Stellingen

behorende bij het proefschrift

Fuzzy Control of Multiple-Input Multiple-Output Processes

Stanimir Mollov
Delft, 3 december 2002

1) Alhoewel model-gebaseerd regelen tot de regelmethoden met terugkoppeling wordt gerekend, kan deze methode worden toegepast op niet-lineaire (*fuzzy*) modellen die op een niet-lineaire manier afhangen van het huidige regelsignaal (dit proefschrift, Hoofdstuk 2 en Hoofdstuk 4).

   Although model predictive control belongs to the feedback control methods, it can be applied to nonlinear (*fuzzy*) models that depend in a nonlinear way on the current control signal (this thesis, Chapter 2 and Chapter 4).

2) Het conservatisme van de robuuste stabiliteitsbeperkingen is evenredig met het aantal Markov parameters van het convolutiemodel dat wordt gebruikt om het proces weer te geven (dit proefschrift, Hoofdstuk 5).

   The conservatism of the robust stability constraints is proportional to the number of Markov parameters of the convolution model used to represent the process (this thesis, Chapter 5).

3) Terwijl De Vries en Van den Boom (1997) de parameter p die wordt gebruikt in (5.30) en (5.31) hebben geïntroduceerd als een filterversterkingsfactor, bepaald deze parameter in essentie een compromis tussen de amplitude en de snelheid van de reactie van de beperkingen (dit proefschrift, Hoofdstuk 5).

   While de Vries and van den Boom (1997) introduced the parameter p used in (5.30) and (5.31) as a filter gain, essentially it provides a compromise between the amplitude and the speed of the constraints' reaction (this thesis, Chapter 5).

   de Vries, R. A. J. and van den Boom, T. J. J. (1997), Robust Stability Constraints for Predictive Control, In *Proceedings of European Control Conference*, 1997, Brussels, Belgium, volume 6, (FR A B2).

4) Terugkoppeling, het grondprincipe van automatische regeling, komt men gewoonlijk in intelligente biologische systemen. Toch vertonen de tot nu ontwikkelde regelmethoden geen intelligentie in welke vorm dan ook.

   Feedback, the underlying principle of automatic control, is commonly encountered in intelligent biological systems. Yet the state-of-the-art control methods do not exhibit any form of intelligence.

5) Het is niet waarschijnlijk dat men door het bestuderen van de voornaamste literatuur over de regeltheorie besef krijgt van de problemen die zich voordoen bij een praktische implementatie. Alhoewel men door MATLAB[1] wordt gewezen op de details die essentieel zijn voor een toepassing, kan men pas echt inzicht krijgen door praktijkervaring.

   By studying the mainstream control theory, one is not likely to become aware of important implementation issues. While MATLAB[1] helps in understanding details that are essential for an application, real insight can only be gained through practical experience.

6) Een promotie-onderzoek duurt net lang genoeg om het gebruik van Latex te gaan waarderen. De overgang naar het bedrijfsleven impliceert echter het gebruik van Microsoft Word[2], want woorden verkopen beter dan formules.

   A Ph.D. project lasts just long enough to start enjoying Latex. The move to the industry, however, inevitably implies the use of Microsoft Word[2]. This is because words sell better than formulas.

7) De huidige trend in het Nederlandse bedrijfsleven is om technische activiteiten tot de lagere niveaus van de hierarchie te laten behoren, terwijl de beslissingen er over op hogere niveaus worden genomen, vaak op basis van niet-technische overwegingen, hetgeen de mensen die deze beslissingen uitvoeren frustreert.

   The current tendency in the Dutch industry is to keep technical activities in the lower levels of the hierarchy, while the decisions about these are made on the higher levels and are mainly based on non-technical issues, frustrating the people who carry out these decisions.

8) Bij de aanschaf van een Volkswagen krijgt men bijna dezelfde kwaliteit maar minder prestige dan bij de aanschaf van een Mercedes, hoewel men aanzienlijke minder betaalt. Bij de aanschaf van een Škoda krijgt men bijna dezelfde kwaliteit maar minder prestige dan bij die van een Volkswagen, hoewel men minder geld kwijt is. Prestige mag dus méér kosten dan kwaliteit in de hedendaagse levensstijl.

   By buying a Volkswagen, one gets almost the same quality and less prestige than with a Mercedes, but for a considerably lower price. By buying a Škoda, one gets almost the same quality and less prestige than with a Volkswagen, but for a lower price. In today's lifestyle, prestige may cost more than quality.

9) Tijdens de overgang van een centraal-geregelde naar een markteconomie hebben alle post-communistische landen te maken gekregen met een niet-minimum fase gedrag. Bij sommige lijkt dit gedrag echter te leiden tot instabiliteit.

   In the transition from a regulated to a market economy, nonminimum-phase behavior has been experienced in all post-communist countries. In some of them, however, this nonminimum-phase behavior seems to be turning into instability.

10) De keuze van de laatste koning als minister-president en van een communist als president laat overduidelijk zien dat de politiek in Bulgarije niet gebaseerd is op programma's maar op geschikte gelegenheiden die zich voordoen.

    The election of the last king for Prime Minister and of a communist for president clearly shows that politics in Bulgaria are not driven by programs but by opportunities.

11) Een veel voorkomende misleidende veronderstelling is dat degene die de macht heeft overal bekwaam in moet zijn, terwijl het voor hem voldoende is om de juiste mensen in te zetten op het juiste moment.

    A common misleading assumption is that the one who rules has to be competent in each and every subject, while it is enough for him or her to activate the right people at the right moment.

12) Oorlog wordt gewoonlijk beschouwd als een berekend risico. Wanneer het de eigen familie betreft, rolt er echter een andere uitkomst uit de berekening.

    War is commonly considered as a calculated risk. However, when it comes to one's own family, the outcome of the calculation is different.

13) Terwijl het vermogen om te leren een gave is, is de bereidheid om te leren een keus. Het is de kunst om deze keuze vol te houden.

    While the ability to learn is a gift, the willingness to learn is a choice. The art is to keep the spirit alive.

---

1. MATLAB is a registered trade mark of The Mathworks Inc., Natick, MA.
2. Microsoft Word is a registered trade mark of Microsoft Corp. Inc.

Fuzzy Control

of

Multiple-Input Multiple-Output Processes

Stanimir Mollov

Fuzzy Control

of

Multi-Input Multi-Output Processes

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 3 december 2002 om 10:30 uur

door Stanimir MOLLOV

electronics and automation engineer, Technische Universiteit Sofia, Bulgarije
geboren te Sliven, Bulgarije

Dit proefschrift is goedgekeurd door de promotoren:
Prof. ir. H. B. Verbruggen
Prof. dr. R. Babuška, M.Sc.

Samenstelling promotiecommissie:

Rector Magnificus,	Technische Universiteit Delft, voorzitter
Prof. ir. H. B. Verbruggen,	Technische Universiteit Delft, promotor
Prof. dr. R. Babuška, M.Sc.	Technische Universiteit Delft, promotor
Prof. dr. ir. A. Ollero,	Universidad de Sevilla, Spanje
Ass. Prof. dr. ir. I. Kalaykov,	Örebro University, Zweden
Prof. dr. ir. J. Hellendoorn,	Technische Universiteit Delft
Dr. ir. U. Kaymak,	Erasmus Universiteit Rotterdam
Dr. ir. A. J. J. van den Boom,	Technische Universiteit Delft

This dissertation has been completed in partial fulfillment of the requirements of the Dutch Institute of Systems and Control DISC for graduate study.

ISBN 90-9016413-8

to Vania, Michail, and Simona

CONTENTS

1 INTRODUCTION

Industrial processes are complex multivariable systems that often have nonlinear and time-varying dynamics. Because more and more aspects must be included in the control design such as stringent requirements for quality control, adaptation to variations in the raw material, changing production aims, energy reduction and environmental constraints, the control design itself is getting more and more complicated. Therefore, the modeling and control of such multiple-input, multiple-output (MIMO) processes have long been research topics in academics and important issues in the industrial control practice.

Classical control methods have proven their applicability to many multivariable control problems in industry, however, nowadays there are situations where these methods cannot provide the required performance. Advanced control techniques have only been partially able to satisfy the imposed demands. Yet, there is a problem of different origin that influences the acceptance of any new technology by the practicing engineering community: factors such as "easy-to-understand" and "easy-to-support" are of dominant importance. These considerations motivate the development of unambiguous and simple-to-understand methods for control design to be applied for nonlinear multivariable systems.

This thesis addresses control issues in complex, nonlinear, or partially unknown MIMO processes by means of techniques that are based on fuzzy set theory and fuzzy logic. Methods for the design of control systems are proposed that are based on fuzzy models. The fuzzy model can be part of the control algorithm, can serve for analysis of the process, can give better insight, and can be used to improve the operation, moni-

toring and (fault) diagnosis. This approach, termed MIMO fuzzy control, should cope with processes that pose problems to conventional techniques due to nonlinearities, lack of precise knowledge, or undesired interactions between the inputs and outputs. Shortly, the MIMO fuzzy control approach should present a *user-friendly* way of designing *nonlinear* controllers for MIMO processes.

1.1 MIMO aspects in process control

Although the need for control can have different origins, its major aim is to change the dynamics of the controlled system to match some desired behaviour within a domain of interest. Control is achieved through manipulation of one or more of the process inputs. When a single controlled variable (output) is operated by a single manipulated variable (control input), it is considered a single-loop process, regardless of the number of disturbing and uncontrolled variables present However, even the simplest processes require the control of at least *two* variables. If the goal is to control the torque produced by an engine, for example, one should control the amount of air and fuel, but also the moment when the spark is flashed (Chapter 6).

The multivariable control design should result either in m independent SISO controllers with proper decoupling or in a single multivariable controller that connects all available output variables with all available manipulated variables. Depending on the goals, different control configurations are possible. They can be classified in two broad groups: centralized and decentralized configurations. One important reason for decomposing the control system into a specific control configuration is that it may allow for simple tuning of the subcontrollers without the need for a detailed process model describing the complete dynamics and interactions. Multivariable controllers usually outperform decomposed (detached) controllers, but this gain in performance must be traded off against the cost of obtaining and maintaining a sufficiently detailed model. Below are briefly introduced a method of the decentralized group and a method of the centralized group which are further investigated in the following chapters: (i) decentralized control with decoupling design, and (ii) Model Predictive Control (MPC).

Decentralized control with decoupling design. Decentralized multivariable control is a desirable control structure because of its corresponding hardware and design simplicity (Morari and Zafiriou, 1989). An illustration of a fully decentralized structure for a linear two-by-two system is given in Fig. 1.1. In a decentralized control scheme, each input-output pairing is controlled by individual controllers

$$\mathbf{C}(z) = \text{diag}\left(c_1(z), c_2(z), \ldots, c_m(z)\right),$$

designed on the basis of a system $\widehat{\mathbf{P}}(z)$ consisting of the diagonal elements of a full multivariable model $\mathbf{P}(z)$

$$\widehat{\mathbf{P}}(z) = \text{diag}\left(p_1(z), p_2(z), \ldots, p_m(z)\right).$$

Different methods can be used to design the controllers for the separate channels of the decentralized control system. While PID controllers have mainly been used within

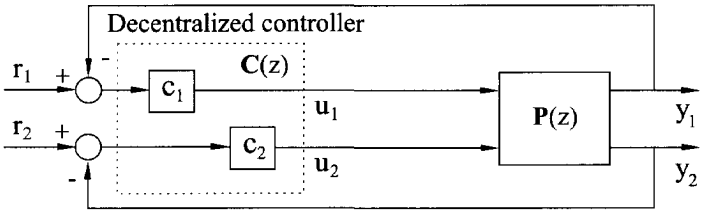


Figure 1.1. Fully decentralized multivariable control structure.

a decentralized control structure (Rivera et al., 1986; Shinskey, 1996), others, such as output feedback controllers, can be also applied (Morari and Zafiriou, 1989).

A fully decentralized control system ignores the effects of interactions in the process. However, this can severely influence the control system performance. A major challenge in the design of control systems for industrial processes is developing a structure that minimizes these interactions, because:

1. In most processes it is not possible to connect pairs of manipulated and controlled variables into (detached) loops that will not interact at all. Nevertheless, usually there is a "best" pairing depending on how well the loops can be paired and at which working points. Determining this pairing is essential for achieving the desired performance.

2. Interaction tends to be so prevalent that even the "best" pairing and "best" configuration of detached loops is a compromise which may fail to give acceptable dynamic performance. In such cases, substantial improvement can be gained by coordinating the variables, where the natural interactions in the process must be taken into account.

The performance of a decentralized control system can often be significantly improved by adding between the controller and the process a decoupler that compensates for the undesirable effects of interactions in the process (Hui, 1983). An illustration of a decentralized structure with a decoupler is shown in Fig. 1.2. The nominal closed-loop transfer function for a control scheme with a decoupler (with transfer matrix $\mathbf{D}(z)$) is

$$\mathbf{y}(z) = \big(\mathbf{I}+\mathbf{P}(z)\mathbf{D}(z)\mathbf{C}(z)\big)^{-1}\big(\mathbf{P}(z)\mathbf{D}(z)\mathbf{C}(z)\mathbf{r}\big).$$

An ideal decoupler $\mathbf{D}(z)$ is

$$\mathbf{D}(z) = \mathbf{P}^{-1}(z)\widehat{\mathbf{P}}(z), \tag{1.1}$$

where $\widehat{\mathbf{P}}(z) = \text{diag}\,(\widetilde{\mathbf{P}}(z))$ results in a diagonal loop transfer matrix $\widetilde{\mathbf{P}}(z)\mathbf{D}(z)\mathbf{C}(z)$; each process input-output combination can be handled by an independent design.

While it is sometimes difficult or impossible to accomplish perfect dynamic compensation of the form (1.1) because such a decoupler may not be physically realizable, e.g., it may have improper transfer functions, one can always carry out steady-state decoupling that eliminates the steady-state interactions. The steady-state decoupler is given by

$$\mathbf{D}(1) = \lim_{z\to 1}\Big[\widetilde{\mathbf{P}}(z)^{-1}\widehat{\mathbf{P}}(z)\Big] = \widetilde{\mathbf{P}}(1)^{-1}\,\text{diag}\,\big(\widetilde{\mathbf{P}}(1)\big).$$

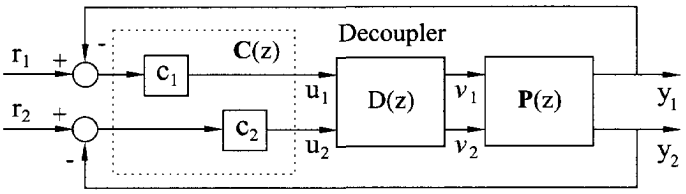


Figure 1.2. Decentralized multivariable control structure with decoupler.

Such a steady-state decoupler is capable of reducing dynamic interactions up to a certain bandwidth. Even so, there are still dynamic interactions which could cause single-loop controllers to counteract if tuned too tightly (as shown in Section 3.4).

To design a decoupler when the process **P** is *a nonlinear* multivariable process, one can either use tools from the differential geometry theory[1] to obtain a global nonlinear decoupler, or design local linear decouplers for the different operating regions. Both alternatives are explored in Chapter 3.

Model predictive control. Model Predictive Control (MPC) is a control method in which the control action is obtained by solving on-line, at each sampling instant, a finite (or infinite) horizon open-loop optimal control problem, using the current states as a starting point[2]. The optimization yields an optimal control sequence and the first component in this sequence is applied to the plant. MPC is one of the few advanced control design technologies that has a significant impact on industrial control problems; a major reason for its success is its ability to simply and effectively handle constraints on control and state signals (Mayne, 2001).

MPC is not a new control design method; it essentially solves standard optimal control problems. The main difference from other controllers is that MPC solves the optimal control problem on-line for the current status of the process, which is a mathematical programming problem, rather than using a pre-computed feedback control law. Determining a feedback law requires a solution of the Hamilton-Jacobi-Bellman (Dynamic Programming) differential or difference equation (Bryson and Ho, 1981; Lewis, 1995), an incomparably more difficult task (except in cases as H_2 or H_∞ linear optimal control, where the cost function can be finitely parameterized (Zhou et al., 1995)). In this sense, MPC differs from other control methods merely in the implementation. The requirement that the open-loop optimal control problem is solvable in a reasonable time (related to process dynamics) necessitates, however, the use of a finite horizon and this raises interesting problems.

[1] The reader interested in a detailed treatment of the differential geometry theory and its application to the nonlinear control design should refer to the trendsetting books by Isidori (1995), Nijmeijer and van der Schaft (1990) and Marino and Tomei (1995), and in the references therein.

[2] We are not interested in the trivial case of a Linear Time-Invariant (LTI) model, quadratic cost function and no constraints for which the control law can be computed off-line.

