_p, \ i \in \{1, 2, 3\} \). Usually, \( m_1, m_2, \) and \( m_3 \) are chosen equal to 1, \( n_1 \) and \( n_3 \)
equal to \( H_p, \) and \( n_2 \) equal to \( H_c. \) Note that the weighting terms \( \alpha_i, \beta_i \) and \( \gamma_i \) must reflect 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 weights different variables, which is not desirable, and it leads
to poor control performance.
The objective function (6.26) 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 outputs 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 parameter vectors containing the weights can be changed, which modifies the objective function in order to lead the system to a desired response. The vectors $\alpha_i$, $\beta_i$ and $\gamma_i$ have thus two functions: normalize the different outputs and inputs of the system and vary the weights of the three different terms in the objective function (6.26) over the time steps.

**Fuzzy objective functions.** When fuzzy multicriteria decision making is applied to determine the objective function, additional flexibility is introduced. Each criterion $\zeta_{ij}$ is described by a fuzzy set, where $i = 1, \ldots, H_p$, stands for the time step $k + i$, and $j = 1, \ldots, T$ 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. For the sake of simplicity and without loss of generality a SISO system is considered, with a control action $u(k)$ and an output $y(k)$. Fig. 6.7 shows examples of general membership functions that can be used for the error $\hat{e}(k + i) = r(k + i) - \hat{y}(k + i)$, change in the predicted output $\Delta \hat{y}(k + i)$ and change in the control action $\Delta u(k + i - 1)$, with $i = 1, \ldots, H_p$.

![Figure 6.7](image)

**Figure 6.7.** Membership functions that represent the satisfaction of decision criteria for the error, change in output and change in the control action.

In this example, the minimization of the output error $\mu_e(\hat{e}(k + i))$ is represented by an exponential membership function, given by

$$
\mu_e = \begin{cases} 
\exp\left(-\frac{|\hat{e}(k + i)|}{K_e}\right), & -\infty < \hat{e}(k + i) < 0; \\
\exp\left(-\frac{|\hat{e}(k + i)|}{K_e}\right), & 0 \leq \hat{e}(k + i) < \infty.
\end{cases} \quad (6.30)
$$
This well-known function has the nice property of being tangent to the triangular membership function defined using the parameters \( K_c^- \) and \( K_c^+ \), see Fig. 6.7. 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 \( K_c^- \) or \( K_c^+ \) magnitude. Therefore, this criterion is considered a fuzzy goal, as explained in Section 6.2.1. This definition of membership function allows for the comparison of the error parameters, \( K_c^- \) and \( K_c^+ \), to the parameters defined for other fuzzy criteria, as e.g., the change in output and change in control, which are defined in the following.

The control effort \( \mu_u(\Delta u(k+i - 1)) \) is, in this case, represented by a triangular membership function around zero, which is considered a fuzzy constraint. The crisp rate constraints on \( \Delta u \) 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 \( \mu_u(\Delta u(k+i - 1)) \) 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. 6.7. Moreover, \( \mu_u \) can be defined as a trapezoidal membership function in a similar way to the one defined for the change in the output.

The change in the output can be represented, e.g. by a trapezoidal membership function \( \mu_y(\Delta \hat{y}(k+i)) \), as shown in Fig. 6.7. 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 can not go below \( K_y^- \) or above \( K_y^+ \). Using these four points, a trapezoidal membership function is defined, which has membership degree 1 in the interval \([S_y^-, S_y^+]\) and has membership degree 0, in the intervals \((-\infty, K_y^-)\) and \([K_y^+, \infty)\), see Fig. 6.7. 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 \( K_c^-, \) and \( K_c^+ \) in the membership function defined for error, as in (6.30). Thus, outside the interval \([S_y^-, S_y^+]\) exponential membership functions as the one defined for the error \( \hat{e}(k+i) \) can be used.

In principle, different criteria can be defined at each time instant \( k+i \), \( i = 1, \ldots, H_p \). The given example has \( T = 3 \) decision criteria, 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. 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 choose carefully 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.

The membership functions $\mu_{\xi_{ij}}$ quantify how much the system satisfies the criteria given a particular control sequence, bringing various quantities into a 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 explicitly specify 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 the several parameters $\alpha_i, \beta_i$ and $\gamma_i$, as in Eq. (6.26), when fuzzy objective functions are used, because the use of membership functions introduce directly the normalization required. For this particular aspect, this feature reduces the effort on designing MBPC with fuzzy objective functions, when compared to classical objective functions.

After the membership functions have been defined, they can be combined by using various decision functions, such as the parametric aggregation operators from the fuzzy set theory. Usually, the additional parameter of the decision function influences the optimization results in a way that cannot be expressed by the weight factors. In this way, the objective function can be tuned with a single parameter which results in improved control performance, see Section 6.4.

Some examples are given in the following, where the aggregation operators $\boxplus$, $\boxplus_c$ and $\boxplus_g$ are taken as the Yager $t$-norm, which combines the control criteria presented in Fig. 6.7, and is given by

$$
\mu_\pi = \max \left\{ 0, 1 - \left( \sum_{i=m_1}^{n_1} (1 - \mu_\pi (e(k+i)))^{w_Y} + \sum_{i=m_2}^{n_2} (1 - \mu_\pi (\Delta u(k+i-1)))^{w_Y} + \right. \\
\left. + \sum_{i=m_3}^{n_3} (1 - \mu_\pi (\Delta \hat{y}(k+i)))^{w_Y} \right)^{1/w_Y} \right\}, \quad w_Y > 0, \quad (6.31)
$$

where the parameters $m_i$ and $n_i$, $i \in \{1, 2, 3\}$ are defined as in (6.26). The parameter $w_Y$ allows for the choice of different $t$-norms, see Section 6.4.

6.3.2 Description of the simulated systems

The influence of the conventional and the fuzzy objective functions in predictive control has been studied using two different systems:

1. a simulated nonminimum phase, open-loop unstable linear system, and

2. a simplified nonlinear model of the air conditioning system described in Chapter 8, which is derived from real data of a test cell using fuzzy modeling techniques.
The following paragraphs describe the systems in more detail. In order to concentrate on the differences between the two control schemes, model-plant mismatch and the implementational aspects are not considered at this point. However, these aspects are considered for the real-time implementation of the air conditioning system presented in Chapter 8.

**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

\[ G(s) = \frac{s - 1}{s^3 + s^2 + s + 2}. \]  

(6.32)

This is a nonminimum 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. 6.8. The system, proceeded by a zero-order-hold circuit, has been discretized with a sample time of 1s.

![Figure 6.8. Position of the poles and zeros of the linear system given by (6.32).](image)

**Air conditioning system**  A simplified version of the Heating, Ventilating and Air Conditioning (HVAC) system used as test case in Chapter 8 is considered in this example, where the air-conditioning process is modeled as a SISO system. Fig. 6.9a shows the HVAC system that is used as simulation example. The supply temperature \( T_s \) measured after the coil is controlled with the heating valve. The fan is operated at low velocity. Both dampers are half open, so that the air before the fan is a mixture between the indoor and outdoor air. A SISO model of the system is determined from input–output measurements made with a sampling period of 30s. The temperature is described as a nonlinear, first-order dynamic system \( T_s(k + 1) = f(T_s(k), u(k)) \), where \( u(k) \in [0.4, 1] \) is the valve opening and \( T_s(k) \in [30, 60] \) is the temperature in °C at time instant \( k \). A Mamdani fuzzy model with singleton consequents is obtained using the identification method described in Babuška
and Verbruggen (1995d). Fig. 6.10 shows the triangular membership functions that are determined for the temperature $T_s(k)$ and the valve opening $u(k)$. Note that the universe of discourse for the valve opening is the interval [0.4, 1] because the valve has a dead-zone behaviour 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 6.1. An example of a rule for this singleton model, as e.g. the first rule, is “If $u(k)$ is Small and $T_s(k)$ is Low then $T_s(k + 1) = 30.3$”.

Table 6.1. Rule base for the fuzzy model of the HVAC process.

<table>
<thead>
<tr>
<th>Valve Opening</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
</tr>
<tr>
<td><strong>Small</strong></td>
<td>30.3</td>
</tr>
<tr>
<td><strong>Medium small</strong></td>
<td>30.0</td>
</tr>
<tr>
<td><strong>Medium high</strong></td>
<td>32.8</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>35.5</td>
</tr>
</tbody>
</table>

Figure 6.9b depicts the piecewise linear mapping that is described by the fuzzy model. The fuzzy model is used for simulating the system and developing model-based predictive controllers. Note that a fuzzy model is used because the system is nonlinear, as shown in Fig. 6.9b.
6.3 *Fuzzy criteria for FDM in control*

![Membership functions for the valve opening](image1)

![Membership functions for temperature](image2)

(a) Membership functions for the valve opening.  
(b) Membership functions for temperature.

**Figure 6.10. Membership functions for the fuzzy model premise variables.**

### 6.3.3 Obtained results

Model-based predictive controllers are designed for the systems described in Section 6.3.2 by using both the conventional objective function (6.26) and the fuzzy objective function (6.31). Since a *t*-norm is used, the decision goal is formulated as the simultaneous satisfaction of all the decision criteria. The response of the controllers is studied using simulations of the systems. 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 limited to two time steps in order to keep the computational load low. To further reduce the computational load, a two-step optimization approach is used where first a rough solution is found by using a coarse discretization of the control space, followed by the calculation of a finer solution around the first solution.

**Linear system** The predictive control scheme is applied to the linear system given by (6.32) without any constraints on the system. Both the conventional criteria and the fuzzy criteria are then 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 $m_1 = m_3 = 1$ and $n_1 = n_3 = H_p$ was found to be sufficient for controlling the system. Note that this example is a SISO system, and the vectors $\alpha_i$, $\beta_i$ and $\gamma_i$ become scalars if they are constant for all the time steps. The following parameters are used for the conventional objective function:
\( \alpha_i = 1, \beta_i = 0 \) and \( \gamma_i = 5, i = 1, \ldots, H_p \). These values are chosen by trial and error until a reasonable response of the system is found. For the fuzzy criteria, the following membership function parameters are used: \( K_e^+ = -K_e^- = 1, K_y^+ = -K_y^- = 1, S_y^+ = -S_y^- = 0.5 \), and \( w_Y = 2 \) for the Yager \( t \)-norm. These values must also be properly tuned. The way to tune the Yager parameter is discussed in Section 6.4. 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 require only the tuning of one of them, because they are related. Finally, the parameters \( S_y^+ \) and \( S_y^- \) are chosen such that the system can move freely to a certain degree, and being penalized outside these limits, which is the interval \([-0.5, 0.5]\) in this case. Note that 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. 6.11 and Fig. 6.12, 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 \( \gamma_i \), but this occurs at the expense of amplifying the oscillations due to the nonminimum phase behaviour. 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 undesired. Hence, this system benefits clearly from the additional flexibility introduced by the fuzzy criteria. Moreover, the prediction horizon can be significantly reduced when fuzzy objective functions are used, without deteriorating the control performance.

![Diagram](image)

Figure 6.11. Step responses for the linear system using the conventional objective function.
Air-conditioning system  The air-conditioning system is simulated and a rate constraint of $|Δu| ≤ 0.1$ is imposed on the system in these experiments. In this system $H_e = 2$ and $H_p = 3$ are chosen. These horizons revealed to be sufficient for controlling the system. It is required that the controller can bring the system to any level in the interval $[30°C, 60°C]$, which is the interval where the temperatures usually range in this system. The output error with $m_1 = n_1 = 3$, the change in the control action with $m_2 = n_2 = 2$ and the change in the output with $m_3 = n_3 = 3$ are used for specifying the objective function. The second change in the control action, chosen by $m_2 = n_2 = 2$, 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 3, because it requires less control effort in the system. Moreover, the use of the two first steps deteriorates the control performance in some of the regions of this system, due to the severe nonminimum phase behaviour detected at some regions of the system.

As this system is again a SISO system, the vectors $α_i$, $β_i$ and $γ_i$ become scalars as in the linear system. The following parameters are used for the conventional objective function: $α_3 = 1$, $β_3 = 500$ and $γ_3 = 50$. The rest of the parameters are zero. The parameters $β_2$ and $γ_3$ are chosen to make a trade-off between the several criteria, and to scale the different terms: error, change in control action and change in the output. Note that the fuzzy
The 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: \( K^+_e = -K^-_e = 30, K^+_y = -K^-_y = 3, S^+_y = -S^-_y = 1 \) and \( K^+_u = -K^-_u = 0.6 \), and \( w_y = 2 \). Note that although 9 parameters are present, only 4 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 only made for energy saving and stability reasons. Finally, the parameter for the Yager \( t \)-norm, \( w_y = 2 \), allows for a good compromise between fast response and small overshoot, see Section 6.4. The responses of the air-conditioning system for several steps using classical and fuzzy criteria are shown in Fig. 6.13 and Fig. 6.14, respectively.

![Figure 6.13](image)

**Figure 6.13.** Step responses for the air-conditioning system using the conventional objective function.

The controller with the fuzzy criteria is more able to use the full range of control actions, and the response of this controller is in general faster, especially for references close to the limits of the range within which they can vary. Further, some overshoots that are noticeable with the conventional criteria are reduced.

Summarizing, for the studied systems, the use of fuzzy criteria significantly 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-based predictive control with fuzzy criteria is that the optimization problem becomes more often non-convex than using classical objective functions, increasing the computational load. Optimization problems are discussed in Chapter 7.

6.4 Choice of aggregation operators for FDM in control

A discussion about the possible aggregation operators to be used in fuzzy decision making is presented in this section. In general, the choice of the operator is application dependent. Special focus is given for control applications. This section presents some clues on the possible use of different aggregation operators, and the advantages and disadvantages of their use in predictive control. A simple, though illustrative, example shows the most relevant results obtained.

The first operator used to aggregate goals and constraints was the min operator (Bellman and Zadeh, 1970). Thus, the operators $\oplus$, $\oplus_g$ and $\oplus_c$ are all substituted by the min operator in (6.24):

$$\mu_\pi = \min \left( \mu_{\xi_1}, \mu_{\xi_2}, \cdots, \mu_{\xi_n} \right).$$

(6.33)
Although this operator is still largely used in FDM, it does not allow any tradeoff or compensation between the several criteria (Fung and Fu, 1974a), because it chooses always the smallest of the $\bar{T}$ 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 allows for some interaction amongst the criteria. The most used aggregation operator after the min operator is perhaps the product $t$-norm:

$$\mu_n = \mu_{\zeta_{i1}} \cdot \mu_{\zeta_{i2}} \cdot \cdots \cdot \mu_{\zeta_{H,T}} \cdot$$

(6.34)

This operator allows some interaction between the criteria, but keeps the characteristics of a $t$-norm, i.e. any low degree of membership for one criteria $\zeta_{ij}$ implies that the degree of membership $\mu_n$ is also low. When the number of criteria increases, $\mu_n$ tends to decrease at an exponential rate. This fact is quite realistic because the larger the number of goals and constraints is, the more difficult it is to satisfy them all. This conclusion can also be drawn for any $t$-norm, because by definition they are always smaller or equal than the min operator (see Appendix A).

Note that the presented aggregation operators assume that the importance of the various 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, see (6.26). Another possibility is to use the approach presented by Yager (1992), where each criterion has a different weight $w_{\zeta_{ij}} \in [0,1]$, reflecting a different importance in the global criterion as in (6.24). In this thesis the weights of the aggregation of fuzzy goals and constraints are not used because the systems used as examples do not have a clear hierarchy related to the importance of the different fuzzy criteria. Moreover, the weights $w_{\zeta_{ij}}$ are usually difficult to tune, especially if a large number of criteria is considered. A different approach can be followed by using parametric $t$-norms, which can generalize a large number of $t$-norms, and then control the degree of compensation between the different goals and constraints. Usually, parametric $t$-norms depend on only one parameter, which makes them much easier to tune when compared to the weighted $t$-norms. On the other hand, they are not so general as the weighted approaches. For the various practical and academical examples presented in this thesis, parametric $t$-norms revealed good control performances. Several parametric $t$-norms can be considered, such as the ones introduced by Hamacher (1978), Yager (1980), Weber (1983), Dubois and Prade (1986b) and Turksen (1986). The Yager $t$-norm, for instance, is given by:

$$\mu_n = \max \left( 0, 1 - \left\{ \sum_{i=1}^{H} \sum_{j=1}^{T} (1 - \mu_{\zeta_{ij}})^{w_Y} \right\}^{1/w_Y} \right), \quad w_Y > 0.$$ 

(6.35)

This operator is widely used because it covers the entire range of $t$-norms, i.e., it goes from the drastic intersection to the min operator.

The fuzzy decisions discussed so far are made under the assumption defined in (6.4), i.e. "$G$ must be accomplished and $C$ must be satisfied.". However, sometimes it is more appropriate
to formulate the FDM problem as follows:

\[ G \text{ must be accomplished or } C \text{ must be satisfied.} \]  \hspace{1cm} (6.36)

This type of situations are found when an agreement between different opinions must be achieved, for instance. In these cases, the fuzzy decision is given by the union of the fuzzy sets, generally described by a s-norm (t-conorm), see Appendix A. If the max operator is used, no tradeoff or interplay between criteria is allowed, similarly to the min operator for the aggregation of criteria. In fact, an alternative is selected based only on the best criteria, regardless all the others are poorly fulfilled. This feature is sometimes referred to as full compensation (Chen and Hwang, 1992). The reasoning about weights and parametric s-norms, can be made similarly to the reasoning made for t-norms. However, in this thesis no discussion about these operators is given, because they proved to be not suitable for control purposes.

There is a large range of fuzzy operators between the t-norms and s-norms which can sometimes be advantageously utilized in the confluence of fuzzy criteria, even for control purposes. Examples are the operator introduced by Zimmermann and Zysno (1980), or the generalized mean (van Nauta Lemke, et al., 1983; Kaymak and van Nauta Lemke, 1993). This last operator is given by

\[ \mu_{\pi} = \left\{ \frac{1}{T} \sum_{i=1}^{T} \sum_{j=1}^{T} w_i \mu_{i,j} \right\}^{1/w_g}, \quad w_g \in \mathbb{R}. \]  \hspace{1cm} (6.37)

It generalizes the harmonic, geometric, arithmetic and quadratic mean when the parameter is \( w_g = -1 \), \( w_g \to 0 \), \( w_g = 1 \) and \( w_g = 2 \), respectively. Moreover, when \( w_g \to -\infty \) the generalized mean becomes the minimum operator, and when \( w_g \to +\infty \) it becomes the maximum operator. Thus, this operator fulfills the complete range from minimum to maximum when \( w_g \) ranges from \( -\infty \) to \( +\infty \). When a large number of criteria is present and some tradeoff between the different criteria is allowed, this operator 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, as defined in Section 6.2.1. 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 \( \oplus \) and a t-norm for the remaining \( \odot \) and \( \ominus \). This choice assures that the 'hard' constraints are not violated but can hamper the advantages of using the generalized mean. Note that this general suggestions over the use of the generalized mean must be adapted for the particular system under control.

Some general rules to choose the appropriate aggregation operators are presented in Yager and Filev (1994), Dubois and Prade (1980a), and especially in Zimmermann (1996). Following this last author, the general guidelines deal with axiomatic strength, adaptability, numerical efficiency, degree of compensation, aggregating behaviour and required scale level of membership functions. In this thesis the choice of parameterized operators 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, alleviating the tuning phase, always present in predictive control.

### 6.4.1 Example of a linear system

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 (6.32). This example was first presented in Kaymak, et al. (1996a). Remind that this system is nonminimum 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 optimization criterion in order to keep the optimization problem transparent and understand clearly 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:

$$
\exp \left( -\frac{|r - \hat{y}(k + i)|}{30} \right),
$$

where $r$ is the reference and $\hat{y}$ is the predicted model output. Note that this function is a particular case of (6.30) for $K_e^+ = -K_e^- = 30$. A crisp constraint $|\Delta u| = 0.5$ is imposed on the rate of the control action, and it is represented by a membership function that is defined on $\Delta u$.

Step responses of the system have been studied. The controller is implemented in the incremental form and the optimization is performed in the discretized $\Delta u$ space. This control space is divided into 11 discrete levels and an enumerative search scheme has been used for determining the best control action. The control horizon $H_e$ is chosen as small as possible for keeping 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. Because the prediction horizon is 6, the multiple step decision problem has six criteria (one for each sample time). Note that even a decision problem with a single goal becomes a multicriteria decision problem in the predictive control scheme. In order to refine the discretization of the control space while keeping the computational load low, an extra optimization with finer discretization has been performed around the optimal solution at each step. The minimum operator (6.38), the generalized minimum (6.39), and the Yager $t$-norm (6.40) are chosen as possible decision functions:

$$
\mu_\mu = \min_{i=1,\ldots,H_p} \mu_i(\epsilon(k + i)) \quad (6.38)
$$

$$
\mu_\mu = \left\{ \frac{1}{H_p} \sum_{i=1}^{H_p} \mu_i(\epsilon_i(k + i)) \right\}_w \quad , \quad w \in \mathbb{R} \quad (6.39)
$$

$$
\mu_\mu = \max \left( 0, 1 - \left\{ \sum_{i=1}^{H_p} (1 - \mu_i(\epsilon(k + i))) \right\}_w \right), \quad w > 0, \quad (6.40)
$$
where \( \bar{e}(k+i) = r(k+i) - \hat{y}(k+i) \) is the predicted output error. The responses with the parametric decision functions have been calculated for several values of the parameters.

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 nonminimum phase behaviour, 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 \( \mu_x \). 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. For the generalized averaging operator as in (6.39), when \( w_g \) 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 \( w_g \). A fast response without an overshoot is obtained for \( w_g \) equal to 1 (arithmetic mean). Figure 6.15 shows the response of the controlled system for several values of the parameter \( w_g \).

\[ \begin{align*}
\text{Generalized mean} \\
\text{process output} \\
\text{time} \\
\text{control signal} \\
\text{time}
\end{align*} \]

**Figure 6.15.** Response of a fuzzy predictive controller using the generalized mean as the decision function. Dashed: \( w_g = -1 \), solid: \( w_g = 1 \), dash–dotted: \( w_g = 3 \).