Suppose that the process being controlled has input $\mathbf{u}(k)$, state $\mathbf{x}(k)$ and output $\mathbf{y}(k)$, and has a nonlinear behaviour governed by the vector difference equations

$$\mathbf{x}(k+1) = \mathbf{f}\big(\mathbf{x}(k), \mathbf{u}(k)\big)$$
$$\mathbf{y}(k) = \mathbf{h}\big(\mathbf{x}(k)\big).$$

The control action is based on optimization of a given cost function

$$\min_{\mathbf{u}} J = \sum_{i=k}^{k+H_\mathrm{p}} l\big(\hat{\mathbf{y}}(i), \mathbf{r}(i), \hat{\mathbf{u}}(i), i\big) + F\big(\hat{\mathbf{y}}(k+H_\mathrm{p})\big),$$

where $l\big(\hat{\mathbf{y}}(i), \mathbf{r}(i), \hat{\mathbf{u}}(i), i\big)$ is a non-negative definite function of the model prediction $\hat{\mathbf{y}}(i)$, desired output $\mathbf{r}(i)$ and the predicted control input $\hat{\mathbf{u}}(i)$, defined over a prediction horizon H_p (often also referred to as a receding horizon). The second term $F\big(\hat{\mathbf{y}}(k+H_\mathrm{p})\big)$ is a non-negative definite function of the final predicted value, known as a terminal cost function. The process state, output and the control input are constrained within some spaces $\mathbf{x} \in X$, $\mathbf{y} \in Y$ and $\mathbf{u} \in U$, respectively.

This is a very general formulation, which can represent virtually any problem by using suitable functions $\mathbf{f}$, F, $\mathbf{h}$ and l. With the aid of the cost functions l and F and the constraining spaces X, Y and U, we can impose properties such as stability, robustness and (sub)optimality on the resulting control system (Mayne et al., 2000; Mayne, 2001). Stability is usually guaranteed when the terminal cost function F is used, terminal equality constraints (or constraint set) are imposed on the final predicted value $\hat{\mathbf{y}}(k+H_\mathrm{p})$ or a combination of the two. Robustness with respect to the uncertainty in the process description and model-plant mismatch can be achieved by selecting the suitable prediction horizon and constraining the control signal, or by considering all possible realizations of the uncertainty (min-max optimal control problem). When the process is linear, the cost function quadratic and the constraints linear, the optimal control problem reduces to a Quadratic Programme (QP) which can be efficiently solved and which yields a global solution to the optimal control problem. However, when the process is nonlinear, the optimal control problem is non-convex and conventional nonlinear programming algorithms will generally yield local (rather than global) solutions, requiring excessive computational time.

The question is then what happens to the properties of MPC if the global solution to the optimal control problem is not available. It was shown by Michalska and Mayne (1993) and Scokaert et al. (1999) that optimality is not predominantly required and, for example, that feasibility rather than optimality suffices for guaranteeing stability. This result allows satisfactory control when optimality is impractical to obtain. Possible strategies are to attempt to find an optimal solution to the optimal control problem and to cease when the time limit is reached, or to solve a simpler version of the optimal control problem. In the Chapter 4, the latter approach is used to obtain a "close-to-the-optimal" solution in a reasonable amount of time. Moreover, the "simplified" control problem can easily be extended to guarantee robustness of the control system with respect to model-plant mismatch, as shown in Chapter 5.

1.2 Fuzzy logic in control

During the past three decades, Fuzzy Logic Control (FLC) has emerged as one of the most active areas in the application of fuzzy set theory and fuzzy logic. The pioneering research of Mamdani on fuzzy control (Mamdani, 1974, 1977; Mamdani and Assilian, 1975) was motivated by Zadeh's seminal papers on the linguistic approach and system analysis based on the theory of fuzzy sets (Zadeh, 1965, 1968, 1973, 1975). Successful applications of fuzzy control in water quality control, automatic train operation systems (Palm and Storjohann, 1997), container crane operation systems, elevator control, automobile transmission control (Titli and Boverie, 1997), and nuclear reactor control (Na, 1998) have shown the potentiality of fuzzy logic to control ill-defined and/or time-varying multivariable processes.

Originally, the objective of fuzzy logic control was to control complex processes using knowledge-based control strategies emulating human reasoning. This type of control is applied when an adequate model of the process is not available, is not possible to obtain, or is too complicated to be used for control purposes. Nevertheless, humans are able to control complex systems, e.g., driving a car, without any formal model. Thus, the design of FLC can be based on empirical knowledge, rather than on a strictly analytic description.

Fuzzy control systems have been recognized as an appealing alternative to classical control schemes when partially known, nonlinear processes are addressed. Still, it was admitted in the fuzzy control literature that this (heuristic-based) approach to the design of fuzzy controllers is difficult to apply to multivariable control problems which represent the largest part of challenging industrial control applications (Palm et al., 1997). On the one hand, it lacks systematic and formally verifiable tuning techniques, while on the other hand stability, performance and robustness issues of the closed-loop control system can only be verified via extensive simulations.

In this thesis, an alternative approach for FLC design is used that follows closely the traditional design of a model-based control system. As in the non-fuzzy case, we start with a (fuzzy) model of the process to control, take into account the design specifications by means of performance criteria, and finally design the controller. To refer to this approach, the term *fuzzy model-based control* is used with the following meaning:

> *Given the model of the process under control and specifications of its desired behaviour, design a feedback control law, such that the closed loop system behaves in the desired way. In this setting, the process model is of a fuzzy type, while the design specifications and the developed controller are classical (non-fuzzy).*

Note that so defined, this approach does not belong to the trial-and-error framework, which makes the design of the fuzzy model-based controller more systematic and, in general, better suited for multivariable control applications.

Different methods have been proposed to design control systems based on fuzzy models. By exploiting the mathematical properties of particular fuzzy model structures, such as local linearity, fuzzy controllers were developed which are closely related to traditional gain scheduling approaches in feedback control and to multiple

model adaptive control. Johansen (1994b) presented a feedback linearizing controller based on a fuzzy model. The same author addressed stability, robustness and performance issues in fuzzy decoupled control (Johansen, 1994a). Considerable effort has been devoted to the study of techniques for designing state feedback and output feedback fuzzy controllers, based on an semidefinite programming tools such as linear matrix inequalities (Sugeno and Tanaka, 1991, 1992; Zhao, 1995; Tanaka et al., 1997; Zhao et al., 1997). Many successful applications of MPC using fuzzy prediction models have been reported, see (Saez and Cipriano, 1997; Kavsek et al., 1997; Fischer et al., 1998a; Fischer et al., 1998b; Hu and Rose, 1999; Nounou and Passino, 1999; Abonyi et al., 2001) for a nonexhaustive survey.

1.3 Research topics investigated in this thesis

The research presented in this thesis focuses on three important issues related to the design of fuzzy controllers for MIMO processes: (i) the analysis of interactions and input-output decoupling control in TS fuzzy models, (ii) the control of constrained multivariable processes for which accurate mathematical models are either not available or for which these cannot be used for control purposes, and (iii) the stability and robustness of the resulting control system.

1.3.1 Analysis of interactions and input-output decoupling in Takagi–Sugeno fuzzy models

To achieve a desired performance in multivariable process control, one must properly pair the manipulated variables with the corresponding controlled outputs and other measured signals. Undesired interactions are often present between the different control loops, which makes the control design complicated because a change in one input affects several different outputs. This problem is even more pronounced when the process is nonlinear. Then it may appear, for example, that within a certain operating range no interaction is present, while in another one the interaction is strong. In the first case, detached SISO controllers may suffice, while in the second case a decoupler or a MIMO controller must be used.

The *Relative Gain Array* (RGA) concept has been widely used for linear multivariable processes as a measure of interactions, in order to provide the best possible input-output pairing (Bristol, 1966; Grossdidier et al., 1985; Hovd and Skogestad, 1992). The RGA gives a measure of the influence that a certain input has on a particular output, relative to other inputs acting on the process. When the model is nonlinear, however, it is only possible to compute the RGA locally, after linearizing the model around an operating point. The approach followed in Chapter 3 makes use of the structure of the TS fuzzy model to obtain a number of RGAs which can indicate the interactions in the model sufficiently well. Depending on the model structure, it is possible to analyze the interactions locally, computing a separate RGA for each rule. However, since the RGA for a point in between two (or more) rules is not a weighted sum of the rule's RGAs, the interactions have to be analyzed point-wise, combining the degrees of fulfillment for that point with the rule consequents.

The *output sensitivity* function indicates the dependence of a process output on variations in one or more of its inputs or states and can provide additional insight in the interaction. The sensitivity function is computed as the partial derivative of the output with respect to a given input, while the remaining inputs are kept constant.

Many applications of decoupled MIMO control have been described, where a separate fuzzy model is used as a decoupler (Yaochu et al., 1995; Reay et al., 1995; Kang and Lei, 1996). Instead, the techniques proposed in Section 3.3.2 decouple the fuzzy model of the process. The rule-base structure of the TS fuzzy models is explored to facilitate the decoupling design. Since at each sampling instant a linear model can be obtained from the fuzzy model, an analytic time-varying decoupler can be designed that corresponds to this linear model. As an alternative to the analytic solution, a decoupling law based on a numerical nonlinear optimization is presented as well.

1.3.2 Fuzzy model predictive control

Many successful applications of MPC using fuzzy prediction models have been reported (Saez and Cipriano, 1997; Kavsek et al., 1997; Fischer et al., 1998a; Fischer et al., 1998b; Hu and Rose, 1999; Abonyi et al., 2001), and recently several papers appeared in which different fuzzy MPC algorithms are analyzed and compared (Espinosa et al., 1999; Nounou and Passino, 1999; de Oliveira and Lemos, 2000). The methods discussed in these references can generally be classified into two groups: (i) methods utilizing directly the fuzzy model in the optimization procedure (Kacprzyk, 1997; Fischer et al., 1998a; Sousa, 1998; Nounou and Passino, 1999; Hu and Rose, 1999; de Oliveira and Lemos, 2000) and (ii) methods using a linearized model instead of the fuzzy one (Saez and Cipriano, 1997; Kavsek et al., 1997; Nounou and Passino, 1999; Roubos et al., 1999; Abonyi et al., 2001). For example, Kavsek et al. (1997) extract a step-response model from the fuzzy model, whereas Abonyi et al. (2001) apply Jacobian linearization. Other possibilities are to compute the control signals for the different fuzzy submodels separately and to weigh them (Huang et al., 2000), or to use only the submodel with the highest membership degree (Saez and Cipriano, 1997; Nounou and Passino, 1999).

The methods presented in these references, however, make a compromise between the accuracy of the model prediction (the second group), and the computational load (and hence the time) required to solve on-line the underlying optimization problem (the first group). The main problem of fuzzy model predictive control using a nonlinear fuzzy model in the optimization routine is that the convexity of the optimization problem is lost, hence time-consuming optimization is necessary, with no guarantee of finding an optimal solution in a real-time application. How long each optimization step will take, whether the optimization procedure will ever terminate and if so, on a local or global minimum, etc. is not clear. This hampers the application to fast processes, where iterative optimization techniques cannot be properly used for short sampling periods.

The methods presented in Chapter 4 offer an effective way for formulating the optimization problem by employing a single state-space local linear model or a set of such models that approximate the fuzzy model. The structure of the underlying optimiza-

tion problem is explored in order to arrive at a suboptimal solution, which is as close as possible to the optimum in a limited amount of time, and this also makes nonlinear MPC suitable for fast processes. Our approach is based on linear time-varying (LTV) prediction models derived by freezing the parameters of the fuzzy model at a given operating point or along a predicted with the fuzzy model trajectory. The control signal is obtained by solving a constrained quadratic programme (QP). To account for errors introduced by the linearization, an iterative optimization scheme is proposed. In such a setting, the QP solution provides a search direction toward the global minimum of the optimization problem. Convergence is guaranteed through a line search mechanism that considers reduction both in the cost function and in the constraint violation. The method belongs to the class of Sequential Quadratic Programming methods (Powell, 1978; Pshenichnyj, 1994), and is formulated with regard to the specific structure of the optimization problem in linear model predictive control. The advantage is its generic form, which does not depend on the structure of the fuzzy model currently used.

1.3.3 Robust stability constraints for fuzzy model predictive control

Although the TS model usually yields a reasonably accurate approximation of the process, one must keep in mind that a certain model-plant mismatch will always be present. The mismatch can be due to unmodeled dynamics, time-varying and/or aging phenomena, etc., and it will not only deteriorate the control performance but may even destabilize the closed-loop system. The availability of tools for the design of a robustly stable predictive controller is therefore of critical importance. So far, this aspect of MPC has only been addressed for LTI process models, typically by using techniques based on infinite prediction horizon (Rawlings and Muske, 1993; Scokaert, 1997), end-point or terminal set constraints (Keerthi and Gilbert, 1988; Clarke and Scattolini, 1991; Mosca and Zhang, 1992; Michalska and Mayne, 1993), or min-max optimization (Campo and Morari, 1987; Bemporad et al., 2001). In the linear case, if no constraints are specified, the MPC controller can be expressed as an LTI controller the robustness of which can easily be analyzed. When constraints are present, robust stability can be guaranteed either by including an explicit contraction constraint (Zeng and Morari, 1995) or by assuring that the criterion function is a contraction through the optimization of the maximum of the criterion function over all possible models (Zeng and Morari, 1993, 1994). The advantage of the latter method is that not only robust stability, but also robust performance is obtained. The disadvantages are the need of using polytopic uncertainty descriptions and the difficult min-max optimization that results. Recently Kothare et al. (1996) derived a method based on the same principle which circumvents all of these disadvantages, but it may become quite conservative. Unfortunately, all of these methods are only valid for linear models and no general framework for predictive control based on a nonlinear model is available.

In Chapter 5 we propose an extension of the method proposed by de Vries and van den Boom (1997), where conditions are given that guarantee robust asymptotic stability for open-loop stable linear systems with an additive ∞-norm bounded model uncertainty. Here similar conditions are derived for open-loop stable *non-linear* systems (possibly non-fuzzy) with an additive ∞-norm bounded model uncertainty. Based

on the uncertainty description, we derive level and rate constraints for the control signal that guarantee bounded-input bounded-output (BIBO) stability for any model-plant mismatch within given bounds. When the nonlinear model is a fuzzy model of the Takagi–Sugeno type, an estimate of the uncertainty bounds can be found. We use the fact that at each sampling instant, new measurements become available and consider the fuzzy model as linear time varying, rather than as nonlinear time invariant. The resulting constraints are similar to (small-gain-based) l_1-control theory, but they are much less conservative as they are based on the uncertainty that currently occurs in the system, instead of on the "allowed worst-case model uncertainty".

1.4 Outline of the thesis

First, a few words on the style used in this thesis. The thesis is based on a number of manuscripts that have been separately published or submitted for publication. We have tried to reduce the redundancy as much as possible, and more or less rewritten each chapter based on one or several publications. The notation within each chapter will be consistent, but there may be some minor inconsistencies in the notation used in the various chapters. However, the notation will be clearly indicated. The mathematical precision varies somewhat throughout the thesis. In particular, we do not always distinguish between a function and its values, or interchangeably use terms like *function* and *mapping*, or *multiple-input multiple-output* and *multivariable* (processes). Throughout the thesis we use representations and models expressed in discrete time, unless stated explicitly otherwise.

Second, this thesis only focuses on certain specific ideas and concepts and develops them into techniques that can be applied to industrial-scale multiple-input multiple-output processes. Readers interested in a detailed and fundamental treatment of fuzzy sets and fuzzy logic can consult research monographs by Pedrycz (1993), Driankov et al. (1993) or Yager and Filev (1994). A comprehensive exposition of the problems faced in MIMO process control systems and existing MIMO control techniques can be found in (Maciejowski, 1989; Zhou et al., 1995; Skogestad and Postlethwaite, 1996; Shinskey, 1996), and recently in the textbook by Goodwin et al. (2001).

Chapter 2 of this thesis contains the background material needed for understanding the sequel chapters. We introduce the Takagi–Sugeno (TS) type of fuzzy models (Takagi and Sugeno, 1985) which is used in the analysis and control methods developed in the thesis. TS fuzzy models can be derived either by using input-output process data, or through linearization of a nonlinear model at one or more operating regions. These models offer a convenient basis for state feedback and output feedback control design: local controllers can be separately developed for each of the local models, and then combined into a global one using again the fuzzy blending.

Thereafter, the chapter offers a short overview of the state feedback and output feedback designs which can be posed in terms of matrix inequalities (LMI). Although this approach has shown a strong potential, there are issues of different character that limit its applications, both in conceptual and implementation aspects. In this thesis, a different paradigm was followed in the development of the methods for fuzzy control design for MIMO processes.

Chapter 3 focuses on the analysis of the input-output interactions and decoupling in TS fuzzy models. Two new methods are proposed that exploit the specific model structure to indicate sufficiently well the interactions based on the Relative Gain Array and output sensitivity, respectively. Next, a decoupling strategy is discussed. Depending on the affineness of the model, an analytic or numeric decoupling law is provided on-line, at each sampling instant. A simulation example and a real-time example are presented.

Chapter 4 addresses the optimization problem in fuzzy model predictive control by using locally linear models obtained through linearization of the fuzzy model at the current operating point at each sampling instant. For predictive control with a long prediction horizon, the fuzzy model is successively linearized along the predicted input and output trajectories in order to obtain a more accurate process description. A SISO simulation example and a real-time MIMO example are presented.

Chapter 5 focuses on the robust stability properties of the fuzzy model predictive controller discussed in Chapter 4. Based on the small-gain theorem from the l_1-control theory, constraints on the control signal and its increment are derived that guarantee closed-loop robust asymptotic stability for open-loop BIBO stable processes with an additive l_1-norm bounded model uncertainty. An algorithm is presented that estimates the bounds on the model uncertainty when the process model is a TS fuzzy model. A SISO simulation example and a real-time MIMO example are given.

Chapter 6 presents a real-world application of the design method for fuzzy predictive control. The process under consideration is a simulation model of a vehicle using a gasoline direct injection (GDI) engine, which is a new concept that provides a solution for the reduction of fuel consumption and pollutant emissions. The complexity of this system exceeds most of the previously reported applications of MPC, mainly because of the switching of combustion modes and the related adaptation of the cost function and constraints. The implemented controller comprises a model predictive control optimizer and a switching logic used to provide smooth switching between the combustion modes. The prediction models are TS fuzzy models identified from input–output data. The control system performance is compared with the performance achieved by two other control strategies.

Chapter 7 demonstrates the advantages of using fuzzy modeling and control design techniques on a simulation model of a distillation column. The column operates in a specific ("LV") configuration, for which there are two manipulated variables and two controlled variables. A closed-loop identification experiment is presented that allows for identification of properties that are important for control and difficult to obtain through an open-loop identification setup. Data from the closed-loop experiment is used for identification of a TS fuzzy model. After identification this model is used for analysis of the existing input-output interactions and for decoupling design. Next the fuzzy model is utilized in a fuzzy MPC algorithm in which linear models are derived from the fuzzy model at the current point and used to construct the optimization problem. To illustrate the influence of the model prediction on the achieved performance, we use a different fuzzy model that has as input a disturbance signal in the optimization algorithm. Finally, the robustifying effect of the stability constraints is shown.

In the last chapter, Chapter 8, the main conclusions are drawn, and some critical remarks are made with respect to the approaches developed. Suggestions for future research are given.

Appendix A presents data-driven methods used to derive input-output TS fuzzy models, which are used throughout the thesis. Appendix B introduces the concept of right invertibility, used in Chapter 3 for the decoupling design. In Appendix C, a Sequential Quadratic Programming (SQP) algorithm is presented as it is the closest to the optimization methods introduced in Chapter 4. A method for fuzzy model linearization with application in model predictive control is outlined in Appendix D. Appendix E summarizes some of $\mathcal{L}_1$-control theory concepts used in Chapter 5. Appendix F gives a short introduction to the basic principles of operation of the distillation columns. Appendix G contains a list of mathematical symbols and abbreviations.

The main contributions of this thesis to the field of fuzzy MIMO control design are presented in Chapter 3 through Chapter 5. The analysis of input-output coupling, and the consequent decoupling design are presented in Chapter 3. Chapter 4 focuses on the optimization problem in fuzzy model predictive control. Chapter 5 derives constraints on the control input that guarantee closed-loop robust asymptotic stability for open-loop BIBO stable processes with an additive l_1-norm bounded model uncertainty. Chapters 6 and 7 give comprehensive descriptions and solutions to two industrial applications.

2 TAKAGI–SUGENO FUZZY MODELS AND FEEDBACK CONTROL DESIGN

This chapter introduces the TS fuzzy model which is used for analysis and control design throughout the thesis. Local controllers and observers can be developed for each of the TS rules and then combined to obtain a global control strategy. Recently much research has been focused on state feedback and output feedback control design by means of optimization problems formulated in terms of linear matrix inequalities (LMIs). We briefly review the main features of this approach and show some of its shortcomings. The methods proposed in the subsequent chapters follow a different paradigm.

2.1 Motivation

Industrial processes are complicated multivariable systems with a complex behaviour which can be described by using either a single, usually complicated globally valid model, or a collection of locally valid, usually simpler models. We can fulfill the requirements imposed on the model for model accuracy, robustness and performance by using either of these approaches. However, while global models have desired properties such as limited storage demands and fast execution times even for MIMO cases, modeling complex MIMO systems with many interactions often requires the adaptation of global nonlinear models, demanding identification procedures which are typically slow and analytically intractable. Additional problems are that it is hard to select the best model structure in a nonlinear MIMO case and to develop validation methods that assess a global model when one only has a finite amount of noisy measurements as a basis.

The above considerations suggest that in many situations a set of local models may be preferable to a single global model. During the past two decades, alternatives to the global modeling techniques have emerged that divide the total operating range of the process into operation regions for which linear models can be derived using simple estimating techniques, see (Johansen, 1994b) for an extensive discussion. Additionally, the controller based on a set of local linear models usually has a structure that is easy to understand and interpret and a reduced computational complexity compared to controllers using a global nonlinear model.

In this thesis, fuzzy models of the Takagi and Sugeno (1985) (TS) type are used to represent the process under consideration as they are well suited for complex multivariable systems. The TS fuzzy models have more in common with the traditional nonfuzzy models used in model-based control than the relational and the linguistic fuzzy models. With TS fuzzy models, the operating regime of the process is represented by overlapping fuzzy sets wherein the process can be represented with sufficient accuracy by a number of linear models with a bias term. Next, an inference mechanism is applied to combine the local linear models in order to construct a nonlinear model of the process.

If a process model is available in the form of a rule-based fuzzy model, it can be used in the control design. We can design a linear control law for each of the local linear models that correspond to the individual TS rules. In such a setup, the control policy is expressed by a set of control rules and the controller operates in a "region by region" manner. Each control rule can be viewed as a local controller, that is valid in a certain operating region.

When the individual rules represent linear systems, one can design the corresponding controllers and observers by using tools of linear feedback control theory. Because linear TS models (i.e., with a zero offset term) can be embedded in the general class of discrete polytopic linear differential inclusions, fuzzy controller synthesis can be expressed in terms of linear matrix inequalities (LMI). For a thorough discussion on the use of LMI in the control theory, see Boyd et al. (1994). By using state feedback and output feedback design we obtain fuzzy controllers that meet the design specifications (Kothare et al., 1996; Zhao et al., 1997; Tanaka et al., 1998; Bergsten, 2001).

Robustness of the control system with respect to uncertainties in the model parameters can be guaranteed, as shown by Zhao et al. (1997). The synthesis of a fuzzy observer is posed as a problem dual to the control design.

The idea of controlling complex systems by means of local controllers is not new. Initially it was investigated within the gain-scheduling framework (Stein, 1980; Sain and Yurkovich, 1982). Gain scheduling is "a nonlinear feedback approach of a special type; it has a linear regulator whose parameters are changed as a function of operating conditions in a preprogrammed manner" (Åström and Wittenmark, 1989a). In other words, it is a linear parameter-varying feedback controller whose parameters are modified as a function of operation conditions. Traditionally, gain scheduling has been the most common systematic approach to the control of strongly nonlinear systems in practice (Whatley and Pott, 1984; Åström and Wittenmark, 1989a; Shamma and Athans, 1990, 1991; Rugh, 1991; Reichert, 1992; Nichols et al., 1993).

A gain-scheduling control system consists of two major components: (i) a family of controllers and (ii) a scheduler that at each sampling instant engages a controller (or a combination of controllers) to be applied to the process. Gain scheduling is a quite general class of control structures characterized by multiple local controllers, including as a subclass the state feedback and output feedback controllers based on TS fuzzy models.

2.2 Takagi–Sugeno fuzzy models

The structure of the Takagi–Sugeno (TS) fuzzy model and the related inference mechanism are shortly presented in this section.

TS fuzzy models are a particular type of rule-based fuzzy models that use **If - then** rules and logical connectives to represent the relationships between variables

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

The antecedent proposition is in the form "X is $\mathcal{A}$" where X is a linguistic variable and $\mathcal{A}$ is a linguistic label (also called linguistic value or linguistic term), defined by a fuzzy set on the universe of discourse of the linguistic variable (Klir and Folger, 1988; Zimmermann, 1991). The consequent proposition is a mathematical function of the model inputs rather than a fuzzy statement.

These models are suitable for modeling a large class of nonlinear multivariable systems. Consider a MIMO system with m inputs $\mathbf{u} \in U \subset \mathbb{R}^m$ and p outputs $\mathbf{y} \in Y \subset \mathbb{R}^p$. Depending on the available prior knowledge, this system can be approximated by a collection of MIMO state-space fuzzy models, or by a collection of coupled input-output multiple-input, single-output (MISO) fuzzy models.

2.2.1 State-space TS models

If the process nonlinearity is such that in a given region all outputs depend uniformly on a certain combination of the antecedent variables, then the TS model can be stated

in the following MIMO state-space form

$$\mathcal{R}_i: \quad \textbf{If } z_1(k) \text{ is } \mathcal{A}_{i,1} \textbf{ and } \ldots \textbf{ and } z_n(k) \text{ is } \mathcal{A}_{i,n} \textbf{ then} \tag{2.1}$$
$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{u}(k)+\mathbf{a}_i \\ \mathbf{y}(k) &= \mathbf{C}_i\mathbf{x}(k)+\mathbf{c}_i \end{cases}$$
$$i = 1,2,\ldots,K$$

where K is the number of rules, $M(i) = \{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{a}_i, \mathbf{c}_i\}$, $i = 1,\ldots,K$ is the ith local affine model, $\mathbf{x}(k) \in \mathbb{R}^\rho$ is the state and $\mathbf{u}(k) \in \mathbb{R}^m$ and $\mathbf{y}(k) \in \mathbb{R}^p$ are the input and output vectors respectively, and $\mathcal{A}_{i,j}$, $j = 1,\ldots,n$ are (one-dimensional) antecedent fuzzy sets defined on some scheduled variables $\mathbf{z}(k) \in \mathbb{R}^n$, respectively. Note that $\mathbf{z}(k)$ can contain model states and inputs.

Since all the locally linear time-invariant (LTI) models in the consequent part of the fuzzy model have an identical structure, the fuzzy model can be regarded as a (quasi)linear time-varying dynamic system, with LTI models valid only in the working range specified in the antecedent part of the fuzzy rules. In this framework, the fuzzy rules and the membership functions for the antecedent variables often have a relevant physical interpretation.

2.2.2 Input-output TS models

An input-output TS model comprises a collection of coupled MISO models of the input-output NARX type

$$\mathbf{y}_l(k+1) = \mathcal{R}_l\big(\mathbf{x}_l(k), \mathbf{u}(k)\big), \quad l = 1,2,\ldots,p. \tag{2.2}$$

The vector $\mathbf{u}(k) \in \mathbb{R}^m$ contains the current inputs and the vector $\mathbf{x}_l(k) \in \mathbb{R}^\rho$ contains the current and delayed outputs and delayed inputs

$$\mathbf{x}_l(k) = \left[\mathbf{y}_1^T(k), \ldots, \mathbf{y}_p^T(k), \mathbf{u}_1^T(k-1), \ldots, \mathbf{u}_m^T(k-1)\right]^T,$$

where

$$\mathbf{y}_i(k) = \left[y_i(k), y_i(k-1), \ldots, y_i(k-n_{y,i})\right]^T, \quad i = 1,\ldots,p$$
$$\mathbf{u}_j(k-1) = \left[u_j(k-1), \ldots, u_j(k-n_{u,j})\right]^T, \quad j = 1,\ldots,m.$$

The indices $n_{y,i}$ and $n_{u,j}$ specify the number of lagged values for the ith output and the jth input, respectively. Sometimes it is more convenient to combine $\mathbf{u}(k)$ and $\mathbf{x}_l(k)$:

$$\boldsymbol{\chi}_l(k) = \left[\mathbf{x}_l^T(k), \mathbf{u}^T(k)\right]^T, \quad \boldsymbol{\chi}_l \in \mathbb{R}^\vartheta, \vartheta = \rho + m. \tag{2.3}$$

The functions $\mathcal{R}_l$ in (2.2) can be parameterized in the following rule-base form (Takagi and Sugeno, 1985)

$$\mathcal{R}_{li}: \quad \textbf{If } x_{l1}(k) \text{ is } \mathcal{A}_{li,1} \textbf{ and } \ldots \textbf{ and } x_{l\rho}(k) \text{ is } \mathcal{A}_{li,\rho} \textbf{ and} \tag{2.4}$$
$$u_1(k) \text{ is } \mathcal{A}_{li,\rho+1} \textbf{ and } \ldots \textbf{ and } u_m(k) \text{ is } \mathcal{A}_{li,\rho+m}$$
$$\textbf{then } y_{li}(k+1) = \boldsymbol{\zeta}_{li}\mathbf{x}_l(k) + \boldsymbol{\eta}_{li}\mathbf{u}(k) + \theta_{li}$$
$$i = 1,2,\ldots,K_l$$

where $\mathcal{A}_{li,h}$ are antecedent membership functions of the ith rule

$$\mu_{\mathcal{A}_{li,h}}(\chi_{lh})\colon \mathbb{R} \to [0,1], \quad h = 1,\ldots,\vartheta, \tag{2.5}$$

and $\boldsymbol{\zeta}_{li}$ and $\boldsymbol{\eta}_{li}$ are vectors containing the consequent parameters, and θ_{li} contains the offsets, respectively. K_l is the number of rules for the lth output. The model output is computed as the weighted average of the individual rules' consequents

$$y_l(k+1) = \frac{\sum_{i=1}^{K_l} \beta_{li}\big(\boldsymbol{\chi}_l(k)\big)\big(\boldsymbol{\zeta}_{li}\mathbf{x}_l(k) + \boldsymbol{\eta}_{li}\mathbf{u}(k) + \theta_{li}\big)}{\sum_{i=1}^{K_l} \beta_{li}\big(\boldsymbol{\chi}_l(k)\big)}. \tag{2.6}$$

The degree of fulfillment of the ith rule $\beta_{li}\big(\boldsymbol{\chi}_l(k)\big)$ is obtained as the product of the membership degrees of the antecedent variables in that rule (recall (2.4))

$$\beta_{li}\big(\boldsymbol{\chi}_l(k)\big) = \prod_{h=1}^{\vartheta} \mu_{\mathcal{A}_{li,h}}(\chi_{lh}). \tag{2.7}$$

Here the product operator is used, as it ensures a smooth degree of fulfillment.

The TS models in the state-space form are useful when the process nonlinearity is such that in a given region all outputs depend uniformly on a certain combination of the antecedent variables. If this is not the case, then it is not possible to aggregate the outputs in the form (2.1). In such situations, input-output TS fuzzy models are more suitable. It can be stated that the state-space form is better suited to accommodate linear models obtained through linearization of a nonlinear process model already obtained, while the input-output form is preferable for modeling of a process solely based on process data. In this thesis, the input-output form of the TS model is used and for the sake of simplicity the term "input-output" is dropped henceforth, unless stated explicitly otherwise.

2.2.3 Inference mechanism

The inference in the TS model is reduced to the algebraic expression (2.6). By denoting the normalized degree of fulfillment

$$w_{li}\big(\boldsymbol{\chi}_l(k)\big) = \frac{\beta_{li}\big(\boldsymbol{\chi}_l(k)\big)}{\sum_{j=1}^{K_l} \beta_{lj}\big(\boldsymbol{\chi}_l(k)\big)}, \tag{2.8}$$

the TS model can be expressed as a linear time-varying (LTV) model:

$$\begin{aligned} y_l &= \left(\sum_{i=1}^{K_l} w_{li}\big(\boldsymbol{\chi}_l(k)\big)\boldsymbol{\zeta}_{li}^T\right)\mathbf{x}_l(k) + \left(\sum_{i=1}^{K_l} w_{li}\big(\boldsymbol{\chi}_l(k)\big)\boldsymbol{\eta}_{li}^T\right)\mathbf{u}(k) + \sum_{i=1}^{K_l} w_{li}\big(\boldsymbol{\chi}_l(k)\big)\theta_{li} \\ &= \boldsymbol{\zeta}_l^T\big(\boldsymbol{\chi}_l(k)\big)\mathbf{x}_l(k) + \boldsymbol{\eta}_l^T\big(\boldsymbol{\chi}_l(k)\big)\mathbf{u}(k) + \theta_l\big(\boldsymbol{\chi}_l(k)\big), \qquad l = 1,\ldots,p \end{aligned} \tag{2.9}$$

where the parameters $\boldsymbol{\zeta}_l(\boldsymbol{\chi}_l(k))$, $\boldsymbol{\eta}_l(\boldsymbol{\chi}_l(k))$ and $\theta_l(\boldsymbol{\chi}_l(k))$ are linear combinations of the consequent parameters $\boldsymbol{\zeta}_{li}$, $\boldsymbol{\eta}_{li}$ and θ_{li}:

$$\begin{aligned}\boldsymbol{\zeta}_l(\boldsymbol{\chi}_l(k)) &= \sum_{i=1}^{K_l} w_{li}(\boldsymbol{\chi}_l(k))\boldsymbol{\zeta}_{li}, \\ \boldsymbol{\eta}_l(\boldsymbol{\chi}_l(k)) &= \sum_{i=1}^{K_l} w_{li}(\boldsymbol{\chi}_l(k))\boldsymbol{\eta}_{li}, \\ \theta_l(\boldsymbol{\chi}_l(k)) &= \sum_{i=1}^{K_l} w_{li}(\boldsymbol{\chi}_l(k))\theta_{li}.\end{aligned} \tag{2.10}$$

In other words, $\boldsymbol{\zeta}_l(\boldsymbol{\chi}_l(k))$, $\boldsymbol{\eta}_l(\boldsymbol{\chi}_l(k))$, $\theta_l(\boldsymbol{\chi}_l(k))$ are bounded to lie within the convex hulls with vertices $\boldsymbol{\zeta}_{li}, \boldsymbol{\eta}_{li}, \theta_{li}$, respectively. This property makes possible the use of the framework of *polytopic systems* (Boyd et al., 1994) for the control design. For TS models given in the state-space form, methods have been developed to design controllers with desired closed-loop characteristics and to analyze their stability. These methods are briefly summarized in the remaining sections of this chapter, as no additional research in this direction had yet been carried out.