Unlike the generalized mean, the system shows fast response for small values of the parameter \( w_Y \), 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 \( w_Y \) should not be chosen very small as the simultaneous satisfaction of the criteria may then be unfeasible. When \( w_Y \) 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
Figure 6.16. Response of a fuzzy predictive controller using the Yager \( t \)-norm as the decision function. Dashed: \( w_Y = 2 \), solid: \( w_Y = 2.8 \), dash–dotted: \( w_Y = 4 \).

decision operator. Fig. 6.16 shows the step response for several values of \( w_Y \). The parameter \( w_g \) (or similarly the parameter \( w_Y \)) can be interpreted as a speed indicator for the response. For small values of \( w_g \), small control actions are preferred and the system response slows down. Large values of \( w_g \) 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.

The computational effort increases exponentially with the number of discretization levels of the control space and the length of the control horizon. The control horizon should be chosen as small as possible for low computational load. The discretization of the control space can be low if a two-step optimization is applied. In this method, a rough solution is found by using a coarse discretization followed by the calculation of a finer solution around the rough solution. Optimization problems related to the computational effort are presented in Chapter 7.

6.5 Summary and concluding remarks

The application of fuzzy decision making to predictive control in close-loop control systems is considered in this chapter. First, fuzzy decision making (FDM) is described in Section 6.1.
6.5 Summary and concluding remarks

The important concepts of fuzzy goal, fuzzy constraint and fuzzy decision are defined. The presence of multiple fuzzy goals and/or fuzzy constraints is discussed afterwards, where fuzzy goals and fuzzy constraints are defined in the same set. However, usually fuzzy goals and constraints are defined in different sets. The classical approach to this problem introduced by Bellman and Zadeh (1970), as well as a different approach introduced in this thesis are presented in Section 6.1.3. Discussions on multidimensional spaces of alternatives and on the solution of the multidimensional FDM problem using discrete alternatives are addressed afterwards.

Fuzzy decision making applied to real-time control is based on multistage decision making. In this thesis, multistage FDM is defined only in the control environment, which however do not suffer from lack of generality as is justified in Section 6.2. This section starts with a discussion on the types of systems, and the termination time defined for this multistage approach, 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 discussed. The use of fuzzy criteria in model-based predictive control is also addressed.

FDM in control has two main design problems, which are the choice of fuzzy criteria and of the aggregation operators. The possible types of fuzzy criteria used for FDM in model-based predictive control are discussed in Section 6.3. The choice of the prediction horizon is discussed, and the generalization of classical objective functions to fuzzy objective functions in MBPC is presented. Two simple examples show the advantages of using fuzzy objective functions.

An important issue in FDM for control applications is the choice of the aggregation operators. A discussion on this choice is presented in Section 6.4. A simple example shows the advantages of using parameterized aggregation operators, which can influence several performance criteria, as e.g. rising time, settling time or overshoot, with only one parameter.
Optimization Problems

Model-based control usually demands for the optimization of an objective function. In the model-based control scheme most utilized in this thesis, i.e., model-based predictive control this is always the case. Section 4.1.3 presented the conditions for which this optimization problem has an analytical solution. These conditions are: a linear model of the system is used, the objective function is described by (4.2) or a similar quadratic equation, and no constraints are active. When only this last condition is not verified, i.e., some constraint is violated, no analytical solution is available. However, the optimization problem is still a quadratic problem. This problem is convex, and it can be solved using fast gradient-descent methods with a guaranteed global solution. 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 for these cases 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, which make them not suitable to be used in systems with short sampling times. Moreover, the convergence can be obtained for local minima, which results usually in poor performance of the MBPC scheme. Alternative optimization methods for non-convex optimization problems, such as dynamic programming, branch-and-bound or genetic algorithms, can be applied when the solution space is discretized.

From 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 the step $k + H_p$ to step $k + 1$. This principle of optimality is not changed when “forward” dynamic programming, i.e. from the step $k + 1$ to step $k + H_p$, is utilized. In predictive control, the on-line implementation do not allow heavy computational effort, which is usually the case when dynamic programming is utilized. Further, 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 thesis.

This chapter is divided in two main parts. The first part discusses possible approaches to solve the nonconvex optimization problems in predictive control, using nonlinear models as presented in Chapter 2. This part is divided in two sections. Section 7.1 presents a branch-and-bound algorithm applied to predictive control using a nonlinear model. A different optimization method is proposed in Section 7.2, where a genetic algorithm is used.

The second main part of this chapter discusses the optimization problem to be solved in predictive control with fuzzy decision functions, presented in Chapter 6. For some particular situations, this optimization problem remains convex. These cases are discussed in Section 7.3. However, a nonconvex problem remains to be solved for many fuzzy predictive control problems. Hence, Section 7.4 proposes a branch-and-bound algorithm for predictive control with fuzzy decision functions, which alleviates the computational effort.

### 7.1 Branch-and-bound optimization for predictive control

A widely used technique to solve difficult (usually nonconvex) optimization problems is the branch-and-bound method (Lawler and Wood, 1966; 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). In this thesis, branch-and-bound is used to solve the optimization problem in predictive control, when this problem is nonconvex. Therefore, this section presents the branch-and-bound algorithm developed for classical predictive control (Sousa, et al., 1995b; Sousa, et al., 1997c), and Section 7.4 presents a branch-and-bound algorithm for optimization in predictive control with fuzzy decision functions introduced by Sousa, et al. (1997).

A branch-and-bound algorithm can be characterized by the following three rules:

1. **Branching rule** - defines how to divide a problem into subproblems.

2. **Bounding rule** - establishes lower and upper bounds in the optimal solution of a subproblem. These bounds allow for the elimination of subproblems that do not constitute an optimal solution.
3. Selection rule - defines the next subproblem 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 to guide the memory storage of the already explored subproblems, improving significantly the efficiency of the method (Chen and Bushnell, 1996).

7.1.1 Application of B&B to predictive control

When the control actions are discretized, the branch-and-bound method can be applied to predictive control. The general MIMO model was given in Section 2.1, by Eq. (2.4). For the sake of simplicity, the B&B algorithm is presented for SISO systems, but it can be generalized for MIMO systems. 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 $u(k)$, and is given by

$$x(k) = [y(k), \ldots, y(k - p + 1), \ldots, u(k - 1), \ldots, u(k - m + 1)]^T.$$  \hspace{1cm} (7.1)

where, $m$ is the order of the input and $p$ is the order of the output. With this state vector, the model of the system under control predict the future outputs of the system $\hat{y}(k + 1), \ldots, \hat{y}(k + H_p)$, and is given by:

$$\hat{y}(k + i) = f(x(k + i - 1), u(k + i - 1)), \hspace{0.5cm} i = 1, \ldots, H_p,$$  \hspace{1cm} (7.2)

where $f$ is a function describing the system. The output values $\hat{y}(k + i), i = 1, \ldots, H_p$, are calculated based on the state vector at time instant $k + i - 1$ and the future control signal $u(k + i - 1)$, which are determined by optimizing a given objective function. Note that the state vector at time step $k + i - 1$ contains predicted and real input and output values. Let the possible inputs of the system be discretized in $M$ possible control actions. Let also the discretized control actions be denoted $\omega_j$. Thus, at each step the control actions $u(k + i - 1) \in \Omega$, are given by

$$\Omega = \{\omega_j \mid j = 1, 2, \ldots, M\}.$$  \hspace{1cm} (7.3)

Note that this set can be seen as the set of possible alternatives in a multidimensional fuzzy decision making problem, see Section 6.1.3, Eq. (6.21). Thus, the problem considered here is just a particular case of fuzzy decision making in control presented in Section 6.2, where the classical objective function (4.2), introduced in Section 4.1, is utilized.

Branch-and-bound methods can be visualized by a search tree, see Fig. 7.2. In this figure, the subsequent steps are depicted, and each point indicates a possible value for the predicted outputs $\hat{y}(k + i), i = 1, \ldots, H_p$. In predictive control, the problem to be solved is normally
Figure 7.1. Branch-and-bound optimization applied to predictive control.

represented by the objective function in (4.2), which will be used as the objective function to be optimized:

$$J = \sum_{i=1}^{H_p} \alpha_i (\hat{e}(k+i))^2 + \sum_{i=1}^{H_c} \beta_i (\Delta u(k+i - 1))^2,$$  \hfill (7.4)

where $\hat{e}(k+i)$ is the predicted output error given by $\hat{e}(k+i) = r(k+i) - \hat{y}(k+i)$. Other objective functions can also be considered, as e.g. the one presented in Section 6.3.1, Eq. (6.26). The optimization problem is successively decomposed by the branching rule into smaller subproblems.

A subproblem in the middle of the tree can be defined as follows. At time instant $k+i$ the cumulative cost of a certain path followed so far, and leading to the state $x(k+i)$ and output $\hat{y}(k+i)$ is given by

$$J^{(i)} = \sum_{i=1}^{i} \left[ \alpha_i (r(k+\ell) - \hat{y}(k+\ell))^2 + \beta_i (\Delta u(k+\ell-1))^2 \right],$$  \hfill (7.5)

where $i = 1, \ldots, H_p$, denotes the level corresponding to the time step $k+i$, see Fig. 7.2. A particular branch $j$ at level $i$ is created if the cumulative cost $J^{(i)}(u_i)$ plus a lower bound on the cost from the level $i$ to the terminal level $H_p$ for the branch $j$, denoted $J_{L_i}$, is lower than an upper bound of the total cost, denoted $J_U$:

$$J^{(i)} + J_{L_i} < J_U.$$  \hfill (7.6)

Let the total number of branches verifying this rule at level $i$ be given by $N$. Figure 7.2 illustrates the subproblem at level $i$, where a particular state and output are considered. In
order to increase the efficiency of the B&B method it is required that this number should be as low as possible, i.e. \( N \ll M \). Note that in the worst case \( N = M \), and all the possible branches are generated for the different alternatives \( \omega_i, i = 1, \ldots, M \). The lower bound can be expressed as a sum of two terms:

\[
J_{L,i} = J_j^{(i)}(\omega_j) + J_L(i + 2)
\]  

(7.7)

The first term, \( J_j^{(i)}(\omega_j) \), is the cost associated with the transition \( \hat{y}(k + i + 1) = f(x(k + i), \omega_j) \), which is computed by evaluating the respective element in the cost function in Eq. (7.5). The second term is an estimated lower bound of the cost over the remaining steps \( i + 2, \ldots, H_p \), denoted \( J_L(i + 2) \), which is generally not known and must be estimated. Note that no branching takes place for \( i > H_c - 1 \) (beyond the control horizon), i.e., the last control action \( u(k + H_c - 1) \) is applied successively to the model, until \( H_p \) is reached.

In order to achieve \( N \ll M \), 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, in order to decrease the number of new branches \( N \).

Note that until now only the branching rule and the bounding rule are 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 used in this thesis, 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 subproblems are branched from, such that the branching is done from the subproblem 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 \( J_U = J^{(\Pi_c)} \) must be estimated. The cost \( J^{(\Pi_c)} \) is derived by branching \( M \) times (for all possible control actions) at each level \( i \), starting at level 0.
The smallest $J^{(*)}_j(\omega_j)$ is chosen at each level $i$. 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 (7.6). The lower bound on the cost over the remaining steps $J_L(H_c)$ must then be estimated. The cost associated with the transition to $\hat{y}(k + H_c)$ 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 (7.6). The new nodes are at level $H_c$, and must be compared with the optimal solution found so far (and giving $J_U$). 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 repeats 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 7.1.1.

### 7.1.2 Application of the B&B method to nonlinear control

An illustrative example is presented comparing the performances of the branch-and-bound algorithm and the sequential quadratic programming (SQP) algorithm, see e.g. Gill, et al. (1981). This example is presented in Sousa, et al. (1997c). The TS fuzzy model derived for an air-conditioning system, and presented in Section 8.3.1 is used as 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 7.3 gives the sum of squared errors (SSE) as a performance measure, 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 presents 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 regard to the computational costs if the control horizon is kept small. This conclusion was confirmed by experiments with other systems as well (Sousa, et al., 1995b).

This example shows that for large control horizons, another optimization method is necessary.
Algorithm 7.1.1 (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 $\omega_j, j = 1, \ldots, M$.

**Step 1: Initialize algorithm.** At each level $i$ (time $k + i$), starting from level 0, the smallest $J_j^{(i)}(\omega_j)$ is chosen, and branching is made for all possible discrete control actions $M$. The best cost at step $H_p$, $J^{(H_p)}$ is chosen as the initial lower bound:

$$J_U = J^{(H_p)}$$

The remaining $M - 1$ nodes created at level $H_c$ are eliminated because they do not constitute an optimal solution. The algorithm goes to level $H_c - 1$.

**Step 2: Estimate lower bound.** The algorithm is at level $i$. The branch $j$ with the best cost function $J_j^{(i)}$ found so far, and that is not fully explored, is chosen. This means that if the best cost function at level $i$ has already branches to level $i + 1$, the second best value for $J_j^{(i)}$ must be chosen.

The lower bound $J_{L_j}$ is estimated by:

$$J_{L_j} = J_j^{(i)}(\omega_j) + J_L(i + 2),$$

with $J_j^{(i)}(\omega_j) = \alpha_i(v(k + i + 1) - \hat{y}(k + i + 1))^2 + \beta_i(\omega_j - u(k + i))^2$, and $\hat{y}(k + i + 1) = f(x(k + i), \omega_j)$. The lower bound on the cost of the remaining steps must be estimated. If no estimation is available, it is simply set to zero: $J_L(i + 2) = 0$.

**Step 3: Apply branch condition.** The branching condition

$$J_j^{(i)} + J_{L_j} < J_U,$$

is applied to the considered branch at level $i$ for all $j, j = 1, \ldots, M$ discretized control actions $\omega_j$. This procedure generates $N$ branches. If no branch is generated go to step 6.

**If** $i + 1 = H_c$,

**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 this $N$ 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 $J_j^{(i)}(\omega_j)$ and go to the next level ($i \rightarrow i + 1$). 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 explored, the optimal solution is the optimal solution found so far, and the algorithm stops.
Figure 7.3. Sum squared error and FLOPS for the optimization methods. Solid line – B&B optimization; dashed line – sequential quadratic programming.

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 hampering the use of the B&B algorithm. A possible non-convex optimization method is presented in Section 7.2, where genetic algorithms are used for optimization in predictive control.

7.1.3 Some considerations on the B&B method applied to MBPC

The proposed algorithm was applied to different systems (Sousa, et al., 1995b; Babuška, 1997), and the results were compared to other optimization techniques, such as Sequential Quadratic Programming (SQP). A comparison for the air-conditioning system introduced in Section 5.3, is presented in Section 7.1.2. The experience shows that the B&B algorithm even with a bad lower bound estimate, $J_L(i + 2)$, is in general faster and more accurate than enumerative search, which explores the complete search space, and 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 must be kept low. Another factor of extreme importance is the number of control alternatives $M$. For computational reasons this number should be as small as possible, however, if $M$ is too small, the coarse discretization of the control signal results in poor control performances. Therefore, a good compromise between computational time and size of $M$ must be made for each control problem.
The three major advantages of the B&B algorithm applied to predictive control over other nonconvex optimization methods are the following:

1. The global discrete minimum containing the optimal solution, or a value close to it, is found, guaranteeing a 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 subproblems. Most optimization algorithms, as e.g. 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 was described in Section 5.2 where a predictive control strategy is combined with inverse control.

### 7.2 Genetic algorithms for optimization in predictive control

One of the techniques that seems to be especially suitable for constrained, nonconvex 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. The 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). However, the literature does not cover the application of GA to model-based predictive control, even though optimization is an important part of model-based predictive controllers. This can partially be explained by the numerical complexity of the GA, which, for the time being, make them suitable only for processes with slow dynamics. Yet, GA may become 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 is based on the paper (Onnen, et al., 1997). It investigates the application of GA to the determination of an optimal control sequence in MBPC. 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. A simulated pressure dynamics of a batch fermenter is considered as an example, where the simulation results with GA are compared with those obtained with the branch-and-bound method, in terms of the achieved control accuracy and computational costs.

### 7.2.1 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 robust with respect to the complexity of the optimization problem compared to the 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 also been reported (Davis, et al., 1994; Gerdes, 1994; Petry, et al., 1994; Davidor, 1991), especially in combination with other artificial intelligence techniques such as neural networks and fuzzy systems. The importance of genetic algorithms in the field of control is increasing, as can be seen from a number of recent publications (Linkens and Nyongesa, 1995; Kacprzyk, 1995b; Surmann, et al., 1993; Karr, 1992).

**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. In accordance with 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 with 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 depend both on this function.

The genetic algorithm evaluates a number of solutions (values) and then generate 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 7.4 depicts a schematic representation of a genetic algorithm. The terms in this figure are explained in the sequel.

![Diagram of a genetic algorithm]

**Figure 7.4. 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 noticed 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 in 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 to mate, 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; multi-point 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 for stopping 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.

**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 (Schoenauer
and Xanthakis, 1993) 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.

### 7.2.2 Application of genetic algorithms to predictive control

When nonlinearities and constraints are present in MBPC, usually a non-convex optimization problem must be solved at each sampling period. The control horizon determines directly the dimension of the optimization problem, which may thus become very complex. Different algorithms such as sequential quadratic programming (SQP) or branch-and-bound (Sousa, et al., 1997c), can be used to circumvent this problem. 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 the use of a genetic algorithm, because this is a robust optimization technique that is able to deal with highly complex nonlinear and non-convex optimization problems. However, a GA 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.

The first task in the GA design is to specify the principles for the fitness evaluation and the representation of the variables in genes. In the second step, the genetic operators are chosen and tuned. In the approach presented here, the objective function to be minimized is the objective function usually applied in predictive control, given by Eq. (4.2) in Section 4.1. 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 system is thus given by the following objective function:

\[ J = \sum_{i=1}^{H_p} \alpha_i (r(k+i) - \hat{y}(k+i))^2 + \sum_{i=1}^{H_c} \beta_i (\Delta u(k+i - 1))^2, \]  

(7.8)
which accounts for minimizing the variance of the process output from the reference, minimizing at the same time the energy. The compromise between the two goal is given by the choices of \( \alpha \) and \( \beta \). 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 following transformation:

\[
f = \frac{1}{1 + J}.
\]  

(7.9)

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. The first step necessary to apply GA to predictive control is presented in the next paragraph, where a feasible coding principle that is able to cope with constraints and with the specific characteristics of the variables used in predictive control is derived.

**Encoding control variables and implementing constraints.** A straightforward coding principle is to represent each change \( \Delta u \) 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 \( k \), the first gene in a chromosome encodes the change in the control action \( \Delta u(k) \) to be added to \( u(k - 1) \). 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...0 and 11...1 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 the GA, because of the waste of genetic material, due to the infeasible solutions in the population. Another approach is proposed here, which implements both the rate and level constraints in the coding mechanism.

Let \( u_{\text{max}} \) and \( u_{\text{min}} \) denote the level constraints, i.e., the maximum and minimum allowed control actions. Similarly, let \( \Delta u_{\text{+}} \) and \( \Delta u_{\text{-}} \) 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 \( k + i - 1 \), with \( 1 \leq i \leq H_c \) (corresponding to gene \( i \)), the maximum negative and positive changes from the previous control action \( u(k + i - 2) \) are given by:

\[
\Delta U_{\text{sup}}(i) = \min[\Delta u_{\text{+}}, u_{\text{max}} - u(k + i - 2)],
\]

\[
\Delta U_{\text{inf}}(i) = \min[\Delta u_{\text{-}}, u(k + i - 2) - u_{\text{min}}].
\]  

(7.10)
Figure 7.5. Genes encoding relative change values with the possible values for \( u(i) \).

Figure 7.5 illustrates the principles of this constraint implementation. The maximum positive and negative changes \( \Delta U_{\text{inf}}(i) \) and \( \Delta U_{\text{sup}}(i) \) are now discretized in \( n_d \) values each. The set of possible changes in the control action, \( \Delta U(i) \) is given by:

\[
C_- = \left\{ -\Delta U_{\text{inf}}(i) \frac{n_d - j}{n_d} \middle| j = 0, 1, \ldots, n_d - 1 \right\},
\]

\[
C_+ = \left\{ \Delta U_{\text{sup}}(i) \frac{n_d - j}{n_d} \middle| j = 0, 1, \ldots, n_d - 1 \right\},
\]

\[
\Delta u(k + i - 1) \in \{ C_-, 0, C_+ \} .
\]

(7.11)

These values are coded such that the string \( 00 \ldots 0 \) corresponds to \( \Delta u(k + i - 1) = -\Delta U_{\text{inf}}(i) \) and \( 11 \ldots 1 \) corresponds to \( \Delta U_{\text{sup}}(i) \). Note that the genes are encoded sequentially, from the first one corresponding to \( \Delta u(k) \) to the last one corresponding to \( \Delta u(k + H_c - 1) \). This procedure makes sure that all chromosomes encode valid control sequences, avoiding the waste of genetic material.

**GA operators.** The GA 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 GAs 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 this application, 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 Fig. 7.6. 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.

<table>
<thead>
<tr>
<th>crossover mask</th>
<th>inverted mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>11100100</td>
<td>00011011</td>
</tr>
</tbody>
</table>

Parent 1

<table>
<thead>
<tr>
<th>Parent 1</th>
<th>10100101</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓↓↓</td>
<td>↓</td>
</tr>
</tbody>
</table>

Offspring 1

<table>
<thead>
<tr>
<th>Offspring 2</th>
<th>01100001</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑↑↑</td>
<td>↑</td>
</tr>
</tbody>
</table>

Parent 2

<table>
<thead>
<tr>
<th>Parent 2</th>
<th>01101011</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑↑↑</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.6. The uniform crossover.**

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.

**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 GA's success. 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 $2n_u + 1$, the control horizon $H_c$ (equal to the number of genes per chromosome $N_g$), the gene length $L_g$ and a constant $K$ to be properly chosen:

$$N_p = K \cdot (N_g + L_g).$$

(7.12)

Various tests have been made for the presented application, and $K = 8$ 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 GAs. 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.

**Figure 7.7. 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 $k$, the genes from $k+2$ to $k + H_c$ are shifted one position, and the last gene takes the same value as at $k + H_c$, as can be seen in Fig. 7.7. 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 order 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 7.8 shows the described population structure, including also 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.

**Termination conditions.** Fixing the number of generations per evolution may restrict the GA'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 GA 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. (7.9).

- **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 \( N_{f_w} \) generations. Note that the first gene represents the current control input \( u(k) \).

In this paper, 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.

7.2.3 Application example

The predictive control scheme based on GA optimization was applied in the simulation of pressure control in a laboratory fermenter. This system is described in Section 5.3. The nonlinear differential equation given by Eq. (5.45), and reminded here, is used for the simulation model of the system:

$$\frac{dP}{dt} = \frac{1000.\frac{RT}{22.4.V_h}}{\Phi_g - (\pi R_l^2)\sqrt{\frac{2P_0}{\rho_0 K_p} \ln\left(\frac{P}{P_0}\right)}}.$$ 

The symbols and respective values are described in Section 5.3. The maximum changes in the valve position are $\Delta u_+ = \Delta u_- = 10\%$ of the total range per sample and the level constraints are $u_{\text{min}} = 0\%$ and $u_{\text{max}} = 100\%$ 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, providing thus 5 possible changes in the control action. For the sake of simplicity, only the error criterion is considered in Eq. (7.8), and thus $\alpha_i = 1$ and $\beta_i = 0$, with $i = 1, \ldots, H_p$. Fig. 7.9 shows an example of a typical simulation experiment.

![Graphs showing pressure and valve position over time](image)

**Figure 7.9.** 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 used by Sousa, et al. (1995b). The branch-and-bound uses five equal discretization intervals, contrary to the GA discretization method, which uses different discretization intervals, see Section 7.2.2. Three comparison criteria are considered:

1. CPU time (in seconds),
2. FLOPS (floating-point operations), and
3. control accuracy (sum-squared error).

Figure 7.10 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.

![Figure 7.10. Comparison of the computational costs in terms of CPU time (- -) and FLOPS (—) for the GA (○) and the B&B method (×).](image)

Figure 7.11 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.

### 7.2.4 Summary and discussion

Summarizing, the genetic algorithm designed to cope with the specific requirements of predictive control has the following characteristics.
Figure 7.11. Comparison of the sum-squared error calculated over the complete reference: for the GA (o) and the B&B method (x).

- 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.

The performance of the designed GA has been compared to a branch-and-bound technique, which is also a discrete optimization method. Experimental results show that the 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. Currently, GA optimization for MBPC can best be applied to processes with relatively slow dynamics and long control horizons. Note that for MIMO systems the computational costs increase linearly with GAs and exponentially with the B&B method. Therefore, it is expected that GAs will outperform B&B for multivariable systems.
7.3 Convex optimization in fuzzy predictive control

In model-based predictive control using 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 nonconvex optimization which is computationally not tractable with conventional optimization techniques. As it is often not possible to find a global optimum in nonconvex optimization, and as it requires large computational effort, fuzzy predictive control can usually only be applied to systems with slow dynamics, where the sample 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 first presented in Sousa, et al. (1996). The fuzzy predictive control scheme with the convex optimization problem is applied 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 \( \mu_y \) as described in Section 6.2.2, Eq. (6.24). 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 Section 3.1, Eq. (3.2). The fuzzy multicriteria decision making problem is defined as an unconstrained problem for the formulation presented in (3.2), 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 (6.25) in Section 6.2.2, to be a convex programming problem, is that the function \( J(v) \) in Eq. (3.2) is a convex function. For convex programming problems, any local minimum \( v^* \) is the global minimum, reducing considerably the computational effort. As the optimization must be done on-line in fuzzy predictive control, 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 nonlinear SISO system, and is described by the auto-regressive linear model:

\[
y(k + 1) = \sum_{i=1}^{p} a_i y(k - i + 1) + \sum_{i=1}^{m} b_i u(k - i + 1),
\]

(7.13)

where \( y(k), \ldots, y(k - p + 1) \) and \( u(k), \ldots, u(k - m + 1) \) are the shifted model outputs and inputs, respectively, with \( p \) and \( m \) as 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
7.3 Convex optimization in fuzzy predictive control

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:

$$
\mu(r(k+i) - \hat{y}(k+i)) = \max \left(1 - \frac{|r(k+i) - \hat{y}(k+i)|}{K}, 0\right),
$$

(7.14)

where $r(k+i)$ is the reference, $\hat{y}(k+i)$ is the predicted model output, $i = 1, \ldots, H_p$, and $K$ is the spread of the membership function. The 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 $\mu_n$ 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 $\mu_n$ does not become zero (assuming that a feasible solution exists within the constrained set). For the system (7.13) under study, any value that does not lead to a non 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 $u(k) \times \cdots \times u(k + H_e - 1)$, 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 using the resolution principle of fuzzy sets (Klir and Folger, 1988). The Yager $t$-norm given, e.g., in (6.35) 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 7.3.1 (Convex optimization in fuzzy predictive control)** Let the system to be controlled be described by (7.13), 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 (7.14). Further, let the constraints on the optimization space $u(k) \times \cdots \times u(k + H_e - 1)$ be crisp (not fuzzy) and convex, and finally let the Yager $t$-norm in (6.35) 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:** The equation (7.13) can be rewritten as an affine function of $u$ for the predicted outputs and for a particular point $j$ in the prediction horizon:

$$
\hat{y}(k + j) = y_0 + \sum_{\ell = 1}^{j} a_{j \ell} u(k + \ell - 1)
$$

(7.15)

with $j = 1, \ldots, H_p$. The variable $y_0$ (constant at time $k$ for a particular value of $j$) depends on the parameters $a_i$, $b_i$, the output values $y(k), \ldots, y(k - p + 1)$ and the control actions
\( u(k), \ldots, u(k - m + 1) \). The values for \( \alpha_{k} \) are related to the parameters of the impulse response and can be derived from \( \alpha_{i} \) and \( b_{i} \). The error is given by

\[
\hat{e}(k+j) = r(k+j) - \hat{y}(k+j)
\]

and it can be written also as an affine function of \( u \):

\[
\hat{e}(k+j) = c_{0} + \sum_{\ell=1}^{j} \alpha'_{\ell} u(k + \ell - 1)
\]  

(7.16)

with \( c_{0} = r(k + j) - y_{0} \) and \( \alpha'_{\ell} = -\alpha_{\ell} \). Considering that the error remains inside the universe of discourse defined, the membership function for the error defined in (7.14) can be described by

\[
\mu(\hat{e}(k+j)) = 1 - \frac{|\hat{e}(k+j)|}{K}.
\]  

(7.17)

Substituting (7.16) in (7.17) one obtains

\[
\mu(\hat{e}(k+j)) = 1 - \frac{|c_{0} + \sum_{\ell=1}^{j} \alpha'_{\ell} u(k + \ell - 1)|}{K}.
\]  

(7.18)

Depending on whether the error is positive or negative, Eq. (7.18) can be written in an affine form:

\[
\mu(\hat{e}(k+j)) = c'_{0} + \sum_{\ell=1}^{j} \beta_{\ell} u(k + \ell - 1),
\]  

(7.19)

where for \( c(k+j) > 0 \), \( c'_{0} = 1 - \frac{c_{0}}{K} \) and \( \beta_{\ell} = -\alpha'_{\ell} \), and for \( c(k+j) < 0 \), \( c'_{0} = 1 + \frac{c_{0}}{K} \) and \( \beta_{\ell} = \alpha'_{\ell} \).

Since the optimal policy is found by a maximization in fuzzy decision making, the term

\[
\left\{ \sum_{i=1}^{H_{P}} (1 - \mu_{i}(\hat{e}(k + i)))^{\omega_{Y}} \right\}^{1/\omega_{Y}}
\]  

(7.20)

should be minimized. This is a \( \omega_{Y} \)-norm of \( 1 - \mu_{i}(\hat{e}(k+j)) \). Since

\[
1 - \mu_{i}(\hat{e}(k + i)) = \frac{|\hat{e}(k+j)|}{K}
\]

is an affine function of \( u \) according to (7.18), Eq. (7.20) is a \( \omega_{Y} \)-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 \( \ell \)-norm with the given membership functions and the possible convex crisp constraints results in convex optimization.

\[ \square \]

### 7.3.1 Algorithms for convex optimization

The controller design problem stated in the previous section 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. Moreover, many of the descent methods use heuristic stopping criteria as when applied to nonconvex optimization algorithms. A survey of these methods can be found in Luenberger (1984). Another possibility is to use cutting-plane or ellipsoid methods. These methods are described e.g. by Boyd and Barret (1991), and require the evaluation of function values and any subgradients of functions. These methods 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 Section 7.3.2), many of the descent methods present faster convergence. From the methods using 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 quasi-Newton methods consists of the different ways of computing the update of the Hessian matrix presented in the quadratic formulation.

7.3.2 Example of application

The described fuzzy optimization environment is applied to the control of the simulated non-minimum phase, open-loop unstable linear system, presented previously in Section 6.3.2, and described by (6.32). The sampling time is 1s. The prediction horizon \( H_p \) is chosen as 6 (related to the settling time) and the control horizon \( H_c \) as 2; 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 \( w_Y \) have been studied. The results are shown in Fig. 7.12.

A convex programming technique is used for the optimization at each step for several values of \( w_Y \). In this example, a gradient descent method is utilized, providing fast convergence as required. A gradient method using the updating method of the Hessian matrix as described in Fletcher (1970) is used. An example of a surface resulting from the optimization of (7.20) is plotted in Fig. 7.13 for the time step \( k = 7 \) and for \( w_Y = 2.8 \). This figure represents the resulting surface as a function of the first two control actions \( u(k+1) \) and \( u(k+2) \) (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.
Figure 7.12. Response of a fuzzy predictive controller using Yager $t$–norm as the decision function. Dashed: $w_Y = 1.9$, solid: $w_Y = 2.8$, dash-dotted: $w_Y = 4$.

Figure 7.13. Example of the cost function defined in (7.20) at time step $k = 7$ ($t = 7s$) with $w_Y = 2.8$ for the control actions $u(k + 1)$ and $u(k + 2)$. 
7.3.3 Discussion

It has been shown that 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 nonconvex optimization techniques that have been used in fuzzy predictive control so far.

The results presented in this section show that, in particular situations, the advantages of fuzzy predictive control can be combined with the advantages of convex optimization for which efficient computational algorithms are available.

7.4 Branch-and-bound optimization for predictive control with fuzzy decision functions

In fuzzy predictive control the membership functions for the fuzzy criteria can have an arbitrary shape and the decision function is usually nonlinear, which results most often in a nonconvex optimization problem. Hence, convex optimization algorithms such as the ones presented in Section 7.3.1 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 of a solution can be performed using the branch-and-bound (B&B) method. Previously, a B&B algorithm for classical model-based predictive control applications was presented in Section 7.1. This section presents the application of the B&B algorithm to fuzzy predictive control, i.e., model-based predictive control with fuzzy decision criteria in the objective function.

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 $\hat{y}(k+1), \ldots, \hat{y}(k+H_p)$ was given in Section 7.1, Eq. (7.2), which is reminded here for the sake of clarity:

$$\hat{y}(k+i) = f(x(k+i-1), u(k+i-1)), \quad i = 1, \ldots, H_p.$$  

The considerations made in Section 7.1 about this model remain valid here. The control actions $u(k), \ldots, u(k+H_c-1)$ are discretized in $M$ possible input values, as in Section 7.1:

$$\Omega = \{\omega_j | j = 1, 2, \ldots, M\}.$$
The tree represented in Fig. 7.1 remains also valid, and at each time step (level of the tree shown in Fig. 7.1), \( M \) control alternatives are considered, yielding a maximum of \( M \) branches.

In fuzzy predictive control the objective function is defined as the aggregation of fuzzy goals and constraints (fuzzy criteria). The criterion \( \zeta_{it} \) denotes the \( \ell \) criterion considered at time step \( k + i \), with \( i = 1, \ldots, H_p \) and \( \ell = 1, \ldots, T \), where \( T \) is the total number of fuzzy criteria. These definitions were presented in Section 6.2.1. The aggregation of the different criteria is given by Eq. (6.24), presented also in Section 6.2.2. In order to apply the branch-and-bound method to fuzzy predictive control, the aggregation operators \( \oplus_{\mathfrak{a}}, \oplus_{\mathfrak{e}} \) and \( \odot \) must be \( t \)-norms. These norms guarantee that the membership degree of the policy \( \mu_{\pi} \) decreases with the time, and the corresponding cost functions increase, which is a necessary condition to apply the B&B algorithm.

Let \( i = 0, 1, \ldots, H_p \) denote the \( i \)th level of the tree \( (i = 0 \) at the initial node) and let \( j \) denote the branch corresponding to the control alternative \( \omega_j \). The partial optimization problem for a branch \( j \) at level \( i \) is defined by the maximization of the criteria at this point. Defining it in a recursive way, one obtains

\[
\mu_{\zeta}^{(i)}(\omega_j) = t \left( \mu_{\zeta}^{(i-1)}(\mu_{\zeta,i}(\omega_j), \ldots, \mu_{\zeta,T}(\omega_j)) \right).
\]

(7.21)

The operator \( t \) represents a triangular norm, and thus, \( \mu_{\zeta}^{(i)} \) is a decreasing function with respect to \( i \). 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 \( \mu_j^{(i)}(\omega_j) \) be the fuzzy complement of the membership degree representing the confluence of decision criteria:

\[
\mu_j^{(i)}(\omega_j) = \overline{\mu_{\zeta}^{(i)}},
\]

(7.22)

where \( \overline{\cdot} \) stands for the fuzzy complement. This value can be seen as a cost because it increases with \( i \); it is actually the complement of a decreasing function. In formal terms, the partial optimization problem for a branch \( j \) at level \( i \) is formulated as follows:

\[
\text{minimize} \quad \mu_j^{(i)}(\omega_j),
\]

with \( \omega_j \in \Omega \).

(7.23)

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 Section 6.2.1. Moreover, constraints on the control actions are directly applied when the \( M \) discrete control actions are chosen. Application of the branching alone would
result in the search of the entire tree (enumerative search), i.e., \( M^{H_c} \) possibilities, which is computationally prohibitive, except for very small control horizons. In order to reduce the number of alternatives, bounding is applied. At level \( i \), the degree of satisfaction for the decision criteria is known and is given by \( \mu_i^{(i)} \), as in (7.21). Let \( \mu_{i+1}^{(i+1,\ldots,H_p)} \) be an upper bound of the remaining degree of satisfaction for the levels \( i+1,\ldots,H_p \). Similar to the cost \( \mu_j^{(i)}(\omega_j) \) for branch \( j \) at level \( i \), the membership value of a lower bound for the remaining cost can be given by the complement of \( \mu_{i+1}^{(i+1,\ldots,H_p)} \):

\[
\mu_{i+1}^{(i+1,\ldots,H_p)} = \mu_i^{(i)}.
\]

(7.24)

Let \( \mu_{J_i} \) be an upper bound for the total cost, that is given by the complement of the membership degree representing a lower bound on \( \mu_i^{(H_p)} \). This lower bound is the confuence of decision criteria when an entire path has been followed. A particular branch \( j \) at level \( i \) is followed if the cost at level \( i \) aggregated with the lower bound on the remaining cost \( \mu_{J_k}^{(i)} \) is smaller than \( \mu_{J_i} \), i.e. if

\[
t\left(\mu_i^{(i)}, \mu_{i+1}^{(i+1,\ldots,H_p)}\right) = s\left(\mu_i^{(i)}, \mu_{i+1}^{(i+1,\ldots,H_p)}\right) = s\left(\mu_{J_i}^{(i)}, \mu_{J_k}^{(i)}\right) < \mu_{J_i},
\]

(7.25)

where \( s \) is the dual of the \( t \)-norm (\( s \)-norm) used in (7.21) (Klir and Yuan, 1995). For the sake of clarity, the dependency on \( \omega_j \) is not explicitly shown in (7.25). The transition between level \( i \) and level \( i + 1 \) is depicted in Fig. 7.14. The efficiency of the bounding mechanism depends on the quality of the bound estimates. The upper bound \( \mu_{J_i} \) should be as close as possible to the optimum and the lower bound \( \mu_{J_k}^{(i)} \) 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 7.4.1.

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 (7.3). The B&B optimization technique applied to fuzzy predictive control finds always 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 \( M \) 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, \( M \) 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.
Algorithm 7.4.1 (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 $\omega_j$, $j = 1, \ldots, M$.

**Step 1:** Initialize algorithm. At each level $i$ (time $k + i$), starting from level 0, the smallest $\mu^{(i)}_j(\omega_j)$ is chosen, and branching is made for all possible discrete control actions $M$. The best cost at step $H_p$, $\mu^{(H_p)}_j$ is chosen as the initial lower bound:

$$\mu_{J_U} = \mu^{(H_p)}_j$$

The remaining $M - 1$ 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 $i$. The branch $j$ with the best cost function $\mu^{(i)}_j$ found so far is chosen. The lower bound $\mu^{(i)}_{J_L}$ is estimated by:

$$\mu^{(i)}_{J_L} = \frac{\mu^{(i+1, \ldots, H_p)}_{J_U}}{\mu^{(i+1, \ldots, H_p)}_{J_U}}.$$

If no information over $\mu^{(i+1, \ldots, H_p)}_{J_U}$ is available, this lower bound is set to zero: $\mu^{(i)}_{J_U} = 0$.

**Step 3:** Apply branch condition. The branching condition

$$s \left( \mu^{(i)}_j, \mu^{(i)}_{J_L} \right) < \mu_{J_U},$$

is applied to the considered branch at level $i$ for all $j$, $j = 1, \ldots, M$ discretized control actions $\omega_j$. This procedure generates $N$ branches. If no branch is generated go to step 6.

If $i + 1 = H_c$,

**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, $\mu_{J_U}$, with this $N$ new costs. If a new optimal solution is found, $\mu_{J_L}$ is replaced by the new $\mu^{(i+1, \ldots, H_p)}_j$. 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 $\mu^{(i)}_j(\omega_j)$ and go to the next level ($i \rightarrow i + 1$). 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.
7.4 Branch-and-bound optimization for predictive control with fuzzy decision functions

Figure 7.14. Branch-and-bound optimization in fuzzy predictive control

7.4.1 Application example

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 Section 6.3.2. The description of the system and the simulations 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 \( T_s(k + 1) \), is modeled by \( T_s(k + 1) = f(T_s(k), u(k)) \), and where \( u(k) \in [0, 1] \) 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 \( \hat{e}(k + i) = r(k + i) - \hat{y}(k + i) \) is chosen, where \( r(k + i) \) are the future references. This criterion is represented by the triangular membership function \( \mu_e(\hat{e}(k + i)) \) as given in Fig. 7.15.

![Figure 7.15. Membership function corresponding to the minimization of the prediction error.](image)

The degrees of satisfaction of the criterion for the \( \Pi_p \) steps is combined using the paramet-
erized family of \(t\)-norms introduced by Yager (1980):

\[
\mu_x = \max \left( 0, 1 - \left( \sum_{i=1}^{H_p} (1 - \mu_x(\hat{e}(k + i)))^{w_y} \right)^{1/w_y} \right), \quad w_y > 0. \tag{7.26}
\]

The Yager operator applied to fuzzy criteria, has been studied in Section 6.4, and it is known that it leads to good control results. The lower bound is chosen as \(\mu_{\text{ij}} = 0, \forall i\). 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, \(\Delta u(k + i - 1)\), with \(i = 1, \ldots, H_p\), 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 7.1 gives the computational costs in two different ways: computational time (CT) and number of floating-point operations (FLOPS), for 11 possible control actions at each step. The table presents simulation runs using Matlab on a 133MHz Pentium PC with 16 Mbytes of memory and running under the Windows 95 environment.

**Table 7.1. Comparison of branch-and-bound and enumerative search for different prediction horizons.**

<table>
<thead>
<tr>
<th>Control horizon</th>
<th>B&amp;B CT</th>
<th>B&amp;B FLOPS</th>
<th>Enum. Search CT</th>
<th>Enum. Search FLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H_c = 1)</td>
<td>1</td>
<td>1</td>
<td>0.98</td>
<td>0.89</td>
</tr>
<tr>
<td>(H_c = 2)</td>
<td>2.87</td>
<td>2.95</td>
<td>6.53</td>
<td>12.5</td>
</tr>
<tr>
<td>(H_c = 3)</td>
<td>7.77</td>
<td>7.80</td>
<td>54.6</td>
<td>134.8</td>
</tr>
<tr>
<td>(H_c = 4)</td>
<td>22.4</td>
<td>21.8</td>
<td>445</td>
<td>1400</td>
</tr>
</tbody>
</table>

The computational requirements of the B&B method for \(H_c = 1\) are taken as 1 (100%), where FLOPS=402 and CT=0.04s for one time step. The comparison is made for control horizons from 1 to 4 steps. As Table 7.1 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 7.2 presents the increase of the computational cost with the number of discretizations used for the control actions \(u(k + i - 1)\), and for \(H_c = 2\).

The number of control actions hampers the application of enumerative search 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.
Table 7.2. Comparison of branch-and-bound and enumerative search for several number of discretizations.

<table>
<thead>
<tr>
<th>Number of discretizations</th>
<th>B&amp;B CT FLOPS</th>
<th>Enum. Search CT FLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>2.87 2.95</td>
<td>6.53 12.5</td>
</tr>
<tr>
<td>21</td>
<td>6.10 6.71</td>
<td>24.7 47.7</td>
</tr>
<tr>
<td>51</td>
<td>20.5 25.1</td>
<td>147 292</td>
</tr>
</tbody>
</table>

Figure 7.16. Step responses for the air-conditioning system.
However, the application of branch-and-bound allows clearly for the application of fuzzy predictive control to systems with smaller sampling times than the use of enumerative search. Fig. 7.16 presents the response of the system for $H_c = 2$, $H_p = 2$ and 51 points of discretization for $u(k + i - 1)$, and for several step references.

7.4.2 Discussion

This section presents a branch-and-bound algorithm for predictive control with fuzzy objective functions, where the fuzzy goals and constraints must be combined using a $t$-norm. The global optimum within the discrete optimization space can be found, while the computational costs are reduced considerably in comparison to a full enumerative search. In this way, the applicability of the fuzzy predictive control approach is increased. However, due to the large number of model evaluations, the approach can only be applied to systems with relatively large sampling periods. Moreover, as the control actions are restricted to a finite number of levels, only an approximate solution is obtained to the overall optimization problem. A real-time application of this B&B algorithm to an air-conditioning system is presented in Section 8.4.4.