2.2.4 Constructing TS fuzzy models

The TS fuzzy models are viewed as a class of integrated local linear models, used to represent the process under consideration by decomposing it into a number of simpler subsystems. The fuzzy sets and fuzzy logic theory offers an excellent tool for representing the overlap associated with the decomposition task, for providing smooth transitions between the individual local models, and for incorporating various types of knowledge in one common medium.

There are two common ways to derive TS fuzzy models. The first way uses a local linearization of an available nonlinear model, usually resulting in a state-space TS model. The second way is based on experimental input-output process data and is more suitable for constructing input-output TS models (Babuška, 1998). A class of fuzzy clustering algorithms for automated generation of input-output TS fuzzy models is outlined in Appendix A. No additional research concerning the subject of fuzzy modeling and identification has been carried out in this thesis.

2.3 Stability of fuzzy systems

Various methods have been proposed to analyse the stability of fuzzy control systems, e.g., the bifurcation method, methods based on the hyperstability theory, the Lyapunov stability theory and the input-output stability theory (Piegat, 2001). In this thesis, we elaborate on the input-output stability theory (Chapter 5), which is used to guarantee the local bounded-input bounded-output (BIBO) stability of the fuzzy predictive controller designed in Chapter 4. Cuesta et al. (1999) showed that input-output stability techniques, such as the conicity criterion and the harmonic balance method, in combination with frequency-domain methods can be applied to multivariable systems, and that in some cases they are superior to the Lyapunov method. The Lyapunov method,

however, is favorable for carrying out a global stability analysis of the fuzzy state and output feedback controllers. In the following we briefly review it. For the unforced part of (2.1), i.e., without inputs and offset terms,

$$\mathbf{x}(k+1) = \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\,\mathbf{A}_i\mathbf{x}(k), \tag{2.11}$$

a sufficient condition for stability is stated as follows (Tanaka and Sugeno, 1992). The equilibrium at zero of (2.11) is globally asymptotically stable if there exists a positive definite matrix $\mathbf{P} = \mathbf{P}^T$ such that

$$\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P} < \mathbf{0}, \quad i = 1,\ldots,K, \tag{2.12}$$

i.e., a common $\mathbf{P}$ has to exist for all subsystems. For $K = 1$, this condition reduces to the Lyapunov stability theorem for LTI systems.

If there exists $\mathbf{P} > \mathbf{0}$ for which (2.12) holds, the system (2.11) is called quadratically stable and

$$V(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x} \tag{2.13}$$

is the corresponding quadratic Lyapunov function.

Note that, in general, the system (2.11) is not stable even if all its subsystems (i.e., all $\mathbf{A}_i$s) are stable. An illustrative example can be found, among others in Wang et al. (1996). In the following, control design methods are reviewed that provide feedback controllers and observers that render the closed-loop system stable. Note also that the presented stability conditions rely on a single quadratic Lyapunov function that may be conservative and for certain systems may lead to infeasible stability conditions. Recently, extensions to the stability analysis and synthesis have been proposed that are based on piecewise quadratic Lyapunov functions, see among others (Rantzer and Johansson, 1997; Johansson et al., 1999).

2.4 Fuzzy state feedback controllers

For simplicity, let us begin with a fuzzy TS model with *linear* rather that affine local subsystems, i.e., in (2.1) $\mathbf{a}_i = 0$ and $\mathbf{c}_i = 0$ for $i = 1,\ldots,K$. The control design is carried out based on the fuzzy model as follows. First, for each local linear model, a linear feedback controller is designed (Fig. 2.1). Then the overall nonlinear controller is obtained as a fuzzy blending of the individual linear controllers

$$\mathcal{R}_i\colon\ \textbf{If } z_1(k) \text{ is } \mathcal{A}_{i,1} \textbf{ and } \ldots \textbf{ and } z_n(k) \text{ is } \mathcal{A}_{i,n} \textbf{ then } \mathbf{u}(k) = -\mathbf{F}_i\mathbf{x}(k) \qquad i = 1,2,\ldots,K. \tag{2.14}$$

Note that here the antecedent variables $z_i(k)$, $i = 1,\ldots,n$ do not contain the current input $\mathbf{u}(k)$ as it is computed by the feedback law in the rule consequent. Since the fuzzy controller has the same antecedents as the fuzzy model (2.1), this implies that also in the fuzzy model the antecedent variables cannot comprise the current input.

The controller output is a weighted average of the individual rules' contributions

$$\mathbf{u}(k) = \frac{-\sum_{i=1}^{K}\beta_i\big(\mathbf{z}(k)\big)\mathbf{F}_i\mathbf{x}(k)}{\sum_{i=1}^{K}\beta_i\big(\mathbf{z}(k)\big)} = -\sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\mathbf{F}_i\mathbf{x}(k) = -\mathbf{F}\big(\mathbf{z}(k)\big)\mathbf{x}(k). \tag{2.15}$$

This is the so-called parallel distributed compensation scheme, proposed by Tanaka and Sano (1994).

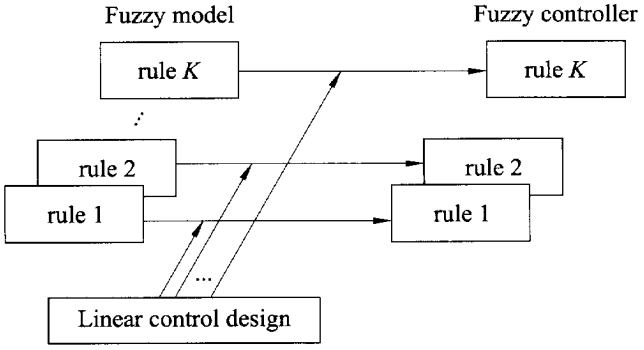


Figure 2.1. Fuzzy state feedback control design.

Substituting (2.15) into (2.1), the closed-loop system is given by

$$\begin{aligned}\mathbf{x}(k+1) &= \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)\big(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j\big)\mathbf{x}(k) \\ &= \sum_{i=1}^{K} w_i\big(\mathbf{x}(k)\big)w_i\big(\mathbf{z}(k)\big)\mathbf{G}_{ii}\mathbf{x}(k) \\ &\quad + 2\sum_{i,j,i<j}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)\left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)\mathbf{x}(k), \qquad (2.16)\end{aligned}$$

where

$$\mathbf{G}_{ij} = \mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j.$$

A generalization of the stability condition (2.12) is that the equilibrium of the closed-loop system (2.16) is globally asymptotically stable if there exists a common positive definite matrix $\mathbf{P}$ such that

$$\mathbf{G}_{ij}^T\mathbf{P}\mathbf{G}_{ij} - \mathbf{P} < 0, \qquad (2.17)$$

for $i, j = 1, \ldots, K$ except for the pairs (i, j), for which $w_j\big(\mathbf{z}(k)\big) = 0, \forall k$.

The above inequalities take into account the influence of all the controllers on a single rule of the model. By considering the closed-loop system that contains model rule i and controller rule i, $i = 1, \ldots, K$ separately from closed-loop systems based on model rule i and controller rule j, $i \neq j$, we obtain the sufficient stability condition

$$\begin{aligned}&\mathbf{G}_{ii}^T\mathbf{P}\mathbf{G}_{ii} - \mathbf{P} < \mathbf{0}, \quad i = 1, \ldots, K, \qquad (2.18)\\ &\left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T\mathbf{P}\left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T - \mathbf{P} \leq \mathbf{0}, \quad j = 1, \ldots, K,\ i = 1, \ldots, j-1.\end{aligned}$$

Depending on the parameters of the individual rules, it might be difficult to find a common $\mathbf{P}$ satisfying the above conditions. Reducing the number of rules results in less conservative stability conditions. Assuming that the number of active rules, s, is always less that K, $1 < s \leq K$, Tanaka et al. (1998) stated that the equilibrium of (2.16) is asymptotically stable if there exists a positive definite matrix $\mathbf{P}$ and a positive semidefinite matrix $\mathbf{Q}$ such that

$$\mathbf{G}_{ii}^T\mathbf{P}\mathbf{G}_{ii} - \mathbf{P} + (s-1)\mathbf{Q} < \mathbf{0}, \tag{2.19}$$
$$\left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T \mathbf{P} \left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T - \mathbf{P} - \mathbf{Q} \leq \mathbf{0}, \quad j = 1,\ldots,K,\ i = 1,\ldots,j-1$$

for $i, j = 1,\ldots,K$, excluding the pairs (i,j) for which $w_j\big(\mathbf{z}(k)\big) = 0, \forall\, k$ and $s > 1$.

Performance specifications. The control design problem is to determine local controllers $\mathbf{F}_i$ that satisfy the above stability conditions. Additional specifications can be imposed such as the speed of convergence, desired poles locations, or constraints on the control input and process output, etc. Here only the convergence rate is discussed, for some of the other issues see (Zhao, 1995; Tanaka et al., 1998; Bergsten, 2001).

The speed of a system response is related to the convergence rate (also called decay rate), i.e., the largest Lyapunov exponent. The system (2.1) is said to be globally exponentially stable if there exist positive constants α with $0 < \alpha < 1$ and κ such that

$$\|\mathbf{x}(k)\| \leq \kappa\alpha^k \|\mathbf{x}(0)\| \tag{2.20}$$

for any initial state $\mathbf{x}(0)$. The parameter α characterizes the rate of convergence of the system state to the equilibrium, as smaller α implies faster convergence. Expressed in terms of the quadratic function (2.13), it can be shown[1] that the above inequality amounts to (Khalil, 1992)

$$\Delta V\big(\mathbf{x}(k)\big) \leq (\alpha^2 - 1)V\big(\mathbf{x}(k)\big). \tag{2.21}$$

Taking this inequality into account, (2.19) is equivalent to

$$\mathbf{G}_{ii}^T\mathbf{P}\mathbf{G}_{ii} - \alpha^2\mathbf{P} + (s-1)\mathbf{Q} < \mathbf{0}, \tag{2.22}$$
$$\left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T \mathbf{P} \left(\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\right)^T - \alpha^2\mathbf{P} - \mathbf{Q} \leq \mathbf{0}.$$

While controllers for TS systems with linear consequents have been thoroughly treated in the literature (Tanaka and Sano, 1994, 1998; Farinwata et al., 2000; Korba, 2000), the control design for TS systems with affine consequents is more difficult. In addition, the majority of publications on feedback fuzzy control has dealt with the *regulation* problem, i.e., stabilization rather than reference tracking. Only recently

[1] Equation (2.20) implies $\mathbf{x}^T(k+1)\mathbf{P}\mathbf{x}^T(k+1) \leq \alpha^2\mathbf{x}^T(k)\mathbf{P}\mathbf{x}^T(k)$. Using the this inequality in (2.13), $\Delta V\big(\mathbf{x}(k)\big) \leq \alpha^2\mathbf{x}^T(k)\mathbf{P}\mathbf{x}^T(k) - \mathbf{x}^T(k)\mathbf{P}\mathbf{x}^T(k)$, or equivalently $\Delta V\big(\mathbf{x}(k)\big) \leq (\alpha^2 - 1)V\big(\mathbf{x}(k)\big)$. Besides, it can be shown that κ in (2.20) is equal to the ratio of the maximum and minimum eigenvalues of matrix $\mathbf{P}$.

Bergsten (2001) presented a procedure for the analysis and design of a nonlinear fuzzy controller that is able to drive the state $\mathbf{x}$ in (2.1) to any desired state $\mathbf{x}_{\text{des}}$ from an arbitrary initial condition $\mathbf{x}_0$. Moreover, the method can not only handle TS models with linear rule consequents, but also these with affine rule consequents.

In tracking problems, for a given desired state $\mathbf{x}_{\text{des}}$, there is a corresponding desired input $\mathbf{u}_{\text{des}}$. Using the fuzzy model (2.1), we can provide $\mathbf{u}_{\text{des}}$ by means of a scheduled trajectory generator:

$$\mathbf{x}_{\text{des}(k+1)} = \mathbf{A}(\mathbf{z}(k))\mathbf{x}_{\text{des}(k)} + \mathbf{B}(\mathbf{z}(k))\mathbf{u}_{\text{des}}(k) + \mathbf{a}(\mathbf{z}(k))$$

that on-line computes the desired input such that the system can be driven to the desired state for any current state.

Consider the control law based on error-state feedback

$$\mathbf{u}(k) = \mathbf{u}_{\text{des}}(k) - \mathbf{F}\big(\mathbf{z}(k)\big)\big(\mathbf{x}(k) - \mathbf{x}_{\text{des}}(k)\big)\,, \tag{2.23}$$

where $\mathbf{F}\big(\mathbf{z}(k)\big)$ is given in (2.15) and the desired input $\mathbf{u}_{\text{des}}(k)$ is computed using a trajectory generator. If we define the error between the current and the desired state as $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{x}_{\text{des}}(k)$, the closed-loop error dynamics are

$$\begin{aligned}
\mathbf{e}(k+1) &= \mathbf{A}\big(\mathbf{z}(k)\big)\mathbf{x}(k) + \mathbf{B}\big(\mathbf{z}(k)\big)\big(\mathbf{u}_{\text{des}}(k) - \mathbf{F}\big(\mathbf{z}(k)\big)\mathbf{e}(k)\big) + \mathbf{a}\big(\mathbf{z}(k)\big) - \mathbf{x}_{\text{des}}(k+1) \\
&= \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j)\mathbf{e}(k) - \mathbf{x}_{\text{des}}(k+1) \\
&\quad + \underbrace{\mathbf{A}\big(\mathbf{z}(k)\big)\mathbf{x}_{\text{des}}(k) + \mathbf{B}\big(\mathbf{z}(k)\big)\mathbf{u}_{\text{des}}(k) + \mathbf{a}\big(\mathbf{z}(k)\big)}_{\boldsymbol{\Gamma}\left(\mathbf{z}(k),\mathbf{x}_{\text{des}}(k),\mathbf{u}_{\text{des}}(k)\right)}\,.
\end{aligned} \tag{2.24}$$

Assume that $\mathbf{x}_{\text{des}}(k)$ is constant for a sufficiently long time. Then the error dynamics (2.24) are asymptotically stable if the following two conditions are fulfilled. The first condition is that

$$\boldsymbol{\Gamma}\big(\mathbf{z}(k), \mathbf{x}_{\text{des}}(k), \mathbf{u}_{\text{des}}(k)\big) \to \mathbf{x}_{\text{des}}(k+1),$$

which holds if the scheduled trajectory generator is used.

Note that if an open-loop trajectory generator

$$\mathbf{x}_{\text{des}(k+1)} = \mathbf{A}(\mathbf{z}_{\text{des}}(k))\mathbf{x}_{\text{des}}(k) + \mathbf{B}(\mathbf{z}_{\text{des}}(k))\mathbf{u}_{\text{des}}(k) + \mathbf{a}(\mathbf{z}_{\text{des}}(k))$$

is used instead, we cannot guarantee the second condition, since in general $\mathbf{z}(k) \neq \mathbf{z}(k)_{\text{des}}(k)$.