7.5 Summary and concluding Remarks

This chapter presented different optimization algorithms to be applied in fuzzy model-based control. The first part of the chapter deals with the problems in predictive control using classical objective functions when the system is nonlinear. Two optimization techniques are presented and compared: the branch-and-bound method and genetic algorithms. The last part of the chapter presents optimization solutions for predictive control with fuzzy objective functions. Special conditions under which the optimization problem remains convex are presented. An extension of the B&B algorithm for MBPC with fuzzy objective functions was introduced afterwards.

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. The number of discrete control alternatives must be chosen such that they are 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 improve the efficiency of the method by eliminating several nodes.

Genetic algorithms can also be applied when a nonconvex optimization problem is present and in the presence of constraints. GA have been applied to model-based predictive control. Some special characteristics for the GA are required for this application. Therefore, termination conditions, a coding scheme to implement level and rate constraints, a method for
7.5 Summary and concluding Remarks

initializing the population, and the introduction of a learning feature are proposed to cope with the MBPC application. 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.

The second part of this chapter addresses optimization problems in MBPC with fuzzy objective functions. First, the conditions under which the optimization problem remains convex are presented. These conditions are the following: the system under control must be linear, one fuzzy goal at each step is defined, which 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. Convex problems can be solved with small computational effort, especially when compared to nonconvex optimization problems. Thus, in some situations, the advantages of using fuzzy objective functions in predictive control can be combined with the advantages of convex optimization.

The last section of this chapter presents a branch-and-bound algorithm for predictive control with fuzzy objective functions, which is an extension of the B&B algorithm derived for classical MBPC. In order to apply this B&B algorithm, the fuzzy goals and the fuzzy constraints must be combined using a $t$-norm. The computational costs are reduced considerably in comparison to a full enumerative search. Besides this fact, the control horizon and the number of discretizations must be kept small for computational reasons. Therefore, this approach is applicable in real-time for processes with relatively large sampling periods.
Application to an Air-Conditioning System

In the last decades, the number of air-conditioning systems and heating and ventilating systems installed in buildings have been continuously increasing. By controlling the indoor climate, human comfort can be significantly increased. The 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 for 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 was presented by Ling and Dexter (1994). A fuzzy MBPC applied to the temperature control of buildings was introduced by Georgescu, et al. (1993). Note that these three approaches use linguistic rules based on expert knowledge, requiring always some trial-and-error method to tune the parameters of the controller, as discussed in Section 1.2. 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, in 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 Section 6.3, are applied to an air-conditioning
This chapter applies some of the control methodologies presented in this thesis to the temperature control of an air-conditioning system. This system, described in Section 8.2, is a good pilot system because different control strategies can be tested. Moreover, it has reasonably slow dynamics, which is important to implement control techniques requiring a nonconvex 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 on air-conditioning systems presented in Section 8.1, the pilot system (test room) considered in this chapter is described in Section 8.2. Section 8.3 presents the several models developed for this system. Section 8.4 presents the controllers applied to the system and their respective results. The controllers are implemented using MatLab and Simulink on Personal Computers running in a Windows environment. The Real-time Toolbox for MatLab is used to control the system.

8.1 Air-conditioning systems

In general, a heating, ventilating and air conditioning (HVAC) system consists of a primary system, e.g. heat exchangers, pipes or dampers, supplying the medium; hot water, steam or chilled water to the terminal system, i.e. any heating or cooling unit responsible for the conditioning of the space (Levenhagen and Spethmann, 1993). Several types of air-conditioning systems are manufactured. In general they are classified in all-air systems, air-and-water systems and refrigerant-based systems.

Air-conditioning systems are in general 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 change thus 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 is however 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 (Peng, 1996). 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 were 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.

8.2 Description of the fan-coil system

One common type of air-conditioning systems using 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. The system used as pilot plant in this thesis is installed in a test cell at the Delft University of Technology, Department of Mechanical Engineering, and consists of a fan-coil unit inside a test cell (room) under control (van Paassen and Lute, 1993; Peng, 1996). This system is depicted in Fig. 8.1. 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 goal for this system is to keep the temperature

![Diagram of a fan-coil unit system](image)

**Figure 8.1. Air conditioning system**

of the working area in the test cell, $T_{\text{work}}$, at a prescribed reference value, and moreover assuring that enough ventilation and renovated air is supplied to the room. For this purpose three different control actions can be used.

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 renovation 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. 8.1, 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. As discussed in
Section 8.1, 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. 8.1. This procedure leads usually to poor control performance. A different approach is
to build a model relating the measured temperature with the temperature in the working zone
(Peng, 1996). 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 of the mixed-air before the fan $T_m$, which should
be considered in the model. Therefore, a well controlled supply temperature is necessary
to control the temperature at the working area. In this thesis the supply temperature is
controlled over a wide range of temperatures. The control of the working zone can be
directly obtained by applying the linear state-space model presented in Peng (1996). 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 should be seen as the first step for controlling
the air-conditioning system.

The temperature under control, i.e., the supply temperature $T_s$ changes quite fast 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 supply 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.
8.3 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 does not consider the buffer effect presented in the room, and the disturbance introduced by the outside temperature. In order to consider both these effects, the mixed-air temperature before the fan, $T_m$ in Fig. 8.1, 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. Summarizing, the developed fuzzy models have the opening of the heating valve as input ($u(k) \in [0, 1]$, with 0 standing for the valve completely closed) and have two outputs: supply temperature $T_s$ and mixed-air temperature $T_m$. Delayed values of these three variables are also used for the modeling.

In order to apply fuzzy model-based control as described in Chapter 5 and Chapter 6 to the air-conditioning system, two fuzzy models are developed:

1. a Takagi-Sugeno fuzzy model developed using product-clustering techniques, as described in Section 2.4;

2. a TS model which is affine on the input $u(k)$, such that the model is invertible.

The following sections describe the identification procedures and present the developed fuzzy models. The details of the developed fuzzy models are given in Appendix C, such that these models can be used in simulations by those interested to include these results in their own research activities.

8.3.1 TS fuzzy model of the air-conditioning system

This model is constructed from process measurements. The antecedent membership functions and the consequent parameters were estimated from a set of input–output measurements by fuzzy clustering and least-squares methods, as presented in Section 2.4.2. After several tests, the sampling period of 30s was found to be sufficient in order to describe the dynamics of the system (Kuchler, 1997). The identification data set contains $N = 800$ samples, collected in two different day periods (morning and afternoon), using the input signal data shown in Fig. 8.2a. The excitation signal $u$ consists of a multi-sinusoidal signal with five different frequencies and amplitudes and of pulses with random amplitude and width. This
signal is chosen 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 decreasing in order to avoid overheating of the test cell, see Fig. 8.2a. Figure 8.2 presents all the data used to identify the fuzzy models.

(a) Valve opening $u$.  

(b) Mixed-air temperature $T_m$.

(c) Supply temperature $T_s$.

Figure 8.2. Identification data.

<table>
<thead>
<tr>
<th>Table 8.1. Model parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
</tr>
<tr>
<td>$T_s$</td>
</tr>
<tr>
<td>$T_m$</td>
</tr>
</tbody>
</table>

The global model can be divided in two models, one for each of the outputs $T_s$ or $T_m$. All the variables $u$, $T_s$ or $T_m$ are considered in the premises, making these two sub-models
MISO models. Note that a MIMO model can always be decomposed in several MISO models, as discussed in Section 2.1. 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 \( m \), and the considered delays are presented in Table 8.1. For the sake of notational simplicity let \( y_1 = T_s \) and let \( y_2 = T_m \) in the following. The number of clusters was initialized to the number of 10 and is was reduced using the compatible cluster merging technique (Kaymak and Babuška, 1995), yielding the final number of 5 clusters. The orders of the inputs and outputs were 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. 8.3.

**Sub-model for the supply temperature** First, the nonlinear MISO model for the supply temperature is considered. This model is described by the nonlinear function

\[
\hat{y}_1(k + 1) = f_1(y_1(k), y_2(k), u(k), u(k - 1)),
\]

where \( f_1 \) is a nonlinear mapping. The membership functions and the rules describing the model are presented in Appendix C.

**Sub-model for the mixed-air temperature** The nonlinear MISO model for mixed-air supply temperature is described by the nonlinear function

\[
\hat{y}_2(k + 1) = f_2(y_1(k), y_2(k), u(k)),
\]

where \( f_2 \) is again a nonlinear mapping. The complete description of this sub-model is presented in Appendix C.

**Model validation** The complete MIMO model consisting of the two models presented in the previous paragraphs is validated using a separate data set, which was measured on

![Diagram](image)

(a) MISO model for the supply temperature \( T_s \).

(b) MISO model for the mixed-air temperature \( T_m \).

**Figure 8.3.** The two developed MISO TS fuzzy models.
another day. Figure 8.4 compares the supply temperature and the mixed-air temperature of the measured and recursively predicted model outputs in a free run test.

![Graph showing supply and mixed-air temperature over time](image)

**Figure 8.4.** Model validation. Solid line – measured output, dashed line – model output.

Note that both sub-models can follow quite well the real data. 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 $y_m$. Denoting ‘var’ as the variance, the VAF is given by:

$$\text{VAF} = 1 - \frac{\text{var}(y - y_m)}{\text{var}(y)} \times 100\%.$$  

(8.3)

This means that for VAF = 100% the model predicts exactly the real outputs. The VAF’s of the two sub-models for the considered temperatures are given in Table 8.2. 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.

<table>
<thead>
<tr>
<th>Output</th>
<th>VAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_s$</td>
<td>97.21</td>
</tr>
<tr>
<td>$T_m$</td>
<td>99.58</td>
</tr>
</tbody>
</table>

**Table 8.2.** VAF values for the validation of the model.
8.3.2 Affine TS model of the air-conditioning system

Inverse control, as presented in Section 5.1.4, can only be applied if the TS model derived for the system under control is affine in the control input \( u(k) \). The model presented in Section 8.3.1 can be made affine by suppressing \( u(k) \) from the antecedents, and re-identifying the consequent parameters. This procedure was performed for the model presented in Section 8.3.1. However, the model revealed a nonminimum phase behaviour, which hampers the application of inverse control. Thus, a new model where the nonminimum phase behaviour 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 presented in Section 5.1.4, Eq. (5.33). By denoting again \( y_1 = T_s \) and \( y_2 = T_m \), as in Section 8.3.1, the MISO model for the supply temperature is now described by

\[
\dot{y}_1(k+1) = f_1(y_1(k), y_1(k-1), y_2(k), y_2(k-1), u(k)).
\]  

(8.4)

The orders of the inputs and outputs were chosen by comparing several candidate structures of first-order and second-order models in terms of the prediction error criterion, and to keep the several local linear models with a minimum phase structure. The premises in this model do not contain the membership functions for \( u(k) \), and the fuzzy model can be inverted, as presented in Section 5.1.4. The membership functions and the rules describing the model are given in Appendix C.

The identified affine TS fuzzy model is validated by a different data set. Figure 8.5 compares the supply temperature measured and recursively predicted by the model in a free run test. When comparing Fig. 8.5 with Fig. 8.4 it is clear that this model is substantially inferior

![Figure 8.5. Model validation. Solid line – measured output, dashed line – model output.](image)

to the non-affine one presented in Section 8.3.1. The VAF of this model has the value of \( VAF = 89.7 \), which confirms the results observed in Fig. 8.5.
8.4 Controllers applied to the air-conditioning system

In order to compare some of the control approaches presented in this thesis, four different controllers are applied to the air-conditioning system:

- PID control,
- inverse control based on an affine TS fuzzy model,
- predictive control based on classical cost functions,
- predictive control based on fuzzy cost functions.

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 scheme is depicted in Fig. 8.6. 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, $e_m(k) = y_1(k) - \hat{y}_1(k)$, is passed through a first-order low-pass digital Butterworth filter $F_1$. The filtered error $e_{mf}(k)$ is given by:

$$e_{mf}(k) = b_1 e_m(k) + b_2 e_m(k - 1) - a_2 e_{mf}(k - 1),$$

where $b_1 = b_2 = 0.086$, $a_2 = -0.83$. Another first-order low-pass digital Butterworth filter $F_2$ is designed for $y_2$:

$$y_{2f}(k) = d_1 y_2(k) + d_2 y_2(k - 1) - c_2 y_{2f}(k - 1),$$

with $d_1 = d_2 = 0.245$, $c_2 = -0.51$. Note that each filter is based on a single parameter, which determines all the others. The value of that parameter is chosen based on simulations,
in order to reliably filter measurement noises, and to provide fast responses. The mixed-air
temperature is directly fed back to the controller, because the sub-model for this temperature
is very good, see Table 8.2, 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 (te Braake, 1997).
The four different controllers and their respective results are presented in the following
sections.

### 8.4.1 PID control of the air-conditioning system

A well-known proportional-integral-derivative (PID) controller was applied to the system
(Åström and Hågglund, 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 if possible (Sousa, et al., 1997d). Figure 8.7 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 presents undesired
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 can not 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. The response
using the PID controller confirms the high nonlinear character of this system.

![Figure 8.7](image)

**Figure 8.7.** Real-time response of a PID controller. The solid line is the
measured output, the dashed line is the reference.
8.4.2 Inverse control based on an affine TS fuzzy model

The model presented in Section 8.3.2 can be inverted using the inversion method for affine TS fuzzy models, presented in Section 5.1.4. This inverted model is used as the controller, in the control scheme shown in Fig. 8.6. The control structure is applied in real-time control, and the results are depicted in Fig. 8.8. Note that a slowly varying reference must be chosen. In fact, faster changes in 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. As the affine TS fuzzy model used is considerably worse than the other models developed for this system (see Section 8.3.2 and Section 8.3.1) this is the main cause for the compulsory slow behaviour of this control system. Note that when the reference is chosen as in Section 8.4.1, the results using this controller are similar to the ones obtained using the PID controller.

![Temperature](image1)

**Figure 8.8.** 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.

8.4.3 Predictive control based on classical cost functions

Given the results obtained 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 Section 8.3.2 revealed good VAF values for both modeled temperatures. This model is thus suitable to be used in a predictive control scheme, as presented in Section 4.1. Note that this TS fuzzy model is nonlinear,
requiring a nonconvex optimization technique to find the best control action, which is applied to the system at each sampling instant. The results presented in Section 7.1.2 strongly suggest the use of the branch-and-bound algorithm applied to predictive control, as described in Section 7.1. The main problem is the discretization of the control actions $u(k)$, because the B&B method requires a finite, and preferably small, number of possible control actions. Another point that can constitute a disadvantage is that predictive control is directly applied without using the control scheme combining inverse control and predictive control as presented in Section 5.2. Therefore, the discretization must be carefully chosen in order to avoid the chattering effect due to the large discretizations. 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, and are the following at each time step $k + i$, $i = 1, \ldots, H_c$:

$$
\Delta u(k + i - 1) : \quad \Omega = [-0.05 \quad -0.02 \quad -0.01 \quad 0 \quad 0.01 \quad 0.02 \quad 0.05].
$$ (8.7)

This choice introduces a rate constraint of $\Delta u(k + i) \leq 0.05$, which does not alter significantly the performance of the system, but it smoothes the control actions avoiding undesired 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 8.3, and are chosen according to the general guidelines given in Section 6.3.1 in the paragraph describing classical objective functions.

<table>
<thead>
<tr>
<th>$H_c$</th>
<th>$H_p$</th>
<th>$\alpha_i$</th>
<th>$\beta_i$</th>
<th>$\gamma_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>500</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 8.3. Parameters of the classical predictive controller

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 $u(k)$. 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 1s, which is more than sufficient to the real-time application (note that the sampling period is $T_s = 30$s). The objective function of this controller is a particular case of the Eq. (6.26), and stated as follows:

$$
J = \sum_{i=3}^{4} (\hat{e}(k + i))^2 + \sum_{i=2}^{4} 500 (\Delta u(k + i - 1))^2 + \sum_{i=3}^{4} 50 (\Delta \hat{y}(k + i))^2.
$$ (8.8)

The parameters $\alpha_i$, $\beta_i$ and $\gamma_i$, as well as the parameters $m_1$, $n_1$, $m_2$, $n_2$, $m_3$, $n_3$ in Eq. (6.26), 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 $\Delta u$ is considered in (8.8), which introduces a smooth constraint in this variable at time step $k + 1$. Then, the first change in control action $\Delta u(k)$ can vary freely in the interval of discretized control actions $\Omega$ 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 showed 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. 8.9. The overshoots presented 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.

![Graph](image)

**Figure 8.9.** 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

$$r_f(k) = 0.086 \, r(k) + 0.086 \, r(k - 1) - 0.83 \, r_f(k - 1).$$  \hspace{1cm} (8.9)

The design of this filter follows the guidelines presented for the other linear filters presented previously in this section. The results obtained using the shaped reference are presented
8.4 Controllers applied to the air-conditioning system

![Graphs showing supply temperature and valve position over time](image)

**Figure 8.10.** 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.

In Fig. 8.10. Note that both overshoots for times $t = 15 \text{ min}$ and $t = 45 \text{ min}$ 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 $t = 15 \text{ min}$. For this step, the rise time as defined in (3.9), and for $\lambda = 0.8$ has the value of $\phi_{\text{rise}} = 1 \text{ min}$ for the controller without the shaped reference, while the controller with the shaped reference has the rise time of $\phi_{\text{rise}} = 14 \text{ min}$; it is thus 14 times slower. The settling time as defined in (3.10) with $\epsilon = 0.05$ is $\phi_{\text{set}} = 15 \text{ min}$ for the controller with the shaped reference, and $\phi_{\text{set}} = 4 \text{ min}$ for the controller without the filtered reference. Also here, the result is much better, and the controller achieves the steady-state much faster. A compromise between small overshoot and reasonable 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 even desirable to increase human comfort. Imagine that a person enters a very cold room. Comfort is given by feeling the temperature increasing as fast as possible, disregarding the fact that this action results in the introduction of 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 present overshoots, the shaped reference is still introduced in this thesis in order to generalize the obtained results for other types of systems.
8.4.4 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, see Fig. 8.9. The overshoot is due to the introduction of the filter $P_t$ in the IMC control scheme presented in Fig. 8.6. Predictive control using fuzzy objective functions is applied in this section. Fuzzy criteria as described in Section 6.3 are utilized. The branch-and-bound method for predictive control with fuzzy decision functions introduced in Section 7.4 is applied in order to find the discrete optimal control actions. The changes in control actions $\Omega$, necessary to perform the optimization, are the same as the ones in (8.7), used in Section 8.4.3, allowing for the comparison of the control results. The control and prediction horizons are the same as in Section 8.4.3, i.e. $H_c = 2$ and $H_p = 4$. These values revealed to be suitable to control the system, and allows for a proper comparison of the fuzzy objective function to the classical one.

![Membership functions](image)

**Figure 8.11.** 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, $\mu_e$, $\mu_y$ and $\mu_u$ respectively, are given in Fig. 8.11. Note that these membership functions are the same as in Fig. 6.7, which are defined for a general system. The parameters $K_v^+, K_v^-, K_y^+, K_y^-, S_y^+, S_y^-, K_u^+, K_u^-$, $H_u^+$ and $H_u^-$, described in Section 6.3.1 are given in Table 8.4. The values for these parameters are also presented in Fig. 8.11. These values are chosen based on the considerations made for general systems presented in Section 6.3. The discussion presented in Section 6.3.3 for the paragraph on the air-conditioning system is generally valid here. However, small adjustments of some parameters for the air-conditioning in this section are required.

<table>
<thead>
<tr>
<th>$K_v^+$</th>
<th>$K_v^-$</th>
<th>$K_y^+$</th>
<th>$K_y^-$</th>
<th>$S_y^+$</th>
<th>$S_y^-$</th>
<th>$K_u^+$</th>
<th>$K_u^-$</th>
<th>$H_u^+$</th>
<th>$H_u^-$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>-30</td>
<td>1.5</td>
<td>-1.5</td>
<td>0.3</td>
<td>-0.3</td>
<td>1</td>
<td>-1</td>
<td>0.05</td>
<td>-0.05</td>
</tr>
</tbody>
</table>
The fuzzy goals are combined using the Yager $t$-norm in the following way:

$$
\mu_{\bar{y}} = \sum_{i=3}^{4} (\mu_{\bar{e}}(\hat{e}(k+i)))^{w_{Y}} + \sum_{i=2}^{2} (\mu_{\bar{u}}(\Delta u(k+i)))^{w_{Y}} + \sum_{i=3}^{4} (\mu_{\bar{y}}(\Delta \hat{y}(k+i)))^{w_{Y}}
$$

$$
\mu_{u} = \max(0, 1 - \mu_{\bar{y}}^{1/w_{Y}}), \quad w_{Y} > 0.
$$

(8.10)

The parameter for this $t$-norm is chosen as $w_{Y} = 2$, which allows for a good balance between fast response and small overshoots, see Section 6.4. The results obtained with this controller are presented in Fig. 8.12.

![Figure 8.12. Real-time response using predictive control with fuzzy objective functions. The solid line is the measured output, the dashed line is the reference.](image)

Comparing the response of the system to the one presented in Fig. 8.9, they are quite similar. However, the predictive controller with fuzzy objective functions can reduce the overshoot at time $t = 15 \text{ min}$, from a value of 34% to 20%. The other significant overshoot for time $t = 45 \text{ min}$ 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 presented in Fig. 8.10. 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 Section 4.3 revealed able to cope with this problem, but it introduces undesired oscillations.

8.5 Concluding remarks

Air-conditioning systems are widely used in different types of infrastructures like buildings, vehicles, etc. These systems require improved controllers, demanding human comfort and energy saving. Further, these systems have thus quite general and fuzzy goals, which can be translated to predictive control using fuzzy decision functions.

This chapter presents low level control as the first step of applying MBPC with fuzzy objective functions to air-conditioning systems. A test cell located at the Delft University of Technology, Department of Mechanical Engineering, described in van Paassen and Lute (1993) is used for real-time control. Four controllers are applied to the system: PID control, inverse control based on an affine TS fuzzy model, predictive control based on classical objective functions and predictive control based on fuzzy decision functions. The first two were able to control the system when the reference presents very slow changes. In general, both predictive controllers can cope with references changing rapidly, and presenting good control performances. 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 suppress these overshoots.
Conclusions

This final chapter presents general conclusions on the research developed in this thesis. Specific conclusions on the different aspects of fuzzy model-based control can be found at the end of each chapter. Suggestions for future research are also included.

9.1 Fuzzy model-based control

Fuzzy model-based control can have different meanings. In this thesis, fuzzy model-based control is defined as the model-based control systems that utilize at least one of the three fuzzy components:

1. fuzzy model,
2. fuzzy design specifications,
3. fuzzy controller.

Fuzzy modeling and identification is used as a tool in this thesis and no research has been performed on this subject. A brief discussion on modeling techniques is presented in order to clarify the importance of models and model-based control. Fuzzy models can give a transparent description of a system when white-box models are not available, or are too complex to be used for control purposes. When no previous knowledge is available, fuzzy models can be directly extracted from data using product-space fuzzy clustering techniques. The type of fuzzy models used in this thesis, i.e., singleton models and TS models are easily interpretable, which is important for understanding the system and defining proper (fuzzy) design specifications.
The design goals for control design are expressed by design specifications. In the classical control theory, the rigorous definition of design specifications for linear systems, results directly in performance criteria. The combination of these criteria is usually defined as an optimal control problem, where a trade-off between the different criteria must be performed. The most important performance specifications are revisited briefly. Fuzzy design specifications can translate informal design goals, which translates directly the designers' wishes. These specifications are transformed in performance criteria by using fuzzy sets describing the fuzzy goals and the fuzzy constraints. These goals and constraints can be combined in a fuzzy decision making approach to derive the desired control actions.

From the several types of model-based controllers, model-based predictive controllers have shown, in general, an increased control performance. Using a model of the system, these type of controllers compute a sequence of control actions by minimizing a given objective function. MBPC can control open-loop unstable systems, systems with nonminimum phase behaviour, and/or systems subjected to constraints. The first control action found after the optimization is applied to the process, the new output is read and the optimization is repeated at the next time sample. This is called the receding horizon principle, and it is implemented in order to cope with model-plant mismatch and disturbances. In practice, this procedure is sometimes insufficient and other methods can be used to diminish the effect of these phenomena. In this thesis, two different techniques are used for this purpose: internal model control and fuzzy compensation.

Internal model control uses a model in parallel with the system, and the difference between the output of the model and the output of the system is added to the reference. The output of the model is used to derive the control actions, and the rest of the control scheme cope with the possible model-plant mismatches. In order to stabilize or increase the robustness of this scheme, a linear filter is usually included to attenuate the effect of model-plant mismatches and disturbances. However, this linear filter deteriorates the closed-loop dynamics, especially when the system under control is nonlinear. A different solution to reduce the effect of model-plant mismatch, called fuzzy compensation, is proposed in this thesis.

The main goal of fuzzy compensation is to compensate for steady-state errors in the closed-loop system. The compensation is based on the derivative of the model output at the corresponding state of the system. This term acts like a variable integral gain for nonlinear systems. A fuzzy set is defined for the error between the reference and the output in order to determine the degree of activation of the fuzzy compensator. This scheme resulted in good results, but is quite sensitive to noise, and its application can not be recommended at this stage.

9.2 Predictive control using fuzzy models

Given the state-of-the-art in fuzzy modeling, the most straightforward approach to fuzzy model-based control is to use fuzzy models in this type of control. Among these approaches,
the most direct one is to invert the derived fuzzy model and use this inversion as the controller directly. In theory, a model is invertible if the system under control is stable and minimum-phase. Several inversion methods are suggested in literature, but the most straightforward way is to invert the model analytically providing fast inversion. Special types of fuzzy models can be inverted in this way. Algorithms for inverting singleton fuzzy models and affine TS fuzzy models are presented in this thesis.

Singleton fuzzy models using the weighted-mean defuzzification, having a partition as antecedent membership functions and being monotonous for the complete domain, can be inverted analytically. The inverted model can be directly used as the controller of the system.

The Takagi-Sugeno fuzzy model, affine on the control action can also be inverted analytically. By suppressing the control action from the antecedent variables, but keeping its influence in the consequents, the inversion is easily obtained. The affine TS model can be identified from a non-affine one by not considering the control action in the antecedents, and recompute the consequents using the least squares algorithm.

Both singleton and TS models can be adapted on-line in the presence of time-varying systems. The model inversion can not be used directly to control the system, because it can not cope with model-plant mismatches, disturbances and constraints. Including the inverted model as controller in the internal model control scheme can diminish significantly the effect of model-plant mismatches. In the presence of constraints, the inversion is combined with predictive control. A supervisory scheme combining inversion control when the constraints are not violated, and predictive control otherwise, is presented.

In the presence of nonlinear models, the optimization problem to be solved using the MBPC control is usually nonconvex. This type of optimization is difficult to solve and time consuming. Therefore, it is advantageous to use the inversion technique whenever possible to reduce the computational time. Note that near to steady-states the constraints are normally not violated, and the inversion can be used. The MBPC problem is solved using the branch-and-bound algorithm, by discretizing the control actions.

### 9.3 Predictive control with fuzzy objective functions

Predictive control has some advantages over other control techniques when a model of the process is available. MBPC can be applied to complex processes, e.g. multivariable or nonlinear plants, and can efficiently deal with constraints. Most of the objective functions used in MBPC utilize the sum squared error and a quadratic term of the control actions or its change. In the presence of linear or linearized models of the system this quadratic form leads to a quadratic problem, for which fast algorithms exist. However, this quadratic form does not translate always to the real control goals. Moreover, if the system is nonlinear, the advantages of having a quadratic problem disappear. Fuzzy sets can be used to defined the control goals, introducing extra flexibility in the control system. Moreover, the constraints can be also fuzzified. This approach was first introduced for the field of decision making.
In fuzzy decision making, fuzzy goals, fuzzy constraints and the fuzzy decision are defined on a set of alternatives, which can be continuous or discrete. Multiple goals and multiple constraints can be defined in the same set, or in different sets. This last problem becomes quite complex. This thesis introduces a formal description for fuzzy goals and fuzzy constraints, where each fuzzy criterion can be defined in a different set. Using cylindrical extensions, all the criteria can be extended to the same multidimensional set, which allows for the use of aggregation operators between the different fuzzy criteria. In this thesis three different aggregation operators are considered: aggregation between goals, between constraints and between a goal and a constraint. This is not the most general approach where more aggregations can be considered, but it is general enough for control applications, and the introduction of unnecessary degrees of freedom is avoided. The multidimensional fuzzy decision making problem revealed to be very complex. A common solution is to discretize the set of possible solutions and search for the optimal solution in this set.

The most general case in fuzzy decision making is where the process is multistage, which is similar to the predictive control problem. The extension of the fuzzy decision making approach to the multistage case requires the definition of fuzzy criteria for each time step. In control, the set of alternatives becomes a policy containing the set of possible control actions. The aggregation of fuzzy criteria is thus a straightforward extension of the aggregation for the multidimensional fuzzy decision making problem, where the same three aggregation operators are considered. Note that the aggregation between a goal and a constraint can now be made at different time steps. Fuzzy decision making for control can be directly translated to the model-based predictive control environment, where the objective function is defined as a confluence of fuzzy goals and constraints. Two main problems emerge in this approach: the proper choice of the fuzzy goals and fuzzy constraints, and the choice of aggregation operators to combine these criteria.

A classical objective function containing the squared terms for the errors, the change in the control actions, and the change in the outputs is generalized by defining membership functions for these variables. In the general case, the fuzzy sets can also be defined for other variables and can be of all shapes. However, taking account for the usual requirements in a control system, i.e., fast response with no overshoot, some almost ‘standard’ membership functions can be selected, facilitating the control design. Moreover, in general the difference between goal and constraint vanishes in the decision making approach, but this thesis presents fuzzy constraints in a different way, such that crisp constraints can also be included. Therefore, the extension of the classical objective function including hard constraints can also be included in the formulation of MBPC using fuzzy decision making. Two simulated examples of two different systems show the advantages of using fuzzy criteria, where it is clear that the control system deliver more rigorous control actions, resulting in better control performance. The examples present also some general empirical rules on how to tune the various parameters in the general membership functions.

As important as the definition of fuzzy goals and constraints, is the choice of the proper aggregation operators to combine them. The most straightforward approach in order to satisfy all the goals and all the constraints is to use a $t$-norm operator. Moreover, the use of
one of these operators is compulsory in order to not violate the crisp constraints. The most used $t$-norm is the minimum operator, but it does not allow for any tradeoff between different goals and constraints. Moreover, it can not be applied for nonminimum phase systems. Weighted operators are not used in this thesis because they introduce extra unnecessary parameters. In control applications, the effects of these weights can be performed by using parametric norms. Parametric $t$-norms depend on only one parameter, and generalize a large number of $t$-norms. This parameter defines the degree of compensation between the different criteria, and it can be used to tune the desired response allowing for faster or slower responses, with or without overshoot. Other parametric operators can be used beyond the $t$-norms. However, the use of these aggregation operators can violate some crisp constraints. This fact most always be taking into account when this type of parametric operators is being used. The optimization problem remains nonconvex and algorithms to solve this type of problems must be used.

9.4 Optimization problems

The solution of the optimization problem is of vital importance in any type of predictive control, especially if the problem is nonconvex. When the search for a solution costs too much time, the control algorithm can not be applied in real-time, and it is thus useless. A compromise between a reasonably good solution, as close as possible to the optimal, and found within the required sampling period must be obtained.

9.4.1 Optimization algorithms for classical MBPC

When the optimization problem is nonconvex due to the nonlinearity of the model used in MBPC, a possible solution is to discretize the possible control actions and search for the best solution in this space. If the discretization is fine enough, the optimal discrete solution is very close to the continuous optimal solution.

The branch-and-bound method is a tree search method to find optimal discrete solutions. It is based on the estimation of lower and upper bounds, selecting the branches to be followed. Good estimates of these bounds are necessary to perform the search efficiently. The B&B algorithm proposed in this thesis performs the initialization step, and revealed to be faster than enumerative search and sequential quadratic programming for the same optimization problem. Note that SQP finds the continuous optimum. However, SQP converges frequently to local minima, while B&B finds the global discrete minimum. Thus, in general better performance can be obtained when B&B is used. The B&B algorithm grows exponentially with respect to the number of levels (control horizon) and the number of discretizations. Therefore, these parameters should be chosen as low as possible, but still guaranteeing good control performance.
A different discrete method is the randomized search known as genetic algorithms. This thesis presents a genetic algorithm developed to the optimization problems found in MBPC. Special termination conditions, a coding scheme, a method to initialize the population and a learning feature are introduced to solve this particular problem. Note that this method grows linearly with the number of discretizations, contrary to B&B that grows exponentially. Therefore, genetic algorithms revealed to be suitable for MBPC problems with large control horizons and several inputs, where they can outperform the branch-and-bound method.

9.4.2 Optimization in MBPC with fuzzy objective functions

In the presence of fuzzy objective functions the optimization problem in MBPC is in general nonconvex. However, under certain conditions it is possible to obtain a convex problem, which can be solved in polynomial time. Thus, when the model of the system is linear, the fuzzy goals are given by the minimization of the errors between the references and the outputs, the constraints are crisp and convex, and the aggregation of fuzzy goals is performed by using the Yager \( t \)-norm, then the optimization problem at each step is convex. Algorithms using descent methods, or other algorithms can be used for this type of problems. Note that the conditions to obtain an optimization problem that is convex are quite strict, and in general a nonconvex problem remains to be solved.

A possible solution for nonconvex problems is to extend the branch-and-bound algorithm for MBPC with fuzzy objective functions. Given a certain \( t \)-norm to aggregate the fuzzy criteria, its respective dual \( s \)-norm and the complement, the optimization problem can be solved using the B&B algorithm. As the optimization problem in multistage fuzzy decision making is a maximization problem, the optimization problem is transformed into a minimization problem by using the fuzzy complement. All the considerations made for the branch-and-bound algorithm applied to MBPC with classical objective functions remain valid for the extended B&B method. Again, the critical parameters are the control horizon and the number of discretizations chosen for the control action. Several applications show that significant computer effort is saved using B&B instead of enumerative search. This method can thus be applied to faster systems, i.e., systems with smaller sampling periods.

9.5 Suggestions for future research

The topics concerning fuzzy model-based control presented in this thesis are not fully explored. Given the obtained results and the problems to be solved, the most promising future research concerns the following topics.

Hierarchical fuzzy goals in control applications. In this thesis fuzzy goals and fuzzy constraints are defined at the same level, and thus no hierarchy is assumed. 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. Although not advised in this thesis, it is also mentioned that they should 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 in sub fuzzy goals, creating a clear hierarchy. The problem can be tackled at different levels, depending on the focus of the control system at a particular moment.

**Adaptive decision alternatives for the B&B method.** The discretization of the branch-and-bound algorithm introduces a trade-off between the number of discrete actions and the control performance. In order to be solved in small computational time, the application of B&B requires a small number of discrete control alternatives. However, the control actions are usually not restricted to a finite number of possible values, and this discretization leads to an approximate solution. This approximation can generate oscillations around non-varying references and slow step responses depending on the selected set of discrete control actions. A possible solution to this problem is to use an adaptive set of discrete control alternatives, based on the fulfillment of some fuzzy criteria. Let the number of control alternatives remains the same and low. If a scaling factor is multiplied by this interval, an adaptation can be performed. Finding the proper (fuzzy) criteria for adapting this scaling factor is the main problem to be solved in this approach. The adaptation factor can depend, for instance, on the error between the reference and the output and on its change. Adaptive decision alternatives guarantee that the number of alternatives can be kept low. The effects on the performance of the system and the choice of the proper fuzzy criteria to define the scaling must be investigated.

**B&B for MIMO systems.** The branch-and-bound algorithms presented in this thesis 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 the adaptive decision alternatives, as described in the previous paragraph. Another possibility is to use a smaller number of discretizations, find a suboptimal solution, and to use this solution as an initialization of a finer discretization afterwards. This method does not guarantee that the solution found is the global optimum, but this solution is in general a good solution. Note that the finer discretization can be used several times. This method require some research 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 must also be tested.
Fuzzy compensation. The fuzzy compensation method presented in this thesis suffers from lack of robustness because it introduces oscillations in some situations. A possible cause of this oscillations is the fuzzy set defined to determine whether the system is in steady-state or not. In fact, sometimes the fuzzy compensator assumes that the system is in steady-state, when this is not the case, and the compensation action is stronger than it should be. Therefore, a new method to determine whether the system is in steady-state must be introduced. A possibility is to consider the error and the change in the error to define the steady-state criteria. This definition can not be given by a simple membership value as before, and a combination of both criteria, must be used. How to combine these criteria is the problem to be solved.
Fuzzy sets – basic definitions

This appendix gives the basic concepts of fuzzy set theory which are used in this thesis. A detailed discussion of fuzzy set theory can be found, for instance, in (Klir and Yuan, 1995; Zimmermann, 1996; Yager and Filev, 1994). The definitions presented follow these authors.

A.1 Basic definitions

Definition A.1.1 (fuzzy set) Assume that \( X \) is a set, which is the universe of discourse. A fuzzy set \( A \) on the universe \( X \) is associated with a membership function \( \mu_A(x) \) which is a mapping from the universe \( X \) into the unit interval, i.e.,

\[
\mu_A(x): X \rightarrow [0, 1].
\]  

Therefore, for each \( x \in X \), \( \mu_A(x) \) indicates the degree to which \( x \) is a member of the set \( A \). This membership degree ranges from 0, where \( x \) does not belong to the set, up to 1, where \( x \) completely belongs to the fuzzy set \( A \).

When \( X \) is a discrete set, the membership grade must be explicitly expressed. For this situation, the fuzzy set \( A \) is defined by a list of ordered pairs, each one containing the membership degree and the element considered. For continuous universe of discourses \( X \), the membership function can be expressed by a function. This thesis uses the following forms of membership functions:
• *Trapezoidal* membership functions:

\[
\mu(x) = \max \left( 0, \min\left( \frac{x - a}{b - a}, 1, \frac{d - x}{d - c} \right) \right),
\]

(A.2)

where \(a, b, c\) and \(d\) are coordinates of the trapezoid apexes. A *triangular* membership function is obtained for \(b = c\).

• *Exponential* membership functions:

\[
\mu(x) = \begin{cases} 
\exp \left( -\frac{x}{a^-} \right), & -\infty < x < 0; \\
\exp \left( -\frac{x}{a^+} \right), & 0 \leq x < \infty.
\end{cases}
\]

(A.3)

where \(a^-\) and \(a^+\) define the tangents of each exponential curve in the left and right parts of the membership functions, respectively.

**Definition A.1.2 (support)** The support of a fuzzy set \(A\), denoted \(\text{supp}(A)\), is the crisp subset of \(X\) that contains all the elements of \(X\) that have nonzero membership grades in \(A\):

\[
\text{supp}(A) = \{ x \mid \mu_A(x) > 0 \}.
\]

(A.4)

**Definition A.1.3 (core)** The core of a fuzzy set \(A\), denoted \(\text{core}(A)\), is a crisp subset of \(X\) consisting of all elements with membership grade one:

\[
\text{core}(A) = \{ x \mid \mu_A(x) = 1 \}.
\]

(A.5)

**Definition A.1.4 (height)** The height of a fuzzy set, denoted \(\text{hgt}(A)\), \(A\) is the largest membership grade obtained by any element in \(A\):

\[
\text{hgt}(A) = \max_{x \in X}(\mu_A(x)).
\]

(A.6)

**Definition A.1.5 (normal fuzzy set)** A fuzzy set \(A\) is normal if \(\text{hgt}(A) = 1\), i.e., it exists at least one element \(x \in X\) such that \(\mu_A(x) = 1\).

### A.2 Operations on fuzzy sets

The classic set theoretic operations are extended to fuzzy sets. There exist various ways to extend these operators. General definitions are presented in this section, which are used in the thesis.
Definition A.2.1 (Complement of a fuzzy set) The complement $c$ is defined by the function:

$$c : [0, 1] \to [0, 1],$$  \hfill (A.7)

and this function must satisfy at least the following two requirements:

1. $c(0) = 1$ and $c(1) = 0$, (Boundary conditions),

2. $\forall a, b \in [0, 1]$, if $a \leq b$ then $c(a) \geq c(b)$, (Monotonicity).

The most used complement has the form:

$$\mu_A(x) = 1 - \mu_A(x).$$  \hfill (A.8)

The intersection of fuzzy sets can be defined by a class of operators fulfilling certain conditions. These operators are known as triangular or $t$-norms.

Definition A.2.2 (Fuzzy intersection: $t$-norm) A $t$-norm $t$ is a binary operation:

$$t : [0, 1] \times [0, 1] \to [0, 1],$$  \hfill (A.9)

that satisfies the following conditions for all $a, b, c \in [0, 1]$:

1. $t(a, 1) = a$,

2. $b \leq c$ implies $t(a, b) \leq t(a, c)$, (Monotonicity),

3. $t(a, b) = t(b, a)$, (Commutativity),

4. $t(a, t(b, c)) = t(t(a, b), c)$, (Associativity).

Some frequently used $t$-norms are:

- Zadeh's intersection: $t(a, b) = \min(a, b)$
- algebraic product: $t(a, b) = ab$
- bounded difference: $t(a, b) = \max(0, a + b - 1)$
- Yager's intersection: $t(a, b) = \max\left\{0, 1 - [(1 - a)^{w_Y} + (1 - b)^{w_Y}]^{1/w_Y}\right\}$, $w_Y > 0$

The fuzzy union operation is generalized by a $t$-conorm or $s$-norm operator.

Definition A.2.3 (Fuzzy union: $s$-norm) A $s$-norm $s$ is a binary operation:

$$s : [0, 1] \times [0, 1] \to [0, 1],$$  \hfill (A.10)

that satisfies the following conditions for all $a, b, c \in [0, 1]$:
1. \( s(a, 0) = a, \)

2. \( b \leq c \) implies \( s(a, b) \leq s(a, c) \), (Monotonicity),

3. \( s(a, b) = s(b, a) \), (Commutativity),

4. \( s(a, s(b, c)) = s(s(a, b), c) \), (Associativity).

Some frequently used \( t \)-conorms are:

- standard union: \( s(a, b) = \max(a, b) \)
- algebraic sum: \( s(a, b) = a + b - ab \)
- bounded sum: \( s(a, b) = \min(1, a + b) \)
- Yager’s union: \( s(a, b) = \min \left[ 1, (a^{w_Y} + b^{w_Y})^{1/w_Y} \right], \quad w_Y > 0 \)

### A.3 Fuzzy relations

**Definition A.3.1 (Fuzzy relation)** An \( n \)-ary fuzzy relation is a mapping

\[
\mathcal{R}: X_1 \times X_2 \times \cdots \times X_n \rightarrow [0, 1],
\]

(A.11)

which assigns membership grades to all \( n \)-tuples \((x_1, x_2, \ldots, x_n)\) from the Cartesian product \( X_1 \times X_2 \times \cdots \times X_n. \)

Fuzzy relation is then a fuzzy set over the Cartesian product \( X_1 \times X_2 \times \cdots \times X_n. \) A membership grade \( \mu_\mathcal{R}(x_1, x_2, \ldots, x_n) \) represents the degree of association among the elements of the different domains \( X_i. \)

### A.4 Projections and cylindrical extensions

Let \( X^i \) denote the Cartesian product \( X_1 \times X_2 \times \cdots \times X_i, \) with \( i \in \mathbb{N}. \) Let also \( i = (x_1, x_2, \ldots, x_i). \)

**Definition A.4.1 (Projection of a fuzzy set)** Let \( \mathcal{R}(x_1, x_2, \ldots, x_i) \) be a relation in the universe \( X^i. \) Let also \( X^j \) be a universe with \( j < i. \) Each point \( j = (x_1, x_2, \ldots, x_j) \) is a subsequence of \( i, \) and is denoted by \( j \prec i. \) The projection of \( \mathcal{R} \) on \( X^j \) is denoted by \( \text{proj}_{X^j} \) and is defined by:

\[
\text{proj}_{X^j}(j) = \max_{j \prec i} \mathcal{R}(i)
\]

(A.12)
Definition A.4.2 (Cylindrical extension of a fuzzy set) Let $\mathcal{R}(x_1, x_2, \ldots, x_j)$ be a relation in the universe $X^j$. Let also $X^i$ be a universe with $j < i$. The cylindrical extension of $\mathcal{R}$ on $X^i$ is denoted by $\text{cext}_{X^i}^{X^j}$ and is defined by:

$$\text{cext}_{X^i}^{X^j}(i) = \mathcal{R}(j)$$ (A.13)

for each $i$ such that $j < i$.