Then the error dynamics (2.24) are asymptotically stable if the system

$$\mathbf{e}(k+1) = \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j)\mathbf{e}(k)$$

is exponentially stable, i.e., $\mathbf{e}(k+1) \to 0$, which implies $\mathbf{x}(k+1) \to \mathbf{x}(k+1)_{\text{des}}(k)$. If this system is exponentially stable, then there always exists a constant $\delta > 0$ such that

$\|\mathbf{e}(k)\| \to 0$ if $\|\mathbf{e}(0)\| \leq \delta$. Sufficient stability conditions have been presented for the design of $\mathbf{F}(\mathbf{z}(k))$ that maximizes δ, see (Tanaka and Sano, 1994, 1998) and recently (Bergsten, 2001).

Another approach for reference tracking using a state feedback controller was proposed in (Korba, 2000), where the tracking is accomplished by means of an integral action.

To summarize, in this section the state feedback design for state-space TS models was reviewed. It was shown that for TS models, state feedback controllers can be obtained for which the resulting control system is stable. To this end, a couple of assumptions had to be made. First, the consequents in the model rules should be linear rather than affine, i.e., no offset terms $\mathbf{a}_i$ and $\mathbf{c}_i$ should be present. This assumption was needed so we could state the controller synthesis problem in terms of inequalities which are affine in the controller gains. Second, and a more severe assumption is that the antecedent (scheduling) variables in the TS rules do not include current control inputs, while in most cases the system operation is determined by the control signal. A method was presented that can deal with the offset terms in the rule consequents, however, it relies on different assumptions, e.g., a constant reference signal.

2.5 Fuzzy observers

In the previous section, a multiple-model design for fuzzy regulators was reviewed under the assumption that all the states of the system are available. In real control problems, however, this assumption does not always hold. Here a fuzzy observer is given that estimates the states of the TS fuzzy model.

Consider the following observer for the TS model (2.1), called the fuzzy TS observer

$$\begin{aligned} \mathcal{R}_i: \quad & \textbf{If } z_1(k) \text{ is } \mathcal{A}_{i,1} \textbf{ and } \ldots \textbf{ and } z_\rho(k) \text{ is } \mathcal{A}_{i,\rho} \textbf{ then} \\ & \begin{cases} \hat{\mathbf{x}}(k+1) = \mathbf{A}_i\hat{\mathbf{x}}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{a}_i + \mathbf{K}_i\big(\mathbf{y}(k) - \hat{\mathbf{y}}(k)\big) \\ \hat{\mathbf{y}}(k) \quad\quad = \mathbf{C}_i\hat{\mathbf{x}}(k) + \mathbf{c}_i \end{cases} \\ & i = 1, 2, \ldots, K, \end{aligned} \tag{2.25}$$

where $\mathbf{K}_i$ are the observer gains for the linear subsystems.

For the analysis of convergence properties of (2.25), two cases can be distinguished (Tanaka et al., 1998): (i) the vector of antecedent variables $\mathbf{z}(k)$ does not depend on the estimated (unmeasurable) states, and (ii) $\mathbf{z}(k)$ depends on the estimated states, which is the more difficult case (Tanaka et al., 1998; Bergsten, 2001). In the following, these cases are treated separately.

Case 1. Here the antecedent variables do not depend on the unmeasurable states. This means that the difference between the real and the estimated state depends only on the initial condition and the mismatch between the active rules and the real system. By defining the error between the real state and the state estimated through (2.25) as $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$, we obtain the following error dynamics:

$$\begin{aligned}\mathbf{e}(k+1) &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\big(\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{u}(k)+\mathbf{a}_i\big) \\ &\quad -\sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\Big(\mathbf{A}_i\hat{\mathbf{x}}(k)+\mathbf{B}_i\mathbf{u}(k)+\mathbf{a}_i+\mathbf{K}_i\big(\mathbf{y}(k)-\hat{\mathbf{y}}(k)\big)\Big) \\ &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\mathbf{A}_i\mathbf{e}(k)-\sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\mathbf{K}_i\sum_{j=1}^{K} w_j\big(\mathbf{z}(k)\big)\mathbf{C}_j\mathbf{e}(k) \\ &= \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)(\mathbf{A}_i-\mathbf{K}_i\mathbf{C}_j)\mathbf{e}(k)\,.\end{aligned} \tag{2.26}$$

The observer should satisfy $\mathbf{e}(k) \to 0$ when $k \to \infty$. This, in turn, requires stability of (2.26). For this reason, the design of an observer for which the error dynamics (2.26) are globally asymptotically stable is a problem dual to the controller design problem (2.18).

Case 2. This is the more difficult case where part or all of the antecedent variables depend on the unmeasurable states, i.e., $\mathbf{z}(k) = \hat{\mathbf{z}}(k)$. The problem here is that the fuzzy observer relies on a correct combination of linear subsystems to match the nonlinearities in the nonlinear system. Therefore, if the antecedent variables are not fully known, there is a time-varying mismatch between the observer system and the real system. This mismatch will go to zero if the state estimation error goes to zero. For simplicity, let us start with constant measurement matrices $\mathbf{C}_1 = \mathbf{C}_2 = \ldots = \mathbf{C}_K = \mathbf{C}$. In this case, the error dynamics are

$$\begin{aligned}\mathbf{e}(k+1) &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\big(\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{u}(k)+\mathbf{a}_i\big) \\ &\quad -\sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big)\Big(\mathbf{A}_i\hat{\mathbf{x}}(k)+\mathbf{B}_i\mathbf{u}(k)+\mathbf{a}_i+\mathbf{K}_i\big(\mathbf{y}(k)-\hat{\mathbf{y}}(k)\big)\Big) \\ &= \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big)(\mathbf{A}_i-\mathbf{K}_i\mathbf{C})\mathbf{e}(k) \\ &\quad +\sum_{i=1}^{K}\Big(w_i\big(\mathbf{z}(k)\big)-w_i\big(\hat{\mathbf{z}}(k)\big)\Big)\big(\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{x}(k)+\mathbf{a}_i\big) \\ &= \tilde{\mathbf{A}}\big(\hat{\mathbf{z}}(k)\big)\mathbf{e}(k)+\Delta\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big)\end{aligned} \tag{2.27}$$

where

$$\tilde{\mathbf{A}}\big(\hat{\mathbf{z}}(k)\big) = \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big)(\mathbf{A}_i-\mathbf{K}_i\mathbf{C})$$

$$\Delta\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big) = \sum_{i=1}^{K}\Big(w_i\big(\mathbf{z}(k)\big)-w_i\big(\hat{\mathbf{z}}(k)\big)\Big)\big(\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{x}(k)+\mathbf{a}_i\big)\,.$$

The term $\Delta\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big) \equiv 0$ if the weights do not depend on the estimated state $\big(w_i(\hat{\mathbf{z}}(k)) = w_i(\mathbf{z}(k))\big)$. More generally, the error dynamics (2.27) describe a polytopic system subjected to a vanishing nonlinear disturbance, which follows from the fact that $\Delta\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big) \to 0$ when $\mathbf{e}(k) \to 0$.

Based on the results presented by Bergsten and Palm (2000b) and Yoneyama and Nishikawa (2001), it can be shown that the error dynamics (2.27) are asymptotically stable if there exist symmetric positive definite matrices $\mathbf{P}$, $\mathbf{Q}$ and a scalar $\kappa \geq 0$ such that

$$
\begin{aligned}
(\mathbf{A}_i - \mathbf{K}_i\mathbf{C})^T\mathbf{P}(\mathbf{A}_i - \mathbf{K}_i\mathbf{C}) &\leq \mathbf{Q} \\
\begin{bmatrix} \mathbf{Q} - \kappa^2 & \mathbf{P} \\ \mathbf{P} & \mathbf{I} \end{bmatrix} &> 0 \\
\|\Delta\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big)\| &\leq \kappa\|\mathbf{e}(k)\|
\end{aligned}
$$

for $i = 1,\ldots,K$.

In case of a varying measurement equation, the error dynamics become

$$
\begin{aligned}
\mathbf{e}(k+1) &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\big(\mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{a}_i\big) \\
&\quad - \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big)\Big(\mathbf{A}_i\hat{\mathbf{x}}(k) + \mathbf{B}_i\mathbf{u}(k) + \mathbf{a}_i + \mathbf{K}_i\big(\mathbf{y}(k) - \hat{\mathbf{y}}(k)\big)\Big) \qquad (2.28) \\
&= \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big) \sum_{j=1}^{K} w_j\big(\hat{\mathbf{z}}(k)\big)(\mathbf{A}_i - \mathbf{K}_i\mathbf{C}_j)\mathbf{e}(k) \\
&\quad + \sum_{i=1}^{K} \Big(w_i\big(\mathbf{z}(k)\big) - w_i\big(\hat{\mathbf{z}}(k)\big)\Big)\big(\mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{x}(k) + \mathbf{a}_i\big) \\
&= \tilde{\mathbf{A}}_1\big(\hat{\mathbf{z}}(k)\big)\mathbf{e}(k) - \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}\big)\mathbf{K}_i\Delta_1\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k)\big) \qquad (2.29) \\
&\quad + \Delta_2\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big)
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{\mathbf{A}}_1\big(\hat{\mathbf{z}}(k)\big) &= \sum_{i=1}^{K} w_i\big(\hat{\mathbf{z}}(k)\big) \sum_{j=1}^{K} w_j\big(\hat{\mathbf{z}}(k)\big)(\mathbf{A}_i - \mathbf{K}_i\mathbf{C}_j) \\
\Delta_1\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k)\big) &= \sum_{j=1}^{K} \Big(w_j\big(\mathbf{z}(k)\big) - w_j\big(\hat{\mathbf{z}}(k)\big)\Big)\big(\mathbf{C}_j\mathbf{x} + \mathbf{c}_j\big) \\
\Delta_2\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big) &= \sum_{i=1}^{K} \Big(w_i\big(\mathbf{z}(k)\big) - w_i\big(\hat{\mathbf{z}}(k)\big)\Big)\big(\mathbf{A}_i\mathbf{x}(k) + \mathbf{B}_i\mathbf{x}(k) + \mathbf{a}_i\big).
\end{aligned}
$$

The error dynamics (2.28) describe a system perturbed by two vanishing disturbances $\Delta_1\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k)\big) \to 0$ when $\mathbf{e}(k) \to 0$ and $\Delta_2\big(\mathbf{z}(k),\hat{\mathbf{z}}(k),\mathbf{x}(k),\mathbf{u}(k)\big) \to 0$ when

$\mathbf{e}(k) \to 0$. The problem here is that disturbance Δ_2 is weighted by the observer gains $\mathbf{K}_i$, thus gains with large norms will make the disturbance bigger due to the state estimation error. One can avoid this complication; use a fuzzy sliding mode observer to circumvent the varying measurement equation, as shown in (Bergsten and Palm, 2000a).

To summarize, this section briefly presented the observer design for the state-space TS model. A problem arises when the antecedent variables depend on the unmeasurable states. Because, in order to match the system nonlinearity the fuzzy observer relies on a correct combination of linear subsystems, there is a mismatch between the observer system and the real system when the antecedent variables are not (fully) known. This mismatch will diminish only if the estimation error goes to zero, however, that in turn depends on the particular combination of linear subsystems.

2.6 Fuzzy output feedback design

This section uses the above results to design a control system comprising a fuzzy controller and a fuzzy observer (Fig. 2.2). Essentially, it is multiple-model output feedback control design, where the controller and the observer must satisfy $\mathbf{x}(k) \to 0$ and $\mathbf{x}(k) - \hat{\mathbf{x}}(k) \to 0$ when $k \to \infty$, respectively.

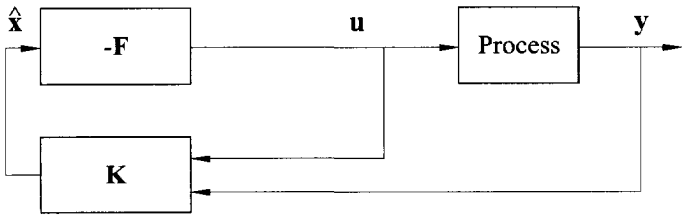


Figure 2.2. Block diagram of a fuzzy state feedback controller and a fuzzy observer.

As in the previous section, there are two cases with respect to the premise variables: in *case A* the vector of antecedent variables $\mathbf{z}(k)$ does not depend on the estimated (unmeasurable) states, and in *case B*, $\mathbf{z}(k)$ depends on the estimated states. In the following, they are treated separately.

In case A, the feedback control law is

$$\mathbf{u}(k) = -\sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)\mathbf{F}_i\hat{\mathbf{x}}(k). \tag{2.30}$$

Then the augmented system including the state estimation is represented as follows:

$$\begin{aligned}\mathbf{x}_a(k+1) &= \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)\mathbf{G}_{ij}\mathbf{x}_a(k)\\ &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big)w_i\big(\mathbf{z}(k)\big)\mathbf{G}_{ii}\mathbf{x}_a(k)\\ &\quad + 2\sum_{i=1}^{K}\sum_{j<j}^{K} w_i\big(\mathbf{z}(k)\big)w_j\big(\mathbf{z}(k)\big)\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\mathbf{x}_a(k),\end{aligned} \tag{2.31}$$

where

$$\mathbf{x}_a(k) = \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{pmatrix}, \quad \mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$$
$$\mathbf{G}_{ij} = \begin{pmatrix} \mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j & \mathbf{B}_i\mathbf{F}_j \\ \mathbf{0} & \mathbf{A}_i - \mathbf{K}_i\mathbf{C}_j \end{pmatrix}.$$

Then the equilibrium of (2.31) is asymptotically stable if there exists a symmetric positive definite matrix $\mathbf{P}$ such that

$$\mathbf{G}_{ij}^T\mathbf{P}\mathbf{G}_{ij} - \mathbf{P} < \mathbf{0} \tag{2.32}$$

for all i and j except for the pairs (i,j) s.t. $w_i(\mathbf{z}(k)w_j(\mathbf{z}(k) = 0, \forall k$.

Note that since the matrices $\mathbf{G}_{ij}$ are decoupled, we can separately design the controllers $\mathbf{F}_j$ and observers $\mathbf{K}_i$ rather than using (2.32). This is the so-called separation principle.

In *Case B* the premise variables $\mathbf{z}(k)$ are unknown since they depend on the state variable estimated by the fuzzy observer (2.25). Therefore $w_i(\hat{\mathbf{z}}(k)$ rather than $w_i(\mathbf{z}(k)$ is used in the fuzzy controller (in general $\mathbf{z}(k) \neq \hat{\mathbf{z}}(k)$, thus $w_i(\mathbf{z}(k) \neq w_i(\hat{\mathbf{z}}(k))$.

In this case, the controller output is given by the equation below (compare it to (2.30))

$$\mathbf{u}(k) = -\sum_{i=1}^{K} w_i\big(\hat{\mathbf{x}}(k)\big)\mathbf{F}_i\hat{\mathbf{x}}(k). \tag{2.33}$$

The augmented (controller & observer) closed-loop system is described through

$$\begin{aligned}\mathbf{x}_a(k+1) &= \sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{s=1}^{K} w_i\big(\mathbf{x}(k)\big)w_j\big(\hat{\mathbf{x}}(k)\big)w_s\big(\hat{\mathbf{x}}(k)\big)\mathbf{G}_{ijs}\mathbf{x}_a(k) \\ &= \sum_{i=1}^{K}\sum_{j=1}^{K} w_i\big(\mathbf{x}(k)\big)w_j\big(\hat{\mathbf{x}}(k)\big)w_s\big(\hat{\mathbf{x}}(k)\big)\mathbf{G}_{ijs}\mathbf{x}_a(k) \\ &\quad + 2\sum_{i=1}^{K}\sum_{j<s}^{K} w_i\big(\mathbf{x}(k)\big)w_j\big(\hat{\mathbf{x}}(k)\big)w_s\big(\hat{\mathbf{x}}(k)\big)\frac{\mathbf{G}_{ijs}+\mathbf{G}_{isj}}{2}\mathbf{x}_a(k),\end{aligned} \tag{2.34}$$

where

$$\mathbf{x}_a(k) = \begin{pmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{pmatrix}, \quad \mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$$
$$\mathbf{G}_{ijs} = \begin{pmatrix} \mathbf{A}_i - \mathbf{B}_i\mathbf{F}_s & \mathbf{B}_i\mathbf{F}_s \\ \mathbf{S}_{ijs}^1 \cdot \mathbf{S}_{ijs}^2 & \mathbf{0} \end{pmatrix}$$
$$\mathbf{S}_{ijs}^1 = \big(\mathbf{A}_i - \mathbf{A}_j\big) - \big(\mathbf{B}_i - \mathbf{B}_j\big)\mathbf{F}_s + \mathbf{K}_j\big(\mathbf{C}_s - \mathbf{C}_i\big)$$
$$\mathbf{S}_{ijs}^2 = \mathbf{A}_i - \mathbf{K}_j\mathbf{C}_s + \big(\mathbf{B}_i - \mathbf{B}_j\big)\mathbf{F}_s\,.$$

Regarding the closed-loop system (2.34), a sufficient condition for the equilibrium to be globally asymptotically stable is the existence of a common positive definite matrix $\mathbf{P}$ such that

$$\mathbf{G}_{iii}^T \mathbf{P} \mathbf{G}_{iii} - \mathbf{P} < \mathbf{0}, \quad i = 1, \ldots, K$$

$$\left(\frac{\mathbf{G}_{ijs} + \mathbf{G}_{isj}}{2}\right)^T \mathbf{P} \left(\frac{\mathbf{G}_{ijs} + \mathbf{G}_{isj}}{2}\right)^T - \mathbf{P} \leq \mathbf{0} \quad i, s = 1, \ldots, K,\ j = 1, \ldots, s-1 .$$

Here the matrices $\mathbf{G}_{ijs}$ are coupled, so it is not possible to separately design the controllers $\mathbf{F}_j$ and observers.