Note that the cylindrical extension is the inverse of the projection. An example of a cylindrical extension is presented in Fig. A.1.

![Diagram of cylindrical extension](image)

**Figure A.1.** Example of a cylindrical extension from $\mathbb{R}$ to $\mathbb{R}^2$. 
Gustafson-Kessel algorithm

Cluster analyses classifies objects according to similarities among them. These techniques can be applied to data. In system identification, clustering finds relationships between the system variables.

The data set $Z$ to be clustered is formed as in (2.25). The columns of $Z$ are denoted by $\{z_{t}, \ell = 1, \ldots, N_{d}\}$. Let $U = [\mu_{\ell}] \in [0, 1]^{(K \times N_{d})}$ denote a fuzzy partition matrix of $Z$. Let $V$ be a vector of cluster prototypes (centers) to be determined, defined by $V = [v_{1}, v_{2}, \ldots, v_{K}]$, and let $F$ be a set of cluster covariance matrices $F = [F_{1}, \ldots, F_{K}]$, where $F_{i}$ are positive definite matrices in $\mathbb{R}^{(n+1) \times (n+1)}$. The GK algorithm searches for an optimal fuzzy partition $U$ and the vector of cluster prototypes $V$ by minimizing the following objective function:

$$J(Z, U, V) = \sum_{i=1}^{K} \sum_{t=1}^{N_{d}} (\mu_{\ell})^{m} d_{i\ell}^{2}, \quad \text{(B.1)}$$

where $m$ is a weighting parameter. The function $d_{i\ell}$ is the distance of a data point to the cluster prototype induced by a positive definite matrix $M_{\ell}$:

$$d_{i\ell}^{2} = (z_{t} - v_{i}^{(l)})^T M_{\ell} (z_{t} - v_{i}^{(l)}). \quad \text{(B.2)}$$

The GK algorithm can be summarized as follows.

Given the data set $Z$, choose the number of fuzzy rules (clusters) $1 < K \ll N$, the weighting exponent $m > 1$ and the termination tolerance $\epsilon > 0$. Initialize the partition matrix randomly.
Repeat for $i = 1, 2, \ldots$

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

\[ \mathbf{v}_i^{(l)} = \frac{\sum_{\ell=1}^{N_d} (\mu_{\ell i}^{(l-1)})^m \mathbf{z}_\ell}{\sum_{\ell=1}^{N_d} (\mu_{\ell i}^{(l-1)})^m}, \quad 1 \leq i \leq K. \]

**Step 2:** Compute covariance matrices:

\[ \mathbf{F}_i = \frac{\sum_{\ell=1}^{N_d} (\mu_{\ell i}^{(l-1)})^m (\mathbf{z}_\ell - \mathbf{v}_i^{(l)})(\mathbf{z}_\ell - \mathbf{v}_i^{(l)})^T}{\sum_{\ell=1}^{N_d} (\mu_{\ell i}^{(l-1)})^m}, \quad 1 \leq i \leq K. \]

**Step 3:** Compute distances:

\[ M_i = \text{det}(\mathbf{F}_i)^{-1/2} \mathbf{F}_i^{-1}, \]

\[ d_{i\ell}^2 = (\mathbf{z}_\ell - \mathbf{v}_i^{(l)})^T M_i (\mathbf{z}_\ell - \mathbf{v}_i^{(l)}). \]

**Step 4:** Update partition matrix:

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

\[ \mu_{i\ell}^{(l)} = \frac{1}{\sum_{j=1}^{K} (d_{i\ell}/d_{j\ell})^{2/(m-1)}}, \]

otherwise

\[ \mu_{i\ell}^{(l)} = 0 \quad \text{if} \quad d_{i\ell} > 0, \quad \text{and} \quad \mu_{i\ell}^{(l)} \in [0, 1] \]

with \[ \sum_{i=1}^{K} \mu_{i\ell}^{(l)} = 1. \]

until $\| \mathbf{U}^{(l)} - \mathbf{U}^{(l-1)} \| < \epsilon$. 
Fuzzy models of the air-conditioning system

C.1 Takagi-Sugeno fuzzy model

The general TS model is divided in two MISO sub-models, see Fig. 8.3: a model for the supply temperature and a model for the mixed-air temperature.

C.1.1 Sub-model for the supply temperature

This model is described by the nonlinear function

\[ \hat{y}_1(k + 1) = f_1(y_1(k), y_2(k), u(k), u(k - 1)), \quad (C.1) \]

where \( f_1 \) is a nonlinear mapping. The membership functions for the antecedents \( y_1(k), y_2(k), u(k) \) and \( u(k - 1) \) are obtained from the projections of the multidimensional fuzzy sets derived from the clustering technique onto the antecedent variables, and are shown in Fig. C.1. Note that no linguistic labels are attributed to the fuzzy sets. As the obtained fuzzy model is not subjected to validation by experts, this step is not compulsory. Thus, the membership functions are described by \( A_{ij} \), where \( i = 1, \ldots, K \) stands for the number of rules, and \( j \) stands for the different antecedents of the fuzzy rules. The fuzzy rules of this
Figure C.1. Membership functions for the TS model with the supply temperature as output.

TS model are given by:

1. If $y_1(k)$ is $A_{11}$ and $y_2(k)$ is $A_{12}$ and $u(k)$ is $A_{13}$ and $u(k-1)$ is $A_{14}$ then
   
   $y_1(k+1) = 0.746y_1(k) + 2.99 \cdot 10^{-2}y_2(k) + 5.83u(k) + 17.3u(k-1) - 0.236$

2. If $y_1(k)$ is $A_{21}$ and $y_2(k)$ is $A_{22}$ and $u(k)$ is $A_{23}$ and $u(k-1)$ is $A_{24}$ then
   
   $y_1(k+1) = 0.804y_1(k) + 4.38 \cdot 10^{-2}y_2(k) + 1.72u(k) + 5.98u(k-1) + 5.10$

3. If $y_1(k)$ is $A_{31}$ and $y_2(k)$ is $A_{32}$ and $u(k)$ is $A_{33}$ and $u(k-1)$ is $A_{34}$ then
   
   $y_1(k+1) = 0.878y_1(k) + 1.31 \cdot 10^{-1}y_2(k) + 0.320u(k) - 0.0046u(k-1) - 0.218$

4. If $y_1(k)$ is $A_{41}$ and $y_2(k)$ is $A_{42}$ and $u(k)$ is $A_{43}$ and $u(k-1)$ is $A_{44}$ then
   
   $y_1(k+1) = 0.601y_1(k) + 1.22 \cdot 10^{-1}y_2(k) + 0.296u(k) - 0.283u(k-1) + 20.9$

5. If $y_1(k)$ is $A_{51}$ and $y_2(k)$ is $A_{52}$ and $u(k)$ is $A_{53}$ and $u(k-1)$ is $A_{54}$ then
   
   $y_1(k+1) = 0.833y_1(k) - 4.24 \cdot 10^{-2}y_2(k) + 1.39u(k) + 4.25u(k-1) + 9.22$
(a) Supply temperature (°C).

(b) Mixed-air temperature (°C).

(c) Valve opening u(k) (% open).

Figure C.2. Membership functions for the TS model with the mixed-air temperature as output.

C.1.2 Sub-model for the mixed-air temperature

The nonlinear MISO model for mixed-air supply temperature is described by the nonlinear function

\[ \hat{y}_2(k+1) = f_2(y_1(k), y_2(k), u(k)), \]

where \( f_2 \) is again a nonlinear mapping. The membership functions for the antecedents \( y_1(k), y_2(k) \) and \( u(k) \) obtained from the projections of the multidimensional fuzzy sets, as before, are shown in Fig. C.2.
The fuzzy rules of this TS model are given by:

1. If $y_1(k)$ is $A_{11}$ and $y_2(k)$ is $A_{12}$ and $u(k)$ is $A_{13}$ then
   \[ y_2(k+1) = 1.45 \cdot 10^{-3} y_1(k) + 9.88 \cdot 10^{-1} y_2(k) + 2.68 \cdot 10^{-1} u(k) + 2.50 \cdot 10^{-1} \]

2. If $y_1(k)$ is $A_{21}$ and $y_2(k)$ is $A_{22}$ and $u(k)$ is $A_{23}$ then
   \[ y_2(k+1) = 1.40 \cdot 10^{-3} y_1(k) + 9.96 \cdot 10^{-1} y_2(k) + 1.09 \cdot 10^{-1} u(k) + 6.01 \cdot 10^{-2} \]

3. If $y_1(k)$ is $A_{31}$ and $y_2(k)$ is $A_{32}$ and $u(k)$ is $A_{33}$ then
   \[ y_2(k+1) = 6.78 \cdot 10^{-3} y_1(k) + 9.94 \cdot 10^{-1} y_2(k) + 9.14 \cdot 10^{-2} u(k) - 6.84 \cdot 10^{-2} \]

4. If $y_1(k)$ is $A_{41}$ and $y_2(k)$ is $A_{42}$ and $u(k)$ is $A_{43}$ then
   \[ y_2(k+1) = -5.13 \cdot 10^{-3} y_1(k) + 9.95 \cdot 10^{-1} y_2(k) + 1.84 \cdot 10^{-1} u(k) + 4.66 \cdot 10^{-1} \]

5. If $y_1(k)$ is $A_{51}$ and $y_2(k)$ is $A_{52}$ and $u(k)$ is $A_{53}$ then
   \[ y_2(k+1) = 3.64 \cdot 10^{-3} y_1(k) + 9.74 \cdot 10^{-1} y_2(k) + 8.00 \cdot 10^{-2} u(k) + 1.08 \cdot 10^{0} \]

### C.2 Affine TS model of the air-conditioning system

The MISO model for the supply temperature is now described by

\[
\hat{y}_1(k+1) = f_1(y_1(k), y_1(k-1), y_2(k), y_2(k-1), u(k)). \tag{C.3}
\]

Note that only the MISO model for the supply temperature is described, i.e., a model where the output is the supply temperature $y_1$. The sub-model for the mixed-air temperature identified in Section 8.3.1 can be utilized in inverse control, because this temperature is not used for control, and the model proved to be a good model.

The antecedent membership functions and the consequent parameters were estimated from the set of identification data presented in Fig. 8.2, by using product-space fuzzy clustering and the least-squares method, as described in Section 2.4. The membership functions for the antecedents $y_1(k)$, $y_1(k - 1)$, $y_2(k)$ and $y_2(k - 1)$ are again obtained from the projections of the multidimensional fuzzy sets derived from the clustering technique in each space, and are shown in Fig. C.3. Note that first a normal non-affine model is identified. Then, the term containing $u(k)$ is suppressed from the premises, and the parameters of the consequent are reidentified using the least-squares algorithm.
This procedure leads to the following five fuzzy rules:

1. If $y_1(k)$ is $A_{11}$ and $y_1(k-1)$ is $A_{12}$ and $y_2(k)$ is $A_{13}$ and $y_2(k-1)$ is $A_{14}$ then
   
   \[ y_1(k+1) = 1.07y_1(k) - 0.21y_1(k-1) + 28.8u(k) - 0.25y_2(k) + 0.39y_2(k-1) - 14.4 \]

2. If $y_1(k)$ is $A_{21}$ and $y_1(k-1)$ is $A_{22}$ and $y_2(k)$ is $A_{23}$ and $y_2(k-1)$ is $A_{24}$ then
   
   \[ y_1(k+1) = 0.44y_1(k) - 0.05y_1(k-1) + 16.9u(k) - 0.56y_2(k) + 0.60y_2(k-1) - 18.6 \]

3. If $y_1(k)$ is $A_{31}$ and $y_1(k-1)$ is $A_{32}$ and $y_2(k)$ is $A_{33}$ and $y_2(k-1)$ is $A_{34}$ then
   
   \[ y_1(k+1) = 1.27y_1(k) - 0.38y_1(k-1) - 0.90u(k) - 2.46y_2(k) + 2.56y_2(k-1) - 0.079 \]

4. If $y_1(k)$ is $A_{41}$ and $y_1(k-1)$ is $A_{42}$ and $y_2(k)$ is $A_{43}$ and $y_2(k-1)$ is $A_{44}$ then
   
   \[ y_1(k+1) = 2.09y_1(k) - 0.91y_1(k-1) + 0.08u(k) - 3.40y_2(k) + 3.34y_2(k-1) - 9.33 \]

5. If $y_1(k)$ is $A_{51}$ and $y_1(k-1)$ is $A_{52}$ and $y_2(k)$ is $A_{53}$ and $y_2(k-1)$ is $A_{54}$ then
   
   \[ y_1(k+1) = 1.12y_1(k) - 0.36y_1(k-1) - 0.96u(k) + 5.52y_2(k) - 5.28y_2(k-1) + 2.99 \]
Symbols and abbreviations

General symbols

\( a, b \) \hspace{1cm} \text{column vectors}
\( A, A_i \) \hspace{1cm} \text{fuzzy sets}
\( \mu, \mu(\cdot) \) \hspace{1cm} \text{membership degree, membership function}
\( R, R_i \) \hspace{1cm} \text{fuzzy relations}
\( \beta, \beta_i \) \hspace{1cm} \text{degrees of fulfillment}
\( N \) \hspace{1cm} \text{set of natural numbers}
\( \mathbb{R} \) \hspace{1cm} \text{set of real numbers}

Symbols related to systems theory

\( P \) \hspace{1cm} \text{system or process under control}
\( M \) \hspace{1cm} \text{model of the system under control}
\( C \) \hspace{1cm} \text{controller}
\( S \) \hspace{1cm} \text{mapping of the sensors}
\( y \) \hspace{1cm} \text{outputs of the process}
\( u \) \hspace{1cm} \text{inputs of the process}
\( x \) \hspace{1cm} \text{states of the process}
\( r \) \hspace{1cm} \text{reference signal}
\( y_i \) \hspace{1cm} \text{i-th output of the process}
\( u_i \) \hspace{1cm} \text{i-th input of the process}
\( x_i \)  
\( i \)-th state of the process

\( m \)  
number of inputs of \( P \)

\( p \)  
number of outputs of \( P \)

\( n \)  
number of states of \( P \)

\( m_i \)  
order of the input \( u_i \)

\( p_i \)  
order of the output \( y_i \)

\( n_i \)  
order of the state \( x_i \)

\( y_m \)  
measured outputs

\( \mathbf{d} \)  
system disturbances

\( \mathbf{d}_m \)  
measurement disturbances

\( \mathcal{Y} \)  
domain of the output variables

\( \mathcal{U} \)  
domain of the input variables

\( \mathcal{X} \)  
domain of the state variables

**Symbols in fuzzy decision making**

\( \Omega = \{\omega\} \)  
set (space) of alternatives

\( \Omega_i \)  
unidimensional space of alternatives

\( \omega, \omega_i \)  
alternative (choice)

\( G, G_i \)  
fuzzy goal

\( C, C_i \)  
fuzzy constraint

\( D \)  
fuzzy decision

\( \zeta_i \)  
fuzzy criterion

\( J \)  
performance function

\( L \)  
superior limit of \( J \)

\( M \)  
total number of alternatives \( \omega_i \)

\( \omega^* \)  
maximizing (optimal) decision

\( \Omega^* \)  
set with the maximizing decisions

\( q \)  
number of fuzzy goals

\( r \)  
number of fuzzy constraints

\( T = q + r \)  
number of fuzzy criteria

\( S \)  
dimension of the space of alternatives

\( \Phi = \{\phi\} \)  
set (space) of effects of alternatives

\( \phi = f(\omega) \)  
effect of an alternative

\( f \)  
mapping from \( \Omega \) to \( \Phi \)

\( \Phi_i \)  
space of the criterion \( \zeta_i \)

\( \phi_i \)  
element in the space \( \Phi_i \)

\( f_i \)  
mapping from \( \Omega \) to \( \Phi_i \)

\( \Phi \)  
Cartesian product of all criteria \( \zeta_i, i = 1, \ldots, T \)

\( f \)  
mapping from \( \Omega \) to \( \Phi \)

\( G_i \)  
induced fuzzy goal

\( R_D \)  
fuzzy relation of fuzzy criteria
Symbols and abbreviations

\( \mathcal{R}_{D_t} \)  
relation of fuzzy criteria in the restricted space \( \Phi_t \)

\( \Phi_t \)  
restriction of \( \Phi \) giving \( f \)

\( \text{Ce}_x \)  
cylindrical extension of \( \Phi_t \) to \( \Phi \)

\( \mu_{\zeta_j} \)  
membership degrees for the criteria \( \zeta_j \) given \( \omega_i \)

\( D \)  
Matrix of membership degrees \( \mu_{\zeta_j} \)

\( \pi \)  
policy with the control actions \( u(k), \ldots, u(k + H_p - 1) \)

\( \pi^* \)  
optimal control policy

\( \zeta_{ij} \)  
criterion \( j \) considered at the time step \( k + i \)

Operators

\( t \)  
\( t \)-norm operator

\( s \)  
\( s \)-norm operator

\( \otimes, \odot \)  
general fuzzy aggregation

\( \oplus_g \)  
fuzzy aggregation of goals

\( \oplus_c \)  
fuzzy aggregation of constraints

\( \text{core}(A) \)  
core of the fuzzy set \( A \)

\( \text{supp}(A) \)  
support of the fuzzy set \( A \)

\( \text{ht}(A) \)  
height of the fuzzy set \( A \)

\( \text{proj}_M(A) \)  
projection of the fuzzy set \( A \) on \( M \)

\( \text{cext}^N_M(A) \)  
cylindrical extension of \( A \) from \( M \) to \( N \)

Abbreviations

AI  
Artificial Intelligence

B&B  
Branch-and-Bound technique

FDM  
Fuzzy Decision Making

FLC  
Fuzzy Logic Control

FMBC  
Fuzzy Model-Based Control

IMC  
Internal Model Control

I/O  
Input/Output

MBPC  
Model-based predictive control

MIMO  
Multiple-Input, Multiple-Output

NARX  
Nonlinear AutoRegressive model with eXogenous inputs

PID  
Proportional Integral Derivative (controller)

SISO  
Single-Input, Single-Output

SQP  
Sequential Quadratic Programming
Summary

The design of control systems is usually carried out differently when classical control design or the design of fuzzy control systems are considered. Classical control design evolved in the past from signal-based control, where only the actual measurements of the different signals in the system are utilized, to model-based control. In the later case, control actions are derived based on the optimization of design criteria, using a model of the system. Control systems are designed following several steps. From the different steps required in model-based control, the modeling of the system, the choice of the design specifications and the selection and design of the controller are the fundamental ones.

The design of conventional fuzzy controllers is quite different from this procedure. These controllers use expert knowledge to build fuzzy rules connecting the inputs and outputs of the controller in the form of If–Then control rules using linguistic terms. The rule base is constituted by the set of all the rules. These type of fuzzy logic controllers are usually tuned by a trial-and-error method using simulations or experiments of the system. Unfortunately this design method has some significant drawbacks. In fact, it is difficult to extract expert knowledge from operators, experiments in an industrial environment are highly undesirable or impossible, and fuzzy logic controllers can not significantly outperform the human operator. However, these controllers are consistent and perform similarly to a human being.

A different design method for fuzzy controllers is used in this thesis, where the design of fuzzy controllers follows the classic control design stages of model-based control systems. Hence, this process starts by modeling the process under control, followed by choosing the design specifications and finally designs the controller. This approach is called fuzzy model-based control (FMBC) in this thesis, where the model, the design specifications and/or the developed controller can be fuzzy. This thesis considers two main approaches for FMBC: the use of fuzzy models in a predictive control scheme, and the consideration of fuzzy performance criteria in the control design.
The development of a model of the system under control is essential for fuzzy model-based control. Models based on first-principles are preferable to other types of models. When possible, first-principle models are used in the thesis. However, these models are difficult to obtain for complex, highly nonlinear or partly known systems. Moreover, this type of models can be difficult to apply in control applications. Therefore, fuzzy modeling and identification is used for a large number of systems. Product-space clustering techniques are used in this thesis to identify fuzzy models directly from data.

Another key factor in fuzzy model-based control is the proper choice of design specifications. The translation of these specifications to performance criteria is straightforward when linear models are used. For nonlinear systems, the definition of performance criteria revealed to be more complex. In fact, the criteria can no longer be analytically computed as in the presence of linear systems. In this way, some definitions lose their computational advantages, and other types of criteria can be defined. One possibility is to define the control goals using the fuzzy set theory. In this approach different fuzzy goals and fuzzy constraints can be defined and combined using different fuzzy aggregators. Note that these fuzzy criteria has the potential of translating better the designer's wishes.

The last of the three more relevant points in control design is the choice of the type of controller to be used. The chosen control scheme must use the advantages of having a model of the system. One of the most widely used model-based control techniques is predictive control, which incorporates the future behaviour of the system when determining the control actions. This control technique has been applied in industry, and can control multivariable (nonlinear) systems under several constraints in an optimal way, with respect to a given objective function. However, MBPC sometimes revealed problems to cope with model errors and disturbances for which no models can be obtained. This thesis uses the internal model control scheme to alleviate this problem. This technique does not control the system directly but the model of the system. The error between the output of the system and the output of the process is used to shift the reference. In most cases, a linear filter must be included in the feedback loop to reduce the gain and improve the robustness characteristics. However, this filter deteriorates the close-loop dynamics in the presence of highly nonlinear systems. Using the information contained in the model, a fuzzy compensator can be used instead of IMC, and the performance is improved. This scheme revealed however to be less robust with respect to variations in the system.

From the several approaches for designing a controller based on a fuzzy model, the simplest way is to invert the model and use it in an open-loop control configuration. The inverse model can be used as a controller, and stable control is guaranteed for minimum phase systems. In practice, any control systems is subjected to model errors and disturbances. These problems can be overcome using internal model control or fuzzy compensation as mentioned before. The inversion of fuzzy models can be made globally or locally. Only local inversion is considered in this thesis.

Two special cases of fuzzy models can be analytically inverted: singleton fuzzy models and Takagi-Sugeno models which are affine with respect to the control actions. A linguistic
fuzzy model with singleton consequents, for which the antecedent membership functions form a partition, and which is monotonous with respect to the singletons, can be analytically inverted. This type of inversion can also be performed for TS fuzzy models affine with respect to a control action, where this control action is not included in the antecedent membership functions. For time-variant systems, the parameters of the fuzzy models can be adapted on-line. A recursive least-squares algorithm is presented for the adaptation of the consequents in the singleton fuzzy model. In the presence of constraints, the inversion can not be applied directly, and a combination of inverse control when the constraints are not violated, with predictive control otherwise, can be utilized. Note that the optimization problem in the predictive control scheme is a nonconvex problem.

A different approach to fuzzy model-based control is to use fuzzy sets at a higher level by defining fuzzy goals and fuzzy constraints. The combination of these fuzzy criteria is aggregated using a properly chosen fuzzy operator. This confluence of goals and constraints can be regarded as a generalized objective function in a predictive control scheme. Fuzzy goals and fuzzy constraints were first introduced for decision making applications. The definition of fuzzy goal, fuzzy constraint and fuzzy decision, and the combination of multiple fuzzy goals and constraints are briefly reviewed in this thesis. A new formulation for fuzzy goals and constraints defined in different spaces is introduced. A solution for multidimensional fuzzy decision making using discrete alternatives is briefly discussed.

Fuzzy decision making (FDM) is applied in this thesis to control. In this type of applications, the decision making problem is a multistage decision making problem to be solved considering all time steps over a given prediction horizon. Most of control applications in the field of decision making are open-loop controlled. This thesis pretends to give the first step in generalizing FDM to closed-loop control. The definitions of fuzzy goal, fuzzy constraint, and their respective aggregation are restricted to control. Fuzzy criteria are utilized in defining a general objective function in model-based predictive control.

Two main problems must be overcome when FDM is used in control: the definition of fuzzy criteria and the choice of proper aggregation operator(s). The first step in defining fuzzy objective functions can be the generalization of the errors, change in control actions and change in the outputs usually defined in classical objective functions. This procedure is followed in this thesis, and two simple examples show clearly the advantage of using fuzzy decision functions in MBPC. The choice of aggregation operators is also very important. In order to not violate hard constraints t-norms should always be chosen. From the different t-norms, the parametric ones allow to tune the response of the system with a single parameter. The same happens for other aggregators, as the generalized mean. However, operators that are not t-norms must be used carefully in order to avoid the violation of hard constraints.

When MBPC is based on fuzzy decision functions, the optimization problem is, in general, nonconvex. Special conditions for which the problem remain convex are presented in this thesis. However, these conditions are quite restrictive, and the more general nonconvex optimization problem must still be solved. A branch-and-bound algorithm solving this type of problems is introduced in this thesis. This algorithm is suitable for problems with small
control horizons and a small number of discrete control actions, because it is quite sensitive to the coarse of dimensionality. Note that the branch-and-bound algorithm can also be used with classical objective functions in MBPC. A different solution to the nonconvex optimization problem is to use special genetic algorithms for predictive control problems. It is shown that genetic algorithms have advantages over branch-and-bound methods for large control horizons, large number of discretization points and for multiple input systems.

Several control methodologies introduced in this thesis were applied to an air-conditioning system. The advantages of some of the proposed methods in terms of control performance become clear. Some extensions of MBPC using fuzzy objective functions are possible for this problem.

It can be concluded that the methodologies proposed in this thesis for fuzzy model-based control result in performance improvements when compared to classical predictive control. The use of fuzzy models for control applications, the introduction of fuzzy criteria in the objective function of model-based predictive control, and the respective use of new algorithms for the nonconvex optimization problems have been tested in different processes. Simulation and real-world applications have shown the advantages of the different methodologies proposed, and the improved control performances.
Samenvatting

Ten opzichte van het ontwerp van klassieke regelsystemen wordt het ontwerp van vage regelsystemen meestal anders uitgevoerd. In klassieke regelsystemen heeft een ontwikkeling plaats gevonden van signaal-gebaseerd regelen, waarbij slechts de actuele metingen van de verschillende signalen wordt gebruikt, naar model gebaseerd regelen. In het laatste geval worden de regelsignalen berekend op basis van de optimalisatie van ontwerpcriteria waarbij gebruik gemaakt wordt van een model van het te regelen systeem. Regelsystemen worden stapsgewijs ontworpen. Van de verschillende stappen die nodig zijn voor model-gebaseerd regelen, zijn de modelvorming van het systeem, de keuze van het ontwerp parameters en het ontwerp van de regelaar zelf, de meest fundamentele.

Het ontwerp van vage regelaars is verschillend van deze klassieke aanpak. Vage regelaars maken gebruik van kennis van experts in een bepaald domein om vage regels te construeren. Deze regels geven een beschrijving van de relatie tussen ingang en uitgang, op basis van een combinatie van "als–dan" regels en linguïstische waarden. De verzameling van al deze regels wordt een "rule-base" genoemd. Bij dit type vage regelaars worden de regels meestal bijgesteld op basis van gerichte aanpassingen die worden beoordeeld met simulaties van het systeem of experimenten aan het systeem. Helaas heeft deze aanpak een aantal moeilijk te overkomen nadelen. Zo is het moeilijk expertkennis uit het veld te extraheren en te formaliseren. Bovendien is het vaak erg duur, ongewenst of bijna onmogelijk om experimenten uit te voeren. Daarnaast is een inherent nadeel van deze aanpak dat dergelijke regelaars niet in staat zullen zijn beter te functioneren dan de beste menselijke evenknie. Een voordeel is echter dat de vage regelaar een consistenter gedrag vertoont.

In dit proefschrift wordt een hiervan verschillende ontwerp-methodiek voor vage regelaars beschreven. Hierbij wordt uitgegaan van de hiervoor beschreven aanpak voor model-gebaseerde regelaars. Daarom begint het ontwerp procedure met de modelvorming van het systeem en wordt het gevolgd door het maken van een keuze van het regelalgoritme afhankelijk van de specificaties en het uiteindelijke ontwerp van de regelaar. Een regelaar
die ontworpen is volgens deze aanpak, wordt een vage model-gebaseerde regelaar, FMBC, (fuzzy model-based control) genoemd. In een dergelijke regelaar kunnen het model, het ontwerp parameters en/of de ontworpen regelaar zelf, vaag zijn. Dit proefschrift beperkt zich tot twee benaderingen voor FMBC: het gebruik van vage modellen in een voorspellende regelaar en het gebruik van vage ontwerp en prestatie criteria.

De ontwikkeling van een model van het te regelen systeem is essentieel voor FMBC. Modellen die gebaseerd zijn op fysisch-chemische behoudswetten zijn natuurlijk in het algemeen te prefereren boven andere meer black-box-achtige modellen. Wanneer dat mogelijk is, wordt in dit proefschrift gebruik gemaakt van dergelijke witte modellen. Echter, modelvorming van complexe niet-lineaire systemen met dit type modellen is in het algemeen moeilijk vanwege de grote hoeveelheid tijd en geld die hiermee gemoeid is. Bovendien zijn dergelijke modellen niet altijd eenvoudig in een regelaar-structuur toe te passen. Daarom zullen voor het merendeel van de in dit proefschrift beschreven systemen vage modelvorming- en identificatiotechnieken worden gebruikt. Clustervormingtechnieken in de produkt-ruimte opgebouwd van in- en uitgangen van het systeem worden gebruikt om vage modellen te identificeren, die slechts gebaseerd zijn op gemeten data van het systeem.

Een andere sleutelelement in FMBC is het maken van een gepaste keuze voor de ontwerpspecificaties. De vertaling van deze specificaties naar prestatie-criteria is eenvoudig wanneer gebruik wordt gemaakt van lineaire modellen. Voor niet-lineaire systemen is de keuze voor prestatie-criteria complexer. Deze criteria kunnen niet meer op een analytische wijze worden berekend vanuit de ontwerpspecificaties. Hierdoor verdwijnen een aantal vooral rekentechnische voordelens waardoor het aantrekkelijker wordt ook andere typen criteria te definiëren en te gebruiken. Een mogelijkheid is om de theorie van de vage verzamelingen te gebruiken voor de definitie van deze criteria. Met deze aanpak kunnen verschillende vage doelen en vage randvoorwaarden gecombineerd worden door gebruik te maken van verschillende vage aggregatie methoden. Merk op dat in feite deze vage criteria bovendien een betere vertaling van de keuzes die een ontwerper moet maken.

Het laatste sleutelelement in het regelaarontwerp is de keuze van het type regelaar dat moet worden toegepast. Het gekozen regelsysteem moet gebruik kunnen maken van het aanwezige systeem model. Een van de meest gebruikte model-gebaseerde regeltechnieken is het “model te gebruiken bij voorspellend regelen” (MBPC). Een dergelijke regelalgoritme neemt het toekomstig systeemgedrag mee in de berekening van de optimale regelacties. Bovendien is het in staat om multivariabele niet-lineaire systemen te regelen, zodanig dat aan alle randvoorwaarden wordt voldaan. Deze techniek wordt daarom ook veel toegepast in de industrie. Soms treden echter problemen met modelfouten op wanneer voor bepaalde systeem delen geen adequaat model kan worden verkregen.

Van de verschillende mogelijkheden om een regelaar op een vaag model te baseren is de meest eenvoudige methode: het inverteren van het vage model om dit vervolgens in een open-lus regelconfiguratie op te nemen. Het geïnverteerde model wordt dan gebruikt als open-lus regelaar, zodat voor minimum-fase systemen een stabiele regelaar resulteert. In de praktijk is echter elk regelsysteem onderhevig aan modelfouten en systeemafwijkingen.
Deze problemen kunnen worden opgelost door gebruik te maken van een regelaar "internal model" of een vage compensatie toe te passen. De inversie van het vage model kan voor een lokaal of globaal werkgebied worden uitgevoerd. Slechts lokale inversie wordt in dit proefschrift behandeld.

Twee speciale gevallen van vage modellen kunnen analytisch worden geïnverteerd: het enkelvoudige vage model (fuzzy singleton model) en de Takagi–Sugeno modellen die affine zijn met betrekking tot de regelactie. Een linguïstisch vage model met enkelvoudige consequenten (singleton consequents), waarbij het antecedent lidmaatschapsfuncties een partitie vormen en waarbij bovendien de enkelvoudige uitgang monotoon stijgt of daalt, kan ook analytisch worden geïnverteerd. Dit type inversie kan ook worden toegepast voor Takagi–Sugeno modellen waarbij de regelactie niet in het antecedent deel van het model voorkomt. Voor tijd-varie rende systemen moeten de parameters van het vage model worden aangepast aan de veranderde omstandigheden. Hiervoor is een recursieve kleinste kwadraten methode ontwikkeld om de consequenten van het vage enkelvoudige model aan te passen. Wanneer randvoorwaarden een rol spelen is het zeker niet meer mogelijk een analytische inversie uit te voeren. In dat geval kan een combinatie van inverse regeling, indien de randvoorwaarden niet mee spelen, met een voorspellende regelaar, indien de randvoorwaarden wel meegenomen moeten worden, uitkomst bieden. Merk daarbij op dat het optimalisatieprobleem in de voorspellende regelaar een niet-convex probleem is.

Een andere aanpak voor vaag model-gebaseerd regelen is het gebruik van vage verzamelingen voor de definitie van vage doelen en vage randvoorwaarden. De combinatie van deze vage criteria wordt geaggregeerd door gebruik te maken van een goed gekozen vage operator. Deze combinatie van doelen en randvoorwaarden kan worden voorgesteld als een gegeneraliseerde doelfunctie in een voorspellende regelaar structuur. Vage doelen en vage randvoorwaarden werden geïntroduceerd voor toepassingen die betrekking hadden op het nemen van beslissingen. De definities van vage doelstellingen, vage randvoorwaarden, vage besluitvorming (fuzzy decision making) en de combinatie van meerdere vage doelen en randvoorwaarden worden beschreven in het proefschrift. Een nieuwe formulering van vage doelstellingen en vage randvoorwaarden, die worden gedefinieerd voor verschillende ruimten, wordt eveneens geïntroduceerd. Daarnaast wordt een oplossing voor multi-dimensionale vage besluitvorming waarbij discrete alternatieven worden gebruikt, kort beschreven.

De toepassing van vage besluitvorming (fuzzy decision making) in een regelsysteem wordt uitgebreid in dit proefschrift beschreven. In dit type toepassingen kan het meertals besluitvormingsprobleem worden opgelost door de toekomstige tijdstappen over een bepaalde voorspellingshorizon mee te nemen. De meeste beschreven regeltoepassingen op het gebied van vage besluitvorming zijn open-lus van aard. Dit proefschrift pretendeert een eerste aanzet te geven om te komen tot een generalisatie van de toepassing van vage besluitvorming op gesloten-lus regelsystemen. De definities van vage doelen, vage randvoorwaarden en de aggregatie daarvan wordt beperkt tot het gebied van regelsystemen. Vage criteria worden gebruikt om een algemene doelfunctie te definiëren voor model gebaseerde voorspellende regelaars.
De twee belangrijkste problemen die moeten worden overwonnen wanneer vage besluitvorming wordt toegepast in regelsystemen zijn: de definitie van vage criteria en de keuze van geschikte aggregatie operatoren. De eerste stap om vage doelfuncties te definiëren is uit te gaan van een generalisatie van fouten, zoals veranderingen in de regelakties en veranderingen in de outputs die meestal worden gebruikt in klassieke doelfuncties. Deze procedure wordt ook gevolgd in het proefschrift en twee eenvoudige voorbeelden laten duidelijk de voordelen van vage besluitvorming in voorspellend regelen zien. De keuze van aggregatie operatoren is ook erg belangrijk. Om harde randvoorwaarden niet te overschrijden zullen altijd $t$-normen moeten worden gebruikt. Van de verschillende te gebruiken $t$-normen bieden de parametrische normen de mogelijkheid de systeemresponsie met slechts één parameter in te stellen. Hetzelfde is ook mogelijk met andere aggregatoren zoals het gegeneraliseerde gemiddelde. Echter, operatoren die geen $t$-norm zijn kunnen slechts onder bepaalde voorwaarden worden gebruikt om te voorkomen dat harde randvoorwaarden worden overschreden.

Wanneer de voorspellende regelaar is gebaseerd op vage besluitvormingsfuncties, dan is het optimalisatieprobleem in het algemeen niet convex. Speciale condities waarvoor het probleem wel convex blijft worden gepresenteerd in het proefschrift. Echter, deze condities zijn wel restrictief wat betekent dat meer algemene optimalisatie problemen nog steeds met complexe technieken moeten worden opgelost. Voor deze gevallen wordt voorgesteld een "branch-and-bound" algoritme te gebruiken. Dit algoritme is toepasbaar voor problemen met een korte regelhorizon en een relatief klein aantal discrete regelacties, omdat de toepasbaarheid van deze methode snel afneemt bij toenemen dimensie. De branch-and-bound methode kan ook worden gebruikt voor de optimalisatie van klassieke doelfuncties zoals gebruikt in standaard voorspellende regelaars. Een andere mogelijkheid is om het optimalisatieprobleem aan te pakken met een speciaal genetisch algoritme. Het wordt aangetoond dat genetische algoritmen bepaalde voordelen hebben ten opzichte van branch-and-bound methode, vooral als sprake is van een langere regelhorizon, met een groot aantal discretisatie punten en als er sprake is van systemen met meerdere inputs.

Verschillende regelmethoden die zijn geïntroduceerd in dit proefschrift zijn toegepast op een airconditioning systeem. De voordelen van sommige van de voorgestelde methoden komen in deze toepassing duidelijk naar voren.

Er kan worden geconcludeerd dat de voorgestelde methoden voor vage model-gebaseerde regelaars resulteren in verbeteringen van de kwaliteit van de regelacties in vergelijking met de klassieke aanpak van voorspellend regelen. Het gebruik van vage modellen voor regelaar toepassingen, de introductie van vage criteria in de doelfunctie van de voorspellende regelaar en het gebruik van nieuwe methoden voor niet-convexe optimalisatie problemen zijn getest op verschillende processen. Gesimuleerde en praktische toepassingen laten de voordelen en de verbeterde regelaarkwaliteit van de verschillende voorgestelde methoden duidelijk zien.
Resumo

O *design* de sistemas de controlo clássico é habitualmente efectuado de forma diferente do *design* de sistemas de controlo vagos (*fuzzy*). Os sistemas clássicos de controlo evoluíram no passado de controlo baseado em sinais, onde apenas os valores medidos dos diferentes sinais no sistema são considerados, para controlo baseado em modelos. Neste último caso, as acções de controlo são derivadas com base na optimização de critérios de *design*, usando para este efeito um modelo do sistema. Os sistemas de controlo são projectados seguindo um pré-determinado número de passos. Considerando os diversos passos requeridos pelo controlo baseado em sistemas, a modelação do sistema, a escolha das especificações de *design* e a selecção e *design* do controlador são sem dúvida os fundamentais.

O projecto de sistemas de controlo *fuzzy* convencionais é bastante distinto do procedimento descrito anteriormente. Estes controladores usam conhecimento proveniente de peritos para construir regras *fuzzy* ligando as entradas e as saídas do controlador, utilizando termos linguísticos na forma de regras de controlo “se–então”. A base de regras é constituída pelo conjunto de todas as regras. Este tipo de controladores *fuzzy* são normalmente afinados por um método de tentativa-erro usando simulações ou experiências no sistema. Infelizmente, este método de projectar controladores *fuzzy* apresenta desvantagens significativas. De facto, é difícil obter as regras dos peritos, as experiências em ambiente industrial são altamente indesejáveis ou mesmo impossíveis, e este tipo de controladores *fuzzy* não podem obter desempenhos muito superiores aos operadores humanos. No entanto, estes controladores são consistentes, e obtêm desempenhos similares aos obtidos por seres humanos.

Esta tese utiliza um método diferente de projectar controladores *fuzzy*, onde o *design* deste tipo de controladores segue os passos do *design* de sistemas de controlo clássico para sistemas de controlo baseado em modelos. Desta forma, o projecto do controlador inicia-se com a modelação do processo a controlar, seguido pela escolha das especificações de *design*, e finalmente escolhe o tipo de controlador para o processo. Nesta tese, esta abordagem é denominada controlo baseado em modelos utilizando conjuntos *fuzzy* (*fuzzy model-based*
control (FMBC)), onde o modelo, as especificações e/ou o controlador desenvolvido podem ser fuzzy. Duas formas de abordar o problema são consideradas em FMBC: a utilização de modelos fuzzy num esquema de controlo predictivo, e a introdução de critérios de desempenho vagos no projecto de controlo.

O desenvolvimento de um modelo do sistema sob controlo é essencial em FMBC. Os modelos baseados em leis físicas ou químicas dos sistemas são preferíveis a outro tipo de modelos. Este tipo de modelos é usado sempre que possível na tese. No entanto, estes modelos são difíceis de obter, sobretudo para sistemas complexos, altamente não lineares ou parcialmente desconhecidos. Conjuntamente, este tipo de modelos é por vezes difícil de aplicar em controlo. Pelas razões apontadas, a modelação e a identificação fuzzy é utilizada num largo número de sistemas. As técnicas de clustering baseadas no espaço resultante do produto das variáveis do modelo, são usadas na tese para a identificação de models fuzzy obtidos directamente de medições das variáveis do processo.

Outro factor-chave em FMBC é a escolha apropriada das especificações de design. A tradução destas especificações para critérios de desempenho é directa quando modelos lineares são utilizados. Em sistemas não lineares, a definição de critérios de desempenho revela-se como sendo bastante mais complexa. De facto, os critérios não podem continuar a ser calculados analiticamente como na presença de sistemas lineares. Desta forma, algumas destas definições perdem as suas vantagens computacionais, e outros tipos de critérios podem ser definidos. Uma possibilidade será a definição dos objectivos de controlo usando a teoria dos conjuntos vagos. Nesta abordagem, diferentes objectivos fuzzy e restrições fuzzy podem ser combinados utilizando diferentes operadores fuzzy. É de realçar que os critérios fuzzy têm o potencial de traduzirem melhor os desejos do projectista.

O último dos três pontos mais relevantes em projectos de controlo é a escolha do tipo de controlador a ser utilizado. O esquema de controlo escolhido deverá utilizar as vantagens da disposição de um modelo do sistema.

Uma das técnicas de controlo baseada em modelos mais largamente utilizada é o controlo predictivo, a qual incorpora o comportamento futuro do sistema quando as acções de controlo são determinadas. Esta técnica de controlo tem vindo a ser aplicada na indústria, e tem a capacidade de controlar sistemas multivariables não-lineares sujeitos a várias restrições de uma forma óptima, em relação a uma determinada função objectivo. No entanto, o controlo predictivo baseado em modelos revela alguns problemas para lidar com erros no modelo e perturbações para as quais não é possível obter um modelo. Na tese, o esquema de controlo baseado em um modelo interno (internal model control) é utilizado para reduzir o efeito destes problemas.

Das várias abordagens possíveis para projectar um controlador baseado num modelo fuzzy, a forma mais simples será inverter o modelo e utilizá-lo num configuração de controlo em anel aberto. O modelo inverso pode ser usado como controlador, e um controlo estável é garantido para sistemas de fase mínima. Na prática, qualquer sistema de controlo está sujeito a erros de modelação e perturbações. Estes problemas podem ser ultrapassados utilizando
um controlo baseado num modelo interno, ou compensação fuzzy. A inversão de modelos fuzzy pode ser efetuada globalmente ou localmente. Nesta tese, a inversão local é a única considerada.