2.7 Robustness issues

Robust stability of systems subject to (vanishing) disturbances has been considered by incorporation of the disturbances into the system description, see (Chen and Han, 1994; Kim, 1995) for a general, i.e., non-fuzzy treatment and recently (Bergsten, 2001) for a fuzzy-oriented approach to the problem.

The stability results presented so far are based on a single quadratic Lyapunov function, also called quadratic stability. While for LTI systems quadratic stability is necessary and sufficient, for polytopic systems (and thus also TS models) quadratic stability is only sufficient and may become quite conservative. In such cases, piecewise quadratic Lyapunov functions (Bogatyrev and Pyatnitskii, 1987) or parameter-dependent Lyapunov functions (Johansson et al., 1999; Bergsten, 2001) can be used to reduce the conservatism.

Yet another approach was proposed by Zhao et al. (1997). The authors consider the uncertainties in the fuzzy model in terms of premise uncertainties, i.e., derived from the membership functions specifying the local regions, and consequent uncertainties which are in the system matrices $\mathbf{A}_i$ and $\mathbf{B}_i$.

The premise uncertainties stem from the uncertainties in the shape of membership functions, and result in uncertainties on the corresponding degree of fulfillment (DOF). If the nominal DOF for the ith rule is β_i (recall (2.1)), then the corresponding uncertainty is denoted by $\Delta\beta_i(k)$ for which

$$\begin{aligned} 0 \leq \beta_i + \Delta\beta_i(k) \leq 1 \\ \sum_{i=1}^{K} \beta_i + \Delta\beta_i(k) = 1, \quad \forall k . \end{aligned} \tag{2.35}$$

In practice, it is more reasonable to study a particular case of (2.35), where $\Delta\beta_i(k)$ is restricted to be within

$$-\gamma^{\text{pre}} \beta_i \leq \Delta\beta_i(k) \leq \gamma^{\text{pre}} \big(1 - \beta_i\big). \tag{2.36}$$

The parameter $0 \leq \gamma^{\text{pre}} \leq 1$ specifies the range of the uncertainty. For example, assuming a trapezoidal shape of the degree of fulfillment β_i and a given γ^{pre}, an admissible region for $\Delta\beta_i(k)$ is shown in Fig. 2.3.

Since any unknown uncertainty $\Delta\beta_i(k)$ can be described as a convex combination of the left- and right-hand sides of (2.36)

$$\begin{aligned} \Delta\beta_i(k) &= \gamma^{\text{pre}} \Big(\alpha_i(k) \big(1 - \beta_i(k)\big) + \big(1 - \alpha_i(k)\big) \big(-\beta_i(k)\big) \Big) \\ &= \gamma^{\text{pre}} \big(\alpha_i(k) - \beta_i(k) \big) \end{aligned}$$

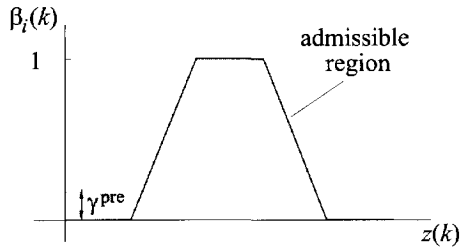


Figure 2.3. Admissible region of the uncertainty on the degree of fulfillment.

with $0 \leq \alpha_i(k) \leq 1$, the antecedent uncertainties can be reduced to matrix terms in the model description (2.1)

$$\begin{aligned}\mathbf{x}(k+1) &= \sum_{i=1}^{K} \Big(w_i\big(\mathbf{z}(k)\big) + \Delta w_i\big(\mathbf{z}(k)\big) \Big) \big(\mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}(k) \big) \\ &= \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big) \Big(\big(\mathbf{A}_i + \Delta \mathbf{A}_i^{\text{pre}}(k) \big) \mathbf{x}(k) + \big(\mathbf{B}_i + \Delta \mathbf{B}_i^{\text{pre}}(k) \big) \mathbf{u}(k) \Big). \end{aligned} \tag{2.37}$$

Assuming a common γ^{pre} for all rules, the unknown matrices $\Delta \mathbf{A}_i^{\text{pre}}(k)$ and $\Delta \mathbf{B}_i^{\text{pre}}(k)$ are bounded in a polytope

$$\big[\Delta \mathbf{A}_i^{\text{pre}}(k), \Delta \mathbf{B}_i^{\text{pre}}(k) \big] \in \gamma^{\text{pre}} \mathcal{C} \big\{ \big[\Delta \mathbf{A}_{i1}^{\text{pre}}, \Delta \mathbf{B}_{i1}^{\text{pre}} \big], \ldots, \big[\Delta \mathbf{A}_{iK}^{\text{pre}}, \Delta \mathbf{B}_{iK}^{\text{pre}} \big] \big\}$$

with vertices

$$\Delta \mathbf{A}_{i1}^{\text{pre}} = -\mathbf{A}_i + \mathbf{A}_l, \quad \Delta \mathbf{B}_{i1}^{\text{pre}} = -\mathbf{B}_i + \mathbf{B}_l, \quad l = 1, \ldots, K.$$

The consequent uncertainties stem from the modeling errors within the local subsystems. A model with consequent uncertainties can be written as

$$\mathbf{x}(k+1) = \sum_{i=1}^{K} w_i\big(\mathbf{z}(k)\big) \Big(\big(\mathbf{A}_i + \Delta \mathbf{A}_i^{\text{con}}(k) \big) \mathbf{x}(k) + \big(\mathbf{B}_i + \Delta \mathbf{B}_i^{\text{con}}(k) \big) \mathbf{u}(k) \Big), \tag{2.38}$$

where $\Delta \mathbf{A}_i^{\text{con}}(k)$ and $\Delta \mathbf{B}_i^{\text{con}}(k)$ are the uncertainties on the consequent of the ith rule, bounded within a polytope with vertices $\big[\gamma^{\text{con}} \Delta \mathbf{A}_{il}^{\text{con}}, \gamma^{\text{con}} \Delta \mathbf{B}_{il}^{\text{con}} \big]$, $l = 1, \ldots, K$ and $\gamma^{\text{con}} \geq 0$. Hence, the consequent uncertainties can be expressed as

$$\big[\Delta \mathbf{A}_i^{\text{con}}(k), \Delta \mathbf{B}_i^{\text{con}}(k) \big] \in \gamma^{\text{con}} \mathcal{C} \big\{ \big[\Delta \mathbf{A}_{i1}^{\text{con}}, \Delta \mathbf{B}_{i1}^{\text{con}} \big], \ldots, \big[\Delta \mathbf{A}_{iK}^{\text{con}}, \Delta \mathbf{B}_{iK}^{\text{con}} \big] \big\},$$

or correspondingly

$$\big[\Delta \mathbf{A}_i^{\text{con}}(k), \Delta \mathbf{B}_i^{\text{con}}(k) \big] \in \gamma^{\text{con}} \sum_{j=1}^{K} w_i\big(\mathbf{z}(k)\big) \big[\Delta \mathbf{A}_{ij}^{\text{con}}, \Delta \mathbf{B}_{ij}^{\text{con}} \big].$$

Comparing (2.37) and (2.38) with the nominal model (2.1), the robust output feedback design amounts to providing a controller and an observer based on the nominal model such that the resulting fuzzy control system has a maximum stability margin for uncertainties in either the antecedent or consequent part.

2.8 Summary and concluding remarks

This chapter introduced the Takagi–Sugeno fuzzy model that is used in methods developed in the following chapters. When developing a process model to be used for control design, one must make sure the process model represents the process with sufficient accuracy in the regions where it is expected to operate. TS fuzzy models offer a convenient way for process modeling as they are built on local linear submodels, by using a fuzzy inference mechanism to interpolate between the submodels. Depending on the knowledge and information available, we can derive state-space or input-output TS fuzzy models.

When a state-space TS model is available, the fuzzy controller can be designed by means of state feedback. The control design then reflects the structure of the TS fuzzy model: linear controllers are developed for the separate rules in the model, and then the overall controller is a fuzzy blending of the individual linear controllers. The same idea is used to obtain fuzzy observers. When the LMI framework is applied, the controller synthesis reduces to an LMI problem which can be solved numerically. In this way performance and robustness specifications can be imposed on the resulting control system. Still, there are issues of a different character that limit the application of the described methods, both where it concerns conceptual and implementation aspects. First, the method cannot deal with TS models including current control inputs in the antecedent part. Second, the method cannot be directly applied in the presence of an offset term in the rule consequents. Third, fuzzy state observers are difficult to design if the antecedent part includes estimated states: the fuzzy observer relies on a correct combination of linear subsystems to account for the nonlinearities in the nonlinear system, thus if the antecedent variables are not fully known, there is a time-varying mismatch between the observer system and the real system, which only disappears if the state estimation error goes to zero. Additionally, the number of local controllers and observers that are needed for accurate control may increase rapidly as the complexity of the system increases. Moreover, when the robustness issue is addressed, the underlying optimization problem becomes extensive even for a limited number of rules. In the consequent chapters, different paradigms were used for the development of methods for fuzzy control design, as shown in Chapter 3 (input-output decoupling), Chapter 4 (fuzzy model predictive control) and Chapter 5 (robust stability issues in fuzzy model predictive control).

3 ANALYSIS OF INTERACTIONS IN TS MODELS AND INPUT-OUTPUT DECOUPLING

Input-output interactions in the inherently nonlinear TS fuzzy models cannot be analyzed with standard methods such as the Relative Gain Array that have been developed for linear systems. In this chapter, two new methods are proposed that exploit the specific model structure of the TS models to indicate interactions. The first method is based on a set of RGAs derived at different operating regions (Section 3.2.1). The second method is based on the output sensitivity function, computed as a partial derivative of the output with respect to the considered input (Section 3.2.2).

Next, in Section 3.3, a decoupling strategy for TS fuzzy models is proposed. The idea is to invert the fuzzy model and to use the inverted model to compensate for the coupling. When the model is an affine one, it is possible to obtain an analytic decoupling law by applying tools from the differential geometry theory. In most cases, however, the TS model is non-affine. In such a situation, we can either compute the input-output gains at each sampling instant and invert them, or numerically invert the model by solving a nonlinear optimization problem.

Both methods for the analysis of interactions are applied to a laboratory-scale cascaded-tanks setup (Section 3.4). Input-output decouplers for this system are designed based on affine and non-affine fuzzy models.

3.1 Introduction

A multivariable process contains several controlled variables that can be influenced by various manipulated variables. When a single controlled variable (output) is controlled by a single manipulated variable (control input), this part of the process is considered as a single-loop process, regardless of the number of disturbing and uncontrolled variables present. When the action of a loop influences another loop, then the loops are interacting, or coupled. In practice, in a multivariable process nearly each output is influenced by each input, and if we want to achieve non-interacting (i.e., decoupled) control, where each input influences only one output and vice versa, then we should use a multivariable controller based either on detached SISO controllers in combination with a decoupler or on a multivariable process model that takes into account the interactions when forming the control signal. In this chapter, we achieve decoupled control based on detached SISO controllers and a decoupler.

For each input there is a path through which this input dominantly influences a certain output. These paths together present the *input-output pairing*. The determination of such a pairing between the process inputs and outputs is an important element in the control design because of the influence it has on the performance achieved by the control system. It is essential to choose the controller type and structure properly because in many cases interactions result in undesirable effects, e.g., the loops upset each other and can even destabilize the entire system. One or more of the controllers can usually be detuned to retain stability, but with a certain loss of performance. In some processes, interactions are so severe that satisfactory control system performance cannot be obtained even with the best loop pairing and optimal tuning, or even worse, stability is may not be attainable without decoupling. In such situations, a decoupling design is needed to eliminate the effects caused by the interacting paths (Hui, 1983; Shinskey, 1996). The decoupling problem becomes even more complicated when the process is nonlinear. Then, within a certain operation region, input-output coupling could appear to be absent, while in another coupling would be significant. For the first region SISO controllers might suffice, while the second would require full or partial decoupling (Isidori, 1995; Kotta, 1995).

3.2 Analysis of interactions in TS fuzzy models

Nonlinear models exhibit operating-regime-dependent input-output coupling. Therefore, tools are needed that take into account the model's nonlinear characteristics when analyzing the input-output interactions. In this section, two methods are presented that exploit the specific structure of the TS model in the analysis of interactions. The first method is based on a number of Relative Gain Arrays computed for the individual fuzzy rules. The second method is based on an output sensitivity analysis (Mollov, Babuška and Verbruggen, 2001b).

3.2.1 RGA approach

The *Relative Gain Array* (RGA) concept has been widely used in linear multivariable processes as a measure of the interaction between the control loops (Bristol, 1966;

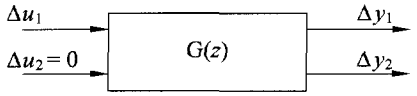
Grossdidier et al., 1985; Hovd and Skogestad, 1992). The RGA indicates the influence of a particular input on the output under consideration, relative to that of other inputs acting on the process. Consider the 2×2 system $\mathbf{G}(z)$ shown in Fig. 3.1. In Fig. 3.1a, a change in input u_1 is introduced, while input u_2 is kept constant. The result is changes in both outputs, Δy_1 and Δy_2. Because u_2 is kept constant, we say that the loop between y_2 and u_2 is open, i.e., no controller is placed between y_2 and u_2. In Fig. 3.1b, the same change in u_1 is introduced, however, this time a controller C is used to keep y_2 constant, i.e., the loop between y_2 and u_2 is closed. The element λ_{11} in the Relative Gain Array is defined as the ratio of the effect of input u_1 on output y_1 with the other loop (y_2 to u_2) open to the effect observed with the other loop closed. For a $p \times m$ system, the elements $\lambda_{l,j}$ in the Relative Gain Array are defined as the ratio of the effect a particular input u_j has on a particular output y_l with all loops open to the effect that would be observed if all other loops were closed (Bristol, 1966)

$$\lambda_{l,j} = \frac{(\Delta y_l / \Delta u_j)_{\Delta u_k = 0,\ k \neq j}}{(\Delta y_l / \Delta u_j)_{\Delta y_k = 0,\ k \neq l}}. \tag{3.1}$$

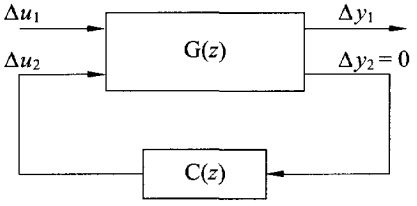
For a non-singular square ($p = m$) transfer matrix $\mathbf{G}(\mathbf{z}) \in \mathbb{C}^{p \times m}$, the relative gain array is a matrix $\Lambda(z) \in \mathbb{C}^{p \times m}$

$$\Lambda(z) \equiv \mathbf{G}(z) .* \left(\mathbf{G}(z)^{-1}\right)^T, \tag{3.2}$$

where $.*$ denotes element-by-element multiplication (Hadamard or Schur product) (Grossdidier et al., 1985). In general $\mathbf{G}(z)$, hence also $\Lambda(z)$ are frequency dependent, but for the sake of simplicity we assume $z = 1$. Thus, here we restrict ourselves to the analysis of static interactions.



(a) Open-loop operation.



(b) Closed-loop operation.

Figure 3.1. Relative Gain Array definition.