Dois tipos de modelos fuzzy podem ser invertidos: modelos fuzzy com consequentes numéricos (singleton fuzzy models) e modelos fuzzy do tipo Takagi-Sugeno (TS), afins em relação a uma dada acção de controlo. Um modelo fuzzy com consequentes numéricos, para o qual as funções de pertença dos antecedentes formam uma partição, e que são monótonas em relação aos consequentes, pode ser invertido analiticamente. Este tipo de inversão pode ser igualmente efectuado para modelos fuzzy TS, afins em relação a uma acção de controlo, quando esta variável não é incluída nas funções de pertença dos antecedentes. Para sistemas variáveis no tempo, os parâmetros dos modelos fuzzy podem ser adaptados em-linha. A adaptação dos consequentes num modelo fuzzy com consequentes numéricos é efectuada utilizando um algoritmo de mínimos quadrados recursivo. Na presença de restrições, a inversão não pode ser aplicada directamente, e desta forma, uma combinação de controlo inverso quando as restrições não são violadas, com controlo predictivo no caso contrário, pode ser aplicada. É de realçar que o problema de optimização no esquema de controlo predictivo é um problema não convexo.

Uma abordagem diferente ao controlo baseado em modelos utilizando conjuntos fuzzy será a utilização de conjuntos vagos a um nível mais elevado, através da definição de objectivos vagos e restrições vagas. Estes critérios fuzzy são agregados utilizando um operador fuzzy convenientemente escolhido. A combinação de objectivos e restrições pode ser visto como uma função objectivo generalizada num esquema de controlo predictivo. Os objectivos vagos e as restrições vagas foram primeiramente introduzidos em aplicações de tomada de decisões. As definições de objectivo vago, restrição vaga e decisão vaga, e a combinação de múltiplos objectivos e restrições fuzzy são brevemente revistos nesta tese. Uma nova formulação para objectivos e restrições vagos definidos em espaços diferentes é introduzida. Uma breve discussão de problemas de tomada de decisões fuzzy multidimensionais é igualmente apresentada.

Tomada de decisões vagas (fuzzy decision making (FDM)) são aplicadas nesta tese em controlo. Neste tipo de aplicações, o problema de tomada de decisões é multi-etapas, e deve ser resolvido considerando todos os passos de tempo num dado horizonte de predição. A maior parte das aplicações de controlo no campo da tomada de decisões são controladas em anel-aberto. Esta tese pretende dar o primeiro passo com vista à generalização de FDM para controlo em anel-fechado. As definições de objectivo vago, restrição vaga, e a sua respectiva agregação é restringida para o campo do controlo. Critérios vagos são utilizados na definição da função objectivo em controlo predictivo baseado em modelos.

Quando FDM é aplicada em controlo, os dois principais problemas a serem resolvidos são: a definição dos critérios vagos e a escolha apropriada do(s) operador(es) agregando esses critérios. O primeiro passo na definição de funções objectivo fuzzy pode ser a generalização dos erros, variação nas acções de controlo e a variação nas saídas, habitualmente definidos nas funções objectivo clássicas. Esta generalização é efectuada nesta tese, e dois simples
exemplos demonstram claramente as vantagens de usar funções objetivo fuzzy em MBPC. A escolha de operadores de agregação é igualmente bastante importante. De forma a não violar as restrições, normas triangulares devem ser sempre escolhidas. De entre as diferentes normas deste tipo, as paramétricas permitem a afinção da resposta do sistema usando um único parâmetro. O mesmo ocorre para outros operadores de agregação, como a média generalizada. No entanto, operadores que não são normas triangulares devem ser usadas de forma cautelosa, de forma a não violar as restrições físicas do sistema.

Quando MBPC é baseado em funções de decisão fuzzy, o problema de otimização é, em geral, não convexo. Condições especiais para o qual o problema permanece convexo são apresentadas nesta tese. Estas condições são no entanto bastante restrictivas, e o problema de otimização não convexo terá de ser igualmente resolvido. Um algoritmo do tipo branch-and-bound é introduzido nesta tese para resolver este tipo de problemas. Este algoritmo é apropriado para problemas com pequenos horizontes de controlo e um número pequeno de acções de controlo discretas, porque é bastante sensível a discretizações com elevado número de pontos. O algoritmo branch-and-bound pode ser também usado com funções objectivo clássicas em MBPC. Outra solução possível para o problema de otimização não convexo é a utilização de um algoritmo genético desenvolvido para problemas de controlo predictivo. Este algoritmo revela vantagens sobre os métodos de branch-and-bound para horizontes de controlo largos, grande número de pontos de discretização e para sistemas com várias entradas.

Várias das metodologias de controlo introduzidas nesta tese foram aplicadas num sistema de ar-condicionado. As vantagens de alguns dos métodos propostos em termos de desempenho são bastante claras. Algumas extensões de MBPC com funções objectivo vagas são igualmente aplicadas neste problema.

Pode-se concluir que as metodologias de controlo propostas nesta tese para FMBC resultam em melhores desempenhos, quando comparados com controlo predictivo clássico. A utilização de modelos fuzzy em aplicações de controlo, a introdução de critérios vagos nas funções objectivo de controlo predictivo, e o uso de novos algoritmos para problemas de otimização não convexos foram testados em vários processos. Aplicações reais e simulações demonstraram as vantagens das diferentes metodologias propostas, e apresentaram melhoramentos significativos em termos de desempenho.
João Miguel da Costa Sousa was born on June 24, 1966 at Lisbon, Portugal. In 1989 he graduated in mechanical engineering, branch of systems from the Technical Superior Institute, at the Lisbon University of Technology. In 1989 he started to work as a research and teaching assistant at the Department of Mechanical Engineering, Technical Superior Institute in Lisbon. In 1992, he received the M.Sc. degree with honors in mechanical engineering, profile of systems from the Technical Superior Institute, at the Lisbon University of Technology. He served the Portuguese Army during the year of 1993. In the same year he started his doctoral research at the Lisbon University of Technology.

In 1994 he continued his research as a Ph.D. student in the field of fuzzy model-based control at the Control Laboratory of the Department of Electrical Engineering, Faculty of Information, Technology and Systems at the Delft University of Technology. He returned back to his position at the Group of Control, Automation and Robotics, Department of Mechanical Engineering, Technical Superior Institute, at the Lisbon University of Technology.
Acknowledgements

First, I would like to thank my supervisors Prof. Henk Verbruggen and ir. Piet Bruijn. They provided the necessary guidance, but gave me also the possibility of free flight, which made this Dutch experience really outstanding. I thank them also for their continuous support, helpful suggestions, critical remarks and their hospitality to me, sometimes unusual, ideas. Especially to Prof. Henk Verbruggen I would like to express my sincere thanks for the hospitality shown since the first moment.

The first of the Stellingen is a special tribute to Prof. Medina Martins who persisted to work in fuzzy logic for many, many years, against all criticism. His contributions in the field made me change my mind in the late 80’s. Thank you.

This work has been carried out at the Control Laboratory, Department of Electrical Engineering, Faculty of Information, Technology and Systems at the Delft University of Technology. I would like to express my sincere thanks to Robert Babuška and Uzay Kaymak, for the fruitful discussions, which resulted in many papers. The Norwegian influence in also present in a paper together with Magne Setnes. Hubert te Braake contributed directly to this thesis by translating the summary into Dutch. I thank to all the members of the control group that smoothed my way, and provided a really great working environment. A special thank to my office mates Magne, Hans and Gerard for supporting me in the last months of writing. To all the members of the Control Laboratory I wish to thank them for creating a pleasant and competent working atmosphere. My special thanks go to Uzay, Robert, Magne, Hans, Hubert, Bart, Jacquelen, Ton, Gerard, Bert, Rob, Ben, Hans, Ger, Anja, Peter, Dan, Elly and Will. I thank also the master students who contributed to this work. A special thank to Christian Onnen for his work in genetic algorithms.

The application in Chapter 8 was developed at the Laboratory for Refrigerating Engineering and Indoor Climate Control of the Mechanical Engineering Faculty at the Delft University of Technology. I thank all the personal that helped me. A special thanks goes to Dr.
van Paassen for helping me at the different stages of the implementation of the developed controllers.

I am grateful to Prof. Sá da Costa for providing the possibility of leaving the Group of Control, Automation and Robotics of the Department of Mechanical Engineering at the Technical University of Lisbon, for the period of my Ph.D.

I am greatly indebted to the Portuguese foundation Fundação Calouste Gulbenkian for the financial support during a great deal of my staying in Delft.

My family and friends in Portugal have provided support for my endeavours, even from distance. A special thanks to my nieces Joana and Mariana. Their constant joy and happiness can give moral support for months. Fortunately, they are still young enough to understand this. :-) To the old Delft friends around Sebastiaansbrug and the “Spanish community” I thank for all the spiritual support.

Daily telephone calls and e-mails are far from sufficient. You’re the first reason to go back. Between us thanks are not necessary. Now I hope to forget the meaning (and especially the feeling...) of the Portuguese word saudade. This thesis is to you Ana.
References


Author index

A
Afshari, A., 163
Ames, R., 136
Asai, K., 89
Assilian, S., 5
Åström, K., 59, 173

B
Belanger, P., 4
Baas, S.M., 90
Baldwin, J., 102
Barret, C.H., 29, 33–35, 150, 151
Batur, C., 63
Bellman, R.E., 8, 9, 39, 89, 90, 92, 93, 96, 100–104, 119, 125, 127
Berenji, H., 75
Bordons, C., 10, 43, 44, 48
Barnard, G., 163
Boyd, S.P., 29, 33–35, 150, 151
Braae, M., 59
Brown, M., 19, 21
Bruijn, P.M., 43, 52, 107, 108, 128
Buckles, B.P., 136
Bushnell, M., 129

C
Camacho, E., 10, 43, 44, 48
Chan, C.-C., 63
Chang, C., 103
Chen, S.-J., 90, 121
Chen, X., 129
Clarke, D., 44, 46, 49
Cutler, C., 44

D
Dage, G., 136
Davidor, Y., 136
Davis, L., 136
Dewilde, P., 13
Dexter, A., 163
Dijkman, J.G., 121
Doyle, J., 36, 38
Driankov, D., 3, 5–7, 16, 18, 59
Dubois, D., 90, 120, 121

E
Economou, C., 43, 50, 51, 60
Esogbue, A., 102

F
Felix, R., 107
Filev, D., 121, 189
Fischer, M., 63, 64
Fletcher, R., 151
Folger, T., 149
Francelin, R., 102
Freg, C.-P., 23
Freund, R., 107
Friedman, J., 19, 61
Fu, K., 102, 120

235
Fung, L., 102, 120

G
Garcia, C., 50
Gath, I., 23
Gawthrop, P., 63, 103
George, R., 136
Georgescu, C., 163
Gerdes, I.S., 136
Geva, A., 23
Giddens, T.D., 141
Gill, P.E., 30, 48, 78, 127, 132
Glover, K., 36, 38
Gluss, B., 104
Goldberg, D.E., 136, 138, 141
Gomide, F., 102
Gorez, R., 7, 16
Goser, K., 136
Graham, B., 59
Guo, K., 102
Gustafson, D., 21, 23

H
Hägglund, T., 173
Hamacher, H., 120
Harris, C., 19, 21
Heijnen, J., 79
Hellendoorn, H., 3, 5–7, 16, 18, 59
Hellinga, C., 79
Henson, M., 109
Hirota, K., 16
Holland, J.H., 136
Horowitz, E., 128, 153
Howard, R., 102
Hunt, K., 63, 103
Hwang, C.-L., 90, 121

I
Ibaraki, T., 131
Isermann, R., 63, 64, 135

J
Jager, R., 17, 18, 66, 94
Jang, J., 75
Jia, L.-M., 90, 103
Johansen, T., 7
Jordan, M., 64

K
Kacprzyk, J., 8, 10, 90, 92, 93, 95, 96, 101–104, 136
Kang, G., 22
Kanstein, A., 136
Karr, C.L., 136
Kessel, W., 21, 23
Khedr, P., 75
Kinzel, J., 75
Klawonn, F., 75
Klir, G., 17, 18, 106, 149, 155, 189
Kosko, B., 16
Krijgsman, A., 79
Krishnapuram, R., 23
Kruse, R., 75
Kuchler, T., 167
Kuipers, B., 59
Kwakernaak, H., 90

L
Langari, R., 89
Lawler, E., 128
Levenhagen, J., 164
Li, W., 3, 4, 31, 108
Ling, K., 163
Linkens, D.A., 44, 136
Ljung, L., 76, 77
Luenberger, D., 151
Lute, P., 165, 180
Luyben, K., 79

M
Mahfouf, M., 44
Mamdani, E., 5, 16, 17, 31, 75, 89
Mattsoson, R., 136
Mead, R., 48, 127
Meiritz, A., 107
Michalewicz, Z., 138
Mitten, L., 128
Miyamoto, S., 89, 103
Mohtadi, C., 44, 46, 49
Morari, M., 43, 50, 51, 60
Murray, W., 30, 48, 78, 127, 132

N
Nelder, J., 48, 127
Nelles, O., 63
Newell, R., 59
Nyongesa, H.O., 136

O
Onnen, C., 135

Palm, R., 3, 7, 59
Palsson, B., 43, 50, 51, 60
Papon, J., 44
Pedrycz, W., 16, 21, 59
Peng, X., 164–166
Petry, F.E., 136
Phoa, Y., 103
Pilsworth, B., 102
Pleeging, M., 121
Potts, J.C., 141
Prabhu, D., 136
Prade, H., 90, 120, 121
Procyk, T., 75

R
Rahmani, K., 163
Ramaker, B., 44
Rault, A., 44
Reinfrank, M., 5, 6, 16, 18, 59
Richalet, J., 43, 44, 48
Rumelhart, D., 64
Rutherford, D., 59

S
Safteruk, K., 102
Sahni, S., 128, 153
Sbarbaro, D., 63, 103
Schoenaer, M., 139
Seber, G., 103
Seborg, D., 109
Setnes, M., 103
Shoureshi, R., 163
Sieja, T., 136
Singh, M., 16
Slotine, J., 3, 4, 31, 108
Soeterboek, R., 10, 43, 44, 46
Spethmann, D., 164
Srikant, R., 136
Srinivasan, A., 63
Staniewsky, P., 102
Stein, W., 102
Sugeno, M., 16, 19, 20, 22, 59, 72, 80, 89
Surmann, H., 136

Takagi, T., 16, 19, 20, 59, 72, 80
Tanaka, K., 16, 59
Terano, T., 89
Testud, J., 44
Theologidu, M., 102
Thia, B., 103
Tuffs, P., 44, 49
Turksen, I., 120

V
van Can, H., 79, 83
van Haeringen, H., 121
van Nauta Lemke, H.R., 93, 103, 121
van Paassen, A., 165, 180
Verhaegen, M., 13, 148

W
Wang, L.-X., 16
Wang, W., 103
Weber, S., 120
Wertz, V., 7, 16
Wild, C., 103
Wood, E., 128
Wright, M., 30, 48, 78, 127, 132

X
Xanthakis, S., 139

Y
Yadav, S.B., 141
Yager, R., 90, 93, 120, 121, 158, 189
Yasunobu, S., 89, 103
Yen, J., 89
Yoshinari, Y., 16
Yuan, B., 17, 18, 106, 155, 189

Z
Zadeh, L.A., 5, 6, 8, 9, 16, 17, 39, 89, 90, 92, 93, 96, 100–104, 119, 125
Zbikowski, R., 63, 103
Zeng, X., 16
Zhang, X.-D., 90, 103
Zhao, J., 7, 16
Zhou, K., 36, 38
Zimmermann, H.-J., 5, 90, 95, 107, 121, 189
Zysno, P., 121
Subject index

A
actuator effort, see specifications
adaptation of fuzzy models, 75
adaptive control structures, 75
adaptive decision alternatives, 187
aggregation of criteria, 105
aggregation operators, 95, 106, 119
  min, 119
  generalized mean, 121
  parametric t-norm, 120
  product t-norm, 120
  Yager t-norm, 120
air-conditioning, 163
  controllers, 172
  fan-coil system, 165
fuzzy models, 167, 197
primary system, 164
terminal system, 164
algorithm
  branch-and-bound
    bounding rule, 129
    branching rule, 129
    in classical MBPC, 128, 133
    in fuzzy MBPC, 153, 156
  search tree, 129
  selection rule, 129
  combined control scheme, 78

Gustafson-Kessel, 23, 74, 195
  recursive least-squares, 76
antecedent membership functions, 23
applications
  air-conditioning, 112, 132, 157, 163
  linear system, 112, 122, 151
  pressure control, 79, 145
  system with dead-zone, 55
asymptotic tracking, 33

B
black-box model, 14
block diagram, 2

C
classical performance criteria, 35
classical performance specifications, 32
cluster
  covariance matrix, 23, 196
  prototypes, 23, 196
compatible cluster merging, 23
compositional rule of inference, 17
consequent parameters, 23, 24
constraints
  hard, 30, 104
  implementation in GAs, 138
  level, 47
rate, 47
soft, 30, 104
control
actions, 2, 101
closed-loop, 103
criteria, 108
model-based, see model-based control
open-loop, 60
signal-based, 1
control design
controller, 4
design specifications, 4
modeling, 4
steps, 4
control horizon, 45
control problems
regulation, 3
tracking, 3
control systems
classical, 1
design, 1
fuzzy, 1
controller, 2
conventional fuzzy controller, 5
convex optimization
algorithms, 150
in fuzzy predictive control, 148
core, 190
cylindrical extension, 97, 193

D
data samples, 21
decision functions, 108
defuzzification
center-of-gravity, 6, 18
fuzzy mean, 19
mean-of-maxima, 6, 18
design
goals, 28
objectives, 28
specifications, 9, 28, 39
nonlinear systems, 31
disturbances, 2, 49
effects of alternatives, 96
encoding variables, 140
enumerative search, 134
F
feedback control system, 2
fuzzy
clustering, 16, 23
complement, 191
constraints, 8, 39, 91, 105
control
descriptive approach, 89
perspective approach, 89
controller, 5, 7
data base, 5
defuzzification, see defuzzification
fuzzification, 5
rule base, 5
criteria, 97, 105
in control, 108
in different spaces, 96
in MBPC, 107
decision, 91
goals, 8, 39, 91, 105
identification, 21
intersection, 191
logic control, 5
model, 7, 8
linguistic, see linguistic fuzzy model
singleton, see singleton fuzzy model
Takagi–Sugeno, see Takagi–Sugeno fuzzy model
modeling, 13, 15
models
input delays, 70
partition, 66
partition matrix, 23, 196
performance criteria, 7, 39
relation, 97, 192
union, 191
fuzzy arithmetic based interpolative reasoning, 104
fuzzy compensation, 10, 52, 188
fuzzy decision making, 40, 90
  in control, 101
fuzzy model-based control, 7, 181
  fuzzy decision functions, 89
  fuzzy models, 59
  using fuzzy models, 182
  with fuzzy objective functions, 183
fuzzy set, 189
  normal, 190
  operations, 190

G
  generalized predictive control, 44
  genetic algorithms, 135
    operators, 141
  global inversion, 62
  gray-box modeling, 16

H
  heating valve, 166
  height, 190
  heuristic search, 131
  hierarchical fuzzy goals, 186

I
  identification, 8, 13
    neuro-fuzzy, 21
    spline adaptive techniques, 21
  induced fuzzy goal, 96
  integral action, 49
  internal model control, 10, 49
  inverse model control, 61
  inversion
    fuzzy models, 60
    singleton fuzzy model, 64
    Takagi-Sugeno fuzzy model, 71
  inversion methods, 63
    direct inverse learning, 63
    specialized inverse learning, 63
  invertibility, 67
  iterative optimization, 48

L
  least-squares method, 24
  linguistic fuzzy model, 16
  linguistic variable, 17
  lower bound, 132, 154

M
  measured output, 2
  measurement disturbances, 2
  membership function, 189
  mixed-air temperature, 166
  model, 101
    affine TS fuzzy model, 73
    autonomous, 3
    black-box, see black-box model
    gray-box, see gray-box model
    inverse, 60
    linear, 148
    MIMO, 14
    non-autonomous, see time-variant
    time-variant, 3
    white-box, see white-box model
  model-based predictive control, 10, 44
  model-plant mismatch, 49
  modeling, 8, 13
    in MBPC, 48
  multicriterion optimization, 29
  multiple fuzzy criteria, 94
  multistage decision making, 101

N
  neural learning methods, 16
  norm
    ∞-norm, 37
    p-norm, 36
    s-norm, 191
    l-norm, 191
    1-norm, 36
    2-norm, 36
    average-absolute value, 37
    induced p-norm, 38
    ITAE norm, 37
    peak, see ∞-norm
    root-mean-square, 37
    signals, 35
  norms of systems, 38
  number of clusters, 23
subject index

O
objective functions, 45, 130, 140
  classical, 109
  fuzzy, 110
open-loop feedback control, 47
optimal control, 29
optimal decision, 94
optimization problems, 127, 185
orthogonal least squares, 16
output trajectory, see reference
output variables, 2
overshoot, 33

P
parameter estimation, 23
partial inversion, 62
perfect control, 51, 60
performance, 31
  criteria, 107
  fuzzy, see fuzzy performance criteria
policy, 105
population structure, 142
prediction horizon, 45
process, 2
product-space fuzzy clustering, 21
projection, 192

Q
quadratic problem, 48

R
receding horizon principle, 46
reference, 2, 3, 46
regressand, 21
regressor, 21
reinforcement learning, 75
research topics, 10
rise time, 33
robustness, 31

S
self-organizing linguistic controller, 75
  semi-norm, 35
  sensors, 2
  sequential quadratic programming, 48, 132
set (space) of alternatives, 91, 100
  multidimensional, 99
set of control actions, 105, 129, 154
settling time, 33
simplex method, 48
singleton fuzzy model, 19, 61
specifications
  actuator effort, 32, 35
  input/output, 32
  regulation, 32, 34
stability, 31, 50
state vector, 15
state-space model, 13
steady-state error, 53
step response, 32
structure identification, 21
supply temperature, 166
support, 190
system, 101
  deterministic, 102
  fuzzy, 102
  MIMO, 15
  MISO, 15
  stochastic, 102

T
Takagi-Sugeno fuzzy model, 16, 19, 61
  affine, 20
  termination conditions, 144
  termination time, 102
tree search, 154

U
undershoot, 33
unified predictive control, 44
universal approximators, 16
upper bound, 132, 154

V
validity measures, 23

W
weighted-max objective, 30
weighted-sum objective, 30
white-box model, 8, 13
working area temperature, 166

Z

zero offset, 51