However, the input-output interactions of fuzzy models of the TS type cannot be analyzed through a single Relative Gain Array. The behaviour of the nonlinear TS

model, so also the input-output coupling, depends on the operating point. A possible way to explore the interactions is to linearize the fuzzy model around the point of interest, and to compute RGA for the linearized model. This is, however, quite a laborious approach. It is more appealing to make use of the model structure – a combination of local linear models. If the structure of the rule antecedents is identical for all outputs, i.e., if (i) all outputs have the same antecedent variables, and (ii) these variables have the same linguistic terms ('Low', 'Medium', etc.), the rules with the same antecedent for the individual outputs can be combined into one rule. In this way, a set of multivariable rules is obtained. Then for each rule, one RGA can be computed based on the consequent parameters

$$\Lambda_i = \mathbf{G}_i . * \left(\mathbf{G}_i^{-1}\right)^T = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1m} \\ \lambda_{21} & \lambda_{22} & \dots & \lambda_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{p1} & \lambda_{p2} & \dots & \lambda_{pm} \end{bmatrix} \quad i = 1, \dots, K \tag{3.3}$$

where $\mathbf{G}_i$ is the matrix of transfer functions for rule i

$$\mathbf{G}_i = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1} & g_{p2} & \cdots & g_{pm} \end{bmatrix}. \tag{3.4}$$

Remark 3.1 *The nature of the interaction can be shown through the RGA elements (Shinskey, 1996). Zero λ_{lj} implies that output l is not influenced by input j. Negative λs mean that the interaction has a destabilizing effect. Positive λs in $[0,1]$ indicate oscillations and an extra phase lag. When λ is above one, the static response is deteriorated (Fig. 3.2).*

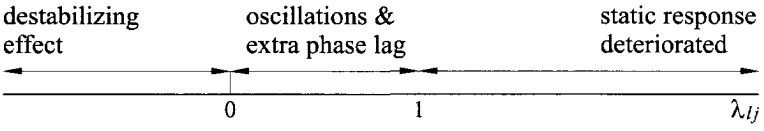


Figure 3.2. Interpretation of the RGA elements.

The elements g_{lj} are the d.c. gains from input j to output l

$$g_{lj} = \frac{\sum_{h=n_{d,j}}^{n_{d,j}+n_{u,j}} \eta_{li,h}}{1 - \sum_{h=1}^{\rho} \zeta_{li,h}} \qquad l = 1, \dots, p, \quad j = 1, \dots, m \tag{3.5}$$

Here, n_d denotes the position at which the first element of the respective input is placed in the input regression vector $\mathbf{u}(k)$, and $\boldsymbol{\zeta}_{li,h}$ and $\boldsymbol{\eta}_{li,h}$ are the coefficients in front of the state and input vectors $\mathbf{x}(k)$ and $\mathbf{u}(k)$ in (2.6), respectively. Note that in order to keep (3.5) simple, we assume that the delayed input values are in the input vector $\mathbf{u}(k)$

rather than in the model state $\mathbf{x}(k)$ as in (2.2). The offset term is omitted since it can be considered as a constant external input, and we are interested in the output variation as a result of a input change; not in its absolute value (recall (3.1)). In the sequel, the RGA that is computed by using formulas (3.3) through (3.5) will be referred to as the "individual RGA." Note that since the interactions indicated in this way are static, to have a meaningful interpretation one needs local linear models that are stable.

The following Example 3.1 is used to illustrate the computation of the RGAs for the separate rules and the influence of the model parameters on the indicated interactions.

Example 3.1 *Consider an artificial fuzzy model with two inputs and two outputs. Four regions are defined by splitting the outputs' domains in two parts ("*Small*" and "*Large*"). For each of these regions, a stable linear model has been obtained. The fuzzy rules are given below.*

Output 1

1. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Small **then**
$y_1(k+1) = \zeta_{111} y_1(k) + \eta_{111} u_1(k) + \eta_{112} u_2(k) + \theta_{11}$
2. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Large **then**
$y_1(k+1) = \zeta_{121} y_1(k) + \eta_{121} u_1(k) + \eta_{122} u_2(k) + \theta_{12}$
3. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Small **then**
$y_1(k+1) = \zeta_{131} y_1(k) + \eta_{131} u_1(k) + \eta_{132} u_2(k) + \theta_{13}$
4. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Large **then**
$y_1(k+1) = \zeta_{141} y_1(k) + \eta_{141} u_1(k) + \eta_{142} u_2(k) + \theta_{14}$

(3.6a)

Output 2

1. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Small **then**
$y_2(k+1) = \zeta_{211} y_2(k) + \eta_{211} u_1(k) + \eta_{212} u_2(k) + \theta_{21}$
2. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Large **then**
$y_2(k+1) = \zeta_{221} y_2(k) + \eta_{221} u_1(k) + \eta_{222} u_2(k) + \theta_{22}$
3. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Small **then**
$y_2(k+1) = \zeta_{231} y_2(k) + \eta_{231} u_1(k) + \eta_{232} u_2(k) + \theta_{23}$
4. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Large **then**
$y_2(k+1) = \zeta_{241} y_2(k) + \eta_{241} u_1(k) + \eta_{242} u_2(k) + \theta_{24}$

(3.6b)

The membership functions for the linguistic terms Small *and* Large *defined on the domains of the outputs y_1 and y_2, respectively, are depicted in Fig. 3.3.*

The steady-state gain matrices for the separate rules are

$$\mathbf{G}_1 = \begin{bmatrix} \frac{\eta_{111}}{1-\zeta_{111}} & \frac{\eta_{112}}{1-\zeta_{111}} \\ \frac{\eta_{211}}{1-\zeta_{211}} & \frac{\eta_{212}}{1-\zeta_{211}} \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} \frac{\eta_{121}}{1-\zeta_{121}} & \frac{\eta_{122}}{1-\zeta_{121}} \\ \frac{\eta_{221}}{1-\zeta_{221}} & \frac{\eta_{222}}{1-\zeta_{221}} \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} \frac{\eta_{131}}{1-\zeta_{131}} & \frac{\eta_{132}}{1-\zeta_{131}} \\ \frac{\eta_{231}}{1-\zeta_{231}} & \frac{\eta_{232}}{1-\zeta_{231}} \end{bmatrix} \quad \mathbf{G}_4 = \begin{bmatrix} \frac{\eta_{141}}{1-\zeta_{141}} & \frac{\eta_{142}}{1-\zeta_{141}} \\ \frac{\eta_{241}}{1-\zeta_{241}} & \frac{\eta_{242}}{1-\zeta_{241}} \end{bmatrix}$$

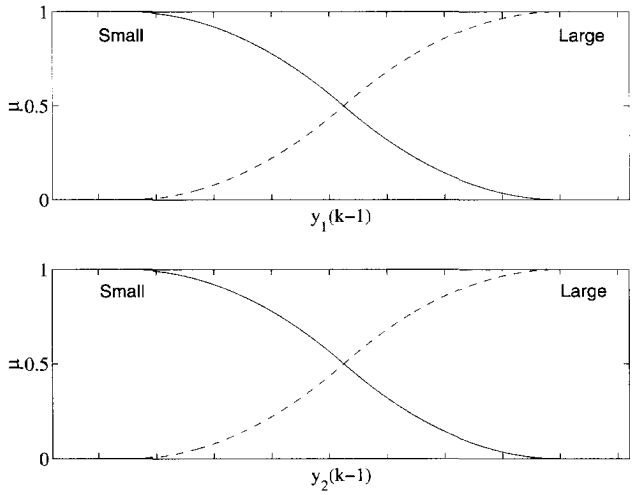


Figure 3.3. Membership functions.

To obtain $\Lambda_1 = \mathbf{G}_1 .* (\mathbf{G}_1^{-1})^T$, *we first compute the transposed inverse of* $\mathbf{G}_1$

$$\left(\mathbf{G}_1^{-1}\right)^T = \frac{1}{\eta_{111}\eta_{212} - \eta_{112}\eta_{211}} \begin{bmatrix} (1-\zeta_{111})\eta_{212} & -(1-\zeta_{111})\eta_{211} \\ -(1-\zeta_{211})\eta_{112} & (1-\zeta_{211})\eta_{111} \end{bmatrix}$$

and then

$$\Lambda_1 = \frac{1}{\eta_{111}\eta_{212} - \eta_{112}\eta_{211}} \begin{bmatrix} \eta_{111}\eta_{212} & -\eta_{112}\eta_{211} \\ -\eta_{211}\eta_{112} & \eta_{212}\eta_{111} \end{bmatrix}. \tag{3.7}$$

From the above example one can see that any row or column sums to one in this simplest case of a 2×2 model (Bristol, 1966; Grossdidier et al., 1985). This property, however, reflects on the indicated interactions in a special way. Let $\eta_{112} = 0$, i.e., u_2 does not influence y_1 in the first rule in (3.6a). In this case, $\lambda_{12} = 0$ as one should expect. Additionally, however, we have $\lambda_{21} = 0$ as well, which indicates that y_2 is not affected by u_1, regardless of the η_{211} value. A similar "symmetry" is to be seen whenever there is an η zero. This situation can be considered as the extreme case of the symmetry in the indicated interactions: for a 2×2 model, Λ is always symmetrical.

The question that naturally arises is whether it is possible to analyze the interactions in between the rules using the RGAs computed for the individual rules, e.g., whether the coupling that the TS model can display between the rules is restricted by the pre-computed RGAs for the separate rules. Unfortunately the answer is, in general, negative. This originates from formula (3.2) and more precisely because of the nonlinearity of the RGA computed in this way with respect to the degree of fulfillment, which is due to the matrix inversion. To illustrate the problem, consider the way the upper-left element of the RGA (λ_{11}) varies during a change in the active rule, e.g., from rule one to rule two. The transfer function matrix is a linear function of the

degree of fulfillment β:

$$\begin{aligned}\mathbf{G}_\beta &= (1-\beta)\cdot\begin{bmatrix}\frac{\eta_{111}}{1-\zeta_{111}} & \frac{\eta_{112}}{1-\zeta_{111}}\\ \frac{\eta_{211}}{1-\zeta_{211}} & \frac{\eta_{212}}{1-\zeta_{211}}\end{bmatrix}+\beta\begin{bmatrix}\frac{\eta_{121}}{1-\zeta_{121}} & \frac{\eta_{122}}{1-\zeta_{121}}\\ \frac{\eta_{221}}{1-\zeta_{221}} & \frac{\eta_{222}}{1-\zeta_{221}}\end{bmatrix}\\ &= \begin{bmatrix}\frac{(1-\beta)\eta_{111}}{1-\zeta_{111}}+\frac{\beta\eta_{121}}{1-\zeta_{121}} & \frac{(1-\beta)\eta_{112}}{1-\zeta_{111}}+\frac{\beta\eta_{122}}{1-\zeta_{121}}\\ \frac{(1-\beta)\eta_{211}}{1-\zeta_{211}}+\frac{\beta\eta_{221}}{1-\zeta_{221}} & \frac{(1-\beta)\eta_{212}}{1-\zeta_{211}}+\frac{\beta\eta_{222}}{1-\zeta_{221}}\end{bmatrix}.\end{aligned} \tag{3.8}$$

Applying formula (3.4) to $\mathbf{G}_\beta$, we obtain Λ_β with elements λ_{ij} that are rational functions of β^2 and the model parameters. For example,

$$\lambda_{11,\beta}(\boldsymbol{\zeta},\boldsymbol{\eta}) = \frac{f_1(\beta^2,\boldsymbol{\zeta},\boldsymbol{\eta})}{f_2(\beta^2,\boldsymbol{\zeta},\boldsymbol{\eta})}, \tag{3.9}$$

i.e., the shape of $\lambda_{11,\beta}$ depends on the model parameters, hence we cannot draw conclusions regarding the monotonicity of $\lambda_{11,\beta}$ nor about the interval in which it is restricted. A more specific example where $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$ have numerical values is presented in Chapter 7.

Below two other methods are proposed that allow for the analysis of interactions when two or more rules are active. The first method is an extension of the individual RGA method in which the separate RGAs are multiplied by the degree of fulfillment (DOF) for the point under consideration.

The second method is based on a modification of (3.5) that computes the RGA for a point between the rules, i.e., when more than one rule is active, or when the rule antecedent structure differs per output. The DOF for that point is combined with the rule consequents: $\beta_{li}\big(\boldsymbol{\chi}_l(k)\big)$ are calculated according to (2.7) and $\boldsymbol{\zeta}_l\big(\boldsymbol{\chi}_l(k)\big)$ and $\boldsymbol{\eta}_l\big(\boldsymbol{\chi}_l(k)\big)$ – according to (2.10). The elements g_{lj} in the matrix $\mathbf{G}$ (3.4) are

$$g_{lj} = \frac{\sum_{h=1}^{1+n_{ul,j}} \eta_{l,h}\big(\boldsymbol{\chi}_l(k)\big)}{1-\sum_{h=1}^{\rho}\zeta_{l,h}\big(\boldsymbol{\chi}_l(k)\big)} \qquad l=1,\ldots,p, \quad j=1,\ldots,m \tag{3.10}$$

and at that point $\big(\boldsymbol{\chi}_l(k)\big)$ the RGA is calculated via (3.2). In the sequel, the RGA computed by using formulas (3.2) and (3.10) will be referred to as the "combined RGA." Note that since the interactions indicated in this way are static, to have a meaningful interpretation one needs to end up with a resulting linear model that is stable.

The following Example 3.2 illustrates how the RGAs are to be computed when the antecedent structure of the TS model is different for the different MISO submodels.

Example 3.2 *Consider again the fuzzy model used in Example 3.1, but now divide the domain of y_1 (see (3.6a)) into three parts ("*Small*", "*Medium*" and "*Large*"), as shown in Fig. 3.4. As before, the $y_2(k)$ domain is split into two regions. The MISO model for y_2 remains the same as that in Example 3.1.*

The corresponding fuzzy rules are

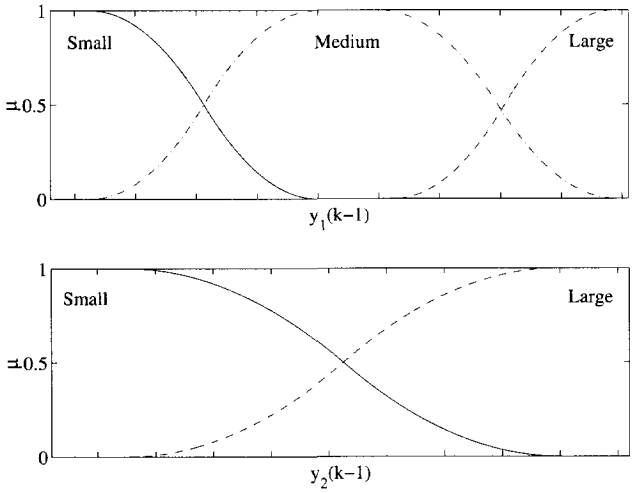


Figure 3.4. Membership functions for output y_1.

Output 1

1. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Small **then**
$y_1(k+1) = \zeta_{111} y_1(k) + \eta_{111} u_1(k) + \eta_{112} u_2(k) + \theta_{11}$
2. **If** $y_1(k)$ *is* Small **and** $y_2(k)$ *is* Large **then**
$y_1(k+1) = \zeta_{121} y_1(k) + \eta_{121} u_1(k) + \eta_{122} u_2(k) + \theta_{12}$
3. **If** $y_1(k)$ *is* Medium **and** $y_2(k)$ *is* Small **then**
$y_1(k+1) = \zeta_{131} y_1(k) + \eta_{131} u_1(k) + \eta_{132} u_2(k) + \theta_{13}$
4. **If** $y_1(k)$ *is* Medium **and** $y_2(k)$ *is* Large **then**
$y_1(k+1) = \zeta_{141} y_1(k) + \eta_{141} u_1(k) + \eta_{142} u_2(k) + \theta_{14}$
5. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Small **then**
$y_1(k+1) = \zeta_{151} y_1(k) + \eta_{151} u_1(k) + \eta_{152} u_2(k) + \theta_{15}$
6. **If** $y_1(k)$ *is* Large **and** $y_2(k)$ *is* Large **then**
$y_1(k+1) = \zeta_{161} y_1(k) + \eta_{161} u_1(k) + \eta_{162} u_2(k) + \theta_{16}$

(3.11)

Now it is not possible to match rules for y_1 and y_2, because the degrees of fulfillment for a given combination $\big(y_1(k), y_2(k)\big)$ are not the same in the MISO models (3.6a) and (3.11) due to the extra linguistic term for $y_1(k)$.

For a given operating point $\boldsymbol{\chi}(k) = \big(y_{10}(k), y_{20}(k)\big)$ we have the DOF for y_1 and y_2 $\boldsymbol{\beta}_1\big(\boldsymbol{\chi}(k)\big) = [\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}]$ and $\boldsymbol{\beta}_2\big(\boldsymbol{\chi}(k)\big) = [\beta_{21}, \beta_{22}, \beta_{23}, \beta_{24}]$, respectively. Combining $\boldsymbol{\beta}_1\big(\boldsymbol{\chi}(k)\big)$ and $\boldsymbol{\beta}_2\big(\boldsymbol{\chi}(k)\big)$ with the rule consequents, we obtain the local linear models (recall (2.9))

$$
\begin{aligned}
y_1(k+1) &= \zeta_1\left(\boldsymbol{\chi}(k)\right) y_1(k) + \eta_{11}\left(\boldsymbol{\chi}(k)\right) u_1(k) + \eta_{12}\left(\boldsymbol{\chi}(k)\right) u_2(k) + \theta_1\left(\boldsymbol{\chi}(k)\right) \\
y_2(k+1) &= \zeta_2\left(\boldsymbol{\chi}(k)\right) y_2(k) + \eta_{21}\left(\boldsymbol{\chi}(k)\right) u_1(k) + \eta_{22}\left(\boldsymbol{\chi}(k)\right) u_2(k) + \theta_2\left(\boldsymbol{\chi}(k)\right)
\end{aligned}
$$

The corresponding transfer matrix $\mathbf{G}(\boldsymbol{\chi}(k))$ *is*

$$\mathbf{G}(\boldsymbol{\chi}(k)) = \begin{bmatrix} \dfrac{\eta_{11}(\boldsymbol{\chi}(k))}{1-\zeta_1(\boldsymbol{\chi}(k))} & \dfrac{\eta_{12}(\boldsymbol{\chi}(k))}{1-\zeta_1(\boldsymbol{\chi}(k))} \\ \dfrac{\eta_{21}(\boldsymbol{\chi}(k))}{1-\zeta_2(\boldsymbol{\chi}(k))} & \dfrac{\eta_{22}(\boldsymbol{\chi}(k))}{1-\zeta_2(\boldsymbol{\chi}(k))} \end{bmatrix}$$

and $\Lambda(\boldsymbol{\chi}(k))$ *is calculated analogously to (3.7).*

There is another interesting question with regard to the input and output dimensions: is the formula (3.2) still valid when $m \neq p$, i.e., when $\mathbf{G}$ is not square. This problem is essentially related to the computation of the matrix inverse in (3.2). When $\mathbf{G}$ is not square, $\mathbf{G}^{\dagger}$ is the *pseudoinverse* rather than the inverse matrix (A pseudoinverse of $\mathbf{G}$ is a matrix $\mathbf{X}$ of the same dimensions as $\mathbf{G}^T$ so that $\mathbf{GXG} = \mathbf{G}$, $\mathbf{XGX} = \mathbf{X}$, and $\mathbf{GX}$ and $\mathbf{XG}$ are Hermitian[1] (Golub and van Loan, 1996)). Then (3.2) can be generalized by

$$\Lambda = \mathbf{G}. * \mathbf{G}^{\dagger}. \tag{3.12}$$

Note, however, that because $\mathbf{G}$ is not square, it is no longer true that any row or column sums to one. Instead, if $m > p$ we have that $\mathbf{GG}^{\dagger} = \mathbf{I}$, hence the elements in each row of the RGA sum to one, $\sum_{j=1}^{m} \lambda_{ij} = 1$. If $m < p$ then $\mathbf{G}^{\dagger}\mathbf{G} = \mathbf{I}$ and thus the elements in each column sum to one, $\sum_{i=1}^{p} \lambda_{ij} = 1$.

In the sequel we do not distinguish between $\mathbf{G}^{\dagger}$ and $\mathbf{G}^{-1}$ to denote the (pseudo-) inverse of $\mathbf{G}$ and use the latter as a generally accepted notation.

3.2.2 Output sensitivity analysis

The RGA methods discussed in the previous section have two common weak points: (i) only the static interactions can be analysed, and (ii) for 2×2 processes, the indicated interactions are always symmetrical, regardless of the actual interactions. These weaknesses can be avoided by using the output sensitivity analysis. A similar method was proposed for the analysis of interaction in linguistic fuzzy models (Gegov et al., 1999), however, no such techniques were available for the TS model.

The output sensitivity analysis quantifies (relatively) the sensitivity of the output under consideration to a certain input. The output sensitivity is given through the function $s_{u_j(k)}^{y_l(k+1)}$, which is the partial derivative of the output with respect to the specified input, while the remaining inputs are kept constant (Mollov, Babuška and Verbruggen, 2001b). It is a measure of sensitivity of the lth process output, $l = 1, \ldots, p$ at time instant $k+1$, to jth input variation, $j = 1, \ldots, m$ at time instant k

$$s_{u_j(k)}^{y_l(k+1)} = \frac{\partial\big(\ln y_l(k+1)\big)}{\partial\big(\ln u_j(k)\big)} = \frac{\partial\big(\ln y_l(k+1)\big)}{\partial u_j(k)} \cdot \frac{\partial u_j(k)}{\partial\big(\ln u_j(k)\big)}. \tag{3.13}$$

[1] A matrix $\mathbf{G} \in \mathbb{C}^{p \times p}$ is called a Hermitian matrix if $\mathbf{G}^H = \mathbf{G}$, where $\mathbf{G}^H$ denotes the transposed skew-symmetric matrix of $\mathbf{G}$.

Notice that

$$\frac{\partial\big(\ln y_l(k+1)\big)}{\partial u_j(k)} = \frac{1}{y_l}\frac{\partial y_l}{\partial u_j} \quad \text{and}$$
$$\frac{\partial\big(\ln u_j(k)\big)}{\partial u_j} = \frac{1}{u_j}.$$

Let $u_{j,0}$ denote the value at which the sensitivity is computed. Then (3.13) can be rewritten as

$$s_{u_j(k)}^{y_l(k+1)}\Bigg|_{u_j(k)=u_{j,0}} = \frac{\partial y_l/y_l}{\partial u_j/u_{j,0}} = \frac{\partial y_l}{\partial u_j}\Bigg|_{u_{j,0}} \cdot \frac{u_j(k)}{y_l(k+1)}. \tag{3.14}$$

For small changes Δu_j from $u_{j,0}$ such that $u_j(k) = u_j(k-1) + \Delta u_j$,

$$\frac{\partial y_l}{\partial u_j}\Bigg|_{u_{j,0}} \approx \frac{\Delta y_l(k)}{\Delta u_j(k-1)}\Bigg|_{u_{j,0}} = \frac{y_l(k+1) - y_l(k)}{\Delta u_j}$$

and the sensitivity function can be rewritten as

$$s_{u_j(k)}^{y_l(k+1)}\Bigg|_{u_j(k)=u_{j,0}} \approx \frac{\Delta y_l(k)}{\Delta u_j(k-1)}\Bigg|_{u_{j,0}} \cdot \frac{u_j(k)}{y_l(k+1)}. \tag{3.15}$$

The sensitivity function is based on the aggregated output of the fuzzy model rather than on the outputs of its local submodels. Therefore the fuzzy model has to be simulated explicitly to obtain the predicted model outputs. Using the above expression, we can obtain static as well as dynamic coupling between output y_l and input u_j. The step response of the fuzzy model is obtained for a step Δu_j, and $s_{u_j(k)}^{y_l(k+\tau)}$ are calculated, where $\tau = 1, 2, \ldots, n$ are the points in the step response. The quantity $s_{u_j(k)}^{y_l(k+n)}$ indicates the static coupling, while the dynamic coupling is given through the series $s_{u_j(k)}^{y_l(k+1)}, \ldots, s_{u_j(k)}^{y_l(k+n)}$.

The second fraction in (3.15) plays a scaling role, making $s_{u_j(k)}^{y_l(k+1)}$ not an absolute but a relative quantity. If only the first term in (3.15) is used, then the contribution of the remaining inputs and delayed inputs to $y_l(k+1)$ is neglected. Since (3.15) deals with one output at a time, all outputs can be analysed consecutively with respect to all inputs. Note that here the indicated coupling for 2×2 processes is not necessarily symmetric, as for RGA.

For linear processes the degree of coupling is invariant with respect to both the operating point and the amplitude of the input change. This does not hold for the nonlinear case, however. The sensitivity of the fuzzy model is thus analysed separately at each point in the variable domain, providing different input variations. This is done by varying a single input with a particular amplitude at a time while keeping the other inputs constant and interpolating between the amplitude values. One can further

extend this interpolation by slightly changing yet another input. This way the output sensitivity to a given input can be viewed as a function of another input.

The information obtained through the sensitivity function has a slightly different interpretation than that of the RGAs. Large $s_{u_j}^{y_l}$ (positive or negative) shows strong coupling between input u_j and output y_l. Positive $s_{u_j}^{y_l}$ means that y_l and u_j change in the same direction, negative $s_{u_j}^{y_l}$ indicates that they change in the opposite direction. Values close to zero show that y_l is weakly influenced by u_j or not at all.

The following Example 3.3 illustrates how we compute the sensitivity functions for the fuzzy model used in Example 3.1. Note, however, that there is no difference in the computational procedure if the modified fuzzy model (Example 3.2) is used instead.

Example 3.3 *Assume that the fuzzy model is in a steady-state operation at* $(y_{10}, y_{20}) = (y_1(k), y_2(k))$ *and* $(u_{10}, u_{20}) = (u_1(k-1), u_2(k-1))$. *Let a variation in input* $u_1(k)$, $u_1(k) = u_{10} + \Delta u_1$ *be introduced while* u_2 *is kept constant. The output reaction at the time instant* $k+1$ *is given by*

$$y_1(k+1) = \sum_{i=1}^{4} w_{1i}(\mathbf{x}_0)\left(\zeta_{1i1}y_{10} + \eta_{1i1}(u_{10} + \Delta u_1) + \eta_{1i2}u_{20} + \theta_{1i}\right)$$
$$y_2(k+1) = \sum_{i=1}^{4} w_{2i}(\mathbf{x}_0)\left(\zeta_{2i1}y_{20} + \eta_{2i1}(u_{10} + \Delta u_1) + \eta_{2i2}u_{20} + \theta_{2i}\right), \qquad (3.16)$$

where $\mathbf{x}_0 = (y_{10}, y_{20})$. *The sensitivity of* y_1 *with respect to* Δu_1 *is then*

$$s_{u_1(k)}^{y_1(k+1)} = \frac{y_1(k+1) - y_1(k)}{\Delta u_1(k)} \cdot \frac{u_1(k)}{y_1(k+1)}.$$

The sensitivity of y_2 *with respect to* Δu_1 *can be computed analogously. Below a matrix is shown that has as elements the sensitivity functions of the two outputs (column-wise) with respect to both inputs (row-wise)*

$$\mathbf{S}_{\mathbf{u}(k)}^{\mathbf{y}(k+1)} = \begin{bmatrix} \frac{y_1(k+1)-y_1(k)}{\Delta u_1(k)} \cdot \frac{u_1(k)}{y_1(k+1)} & \frac{y_1(k+1)-y_1(k)}{\Delta u_2(k)} \cdot \frac{u_2(k)}{y_1(k+1)} \\ \frac{y_2(k+1)-y_2(k)}{\Delta u_1(k)} \cdot \frac{u_1(k)}{y_2(k+1)} & \frac{y_2(k+1)-y_2(k)}{\Delta u_2(k)} \cdot \frac{u_2(k)}{y_2(k+1)} \end{bmatrix}.$$

The sensitivity analysis thus gives additional insight in the control design. Contrary to the RGAs, the sensitivity functions can indicate non-symmetrical coupling occurring between the control loops. Further, not only the static but also the dynamic coupling between outputs and inputs can be represented that characterizes the intermediate effects of an input change on the considered output.

3.3 Decoupling control design

We can use the input-output analysis (carried out on the process model) to decide whether the interactions between the process outputs can remain unchanged or should be compensated for. In the latter case, one may either try to cut off the coupling paths or to add a compensation path in parallel to the coupling path which has the

same dynamics but with an opposite sign to eliminate the interactions. However, both methods can only be applied in block diagrams. Therefore, for a coupled multivariable process, the decoupling elements are usually incorporated into the control system.

The purpose of decoupling is to reduce undesirable effects of interactions in the process. Most frequently it is accomplished by inverting the model and using the inverted model to compensate for the coupling (Isidori, 1995). The decoupling elements have essentially coupling effects that compensate for the original coupling problem (Fig. 3.5). Moreover, in nonlinear multivariable systems, the decoupling also compensates for the nonlinearities, making the system linear with regard to the reference signal. This technique is also known as *input-output linearization.*

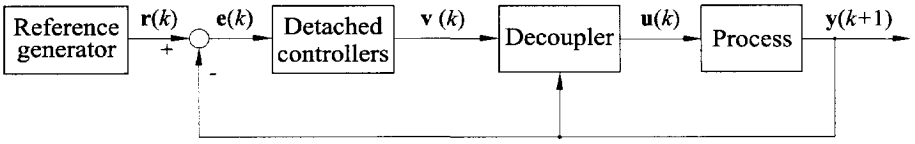


Figure 3.5. Input-output decoupling scheme.

Different applications of decoupled MIMO control have been reported, based on fuzzy decouplers see, e.g., (Reay et al., 1995; Kang and Lei, 1996). Reay et al. (1995) used a linguistic fuzzy system for inverse-based control of a four-phase switched reluctance motor that removes the coupling effects between the phase current, the rotor position and overlap angle between the rotor and stator poles. Kang and Lei (1996) presented an application of fuzzy control for temperature control of the heating system for the barrel of an injection moulding machine. However, the majority of the reported applications of decoupled fuzzy control (including the ones mentioned above) use the heuristic approach for the decoupler design. The decoupling methods proposed here on the other hand are based on the fuzzy model, which makes the design more systematic and less application dependent.

Due to the nonlinearities in the TS models, there are no guarantees that an analytic decoupling law can be obtained in the general case. Such an analytic decoupling law can be derived for an affine fuzzy model, i.e., the model is linear with respect to the input signal. To deal with non-affine fuzzy models, we suggest two other methods. One exploits the structure of the fuzzy model to provide the decoupling law, while the other uses numerical optimization to invert the model.

3.3.1 Decoupling of affine TS fuzzy models

Currently for continuous-time affine systems there is a commonly used theory for input-output decoupling (Nijmeijer and van der Schaft, 1990; Isidori, 1995), but unfortunately, it cannot be applied to discrete-time systems. In the discrete-time case, the system dynamics are represented by differences rather that derivatives. As a result, the affineness of a discrete-time system does not provide extra tools to solve nonlinear control problems – the chain rule for the calculation of derivatives cannot be applied to differences. Therefore, the "algebraic approach" of the differential geometry theory

(Fliess et al., 1983; Monaco and Normand-Cyrot, 1995; Kotta, 1995) is used to solve the discrete-time decoupling problem (Mollov, Babuška and Verbruggen, 2000a). The basic idea of the algebraic approach is the same as in the continuous-time case: obtain a second discrete-time nonlinear system such that when it is placed in front of the original system, the outputs of the original system are equal to the inputs of the second system. The underlying theory is summarized in Appendix B.

An analytical solution to the decoupling problem for a general (non-affine) TS fuzzy model is difficult to obtain, as shown in Example 3.4.

Example 3.4 *Consider a* 2×2 *non-affine MIMO fuzzy model with* K_1 *and* K_2 *rules for the first and the second output, respectively*

$$y_1(k+1) = \frac{1}{\sum_{i=1}^{K_1} \beta_{1i}(k)} \sum_{i=1}^{K_1} \beta_{1i}(k)[\zeta_{1i1} y_1(k) + \zeta_{1i2} y_2(k) + \eta_{1i1} u_1(k) \eta_{1i2} u_2(k) + \theta_{1i}]$$

$$y_2(k+1) = \frac{1}{\sum_{i=1}^{K_2} \beta_{2i}(k)} \sum_{i=1}^{K_2} \beta_{2i}(k))[\zeta_{2i1} y_1(k) + \zeta_{2i1} y_2(k) + \eta_{2i1} u_1(k) \eta_{2i2} u_2(k) + \theta_{2i}]$$

where

$$\beta_{li}(k) = \mu_{i1}\big(y_1(k)\big)\mu_{i2}\big(y_2(k)\big)\mu_{i1}\big(u_1(k)\big)\mu_{i2}\big(u_2(k)\big).$$

Because the membership functions nonlinearly depend on the inputs (recall (2.7)), a system of nonlinear algebraic equations results that has to be explicitly solved for $\mathbf{u}(k)$ *substituting* $\mathbf{y}(k+1) = \mathbf{y}_{\text{ref}}(k+1)$. *Even for* $K_l = 2$, $l = 1, 2$, *the analytic solution is cumbersome (if one exists at all).*

Therefore as a first step we consider affine fuzzy models with the input-output representation

$$\mathbf{y}(k+1) = \mathcal{F}\big(\mathbf{x}(k)\big) + \mathcal{G}\big(\mathbf{x}(k)\big)\mathbf{u}(k). \tag{3.17}$$

Here, the regression vector in the premise part of the fuzzy rule (2.4), $\mathbf{x}_l$ does not contain the current input $\mathbf{u}(k)$, i.e. the DOFs are a function of the state vector $\mathbf{x}(k)$ (recall (2.7))

$$\beta_{li}\big(\mathbf{x}_l(k)\big) = \prod_{h=1}^{\rho} \mu_{\mathcal{A}_{li,h}}(x_{lh}).$$

From (2.6) and (3.17)

$$\mathcal{F}_{l,1}\big(\mathbf{x}(k)\big) = \frac{\sum_{i=1}^{K_l} \beta_{li}\big(\mathbf{x}_l(k)\big)\big(\boldsymbol{\zeta}_{li}\mathbf{x}_l(k) + \theta_{li}\big)}{\sum_{i=1}^{K_l} \beta_{li}\big(\mathbf{x}_l(k)\big)} \tag{3.18}$$

$$\mathcal{G}_{l,j}\big(\mathbf{x}(k)\big) = \frac{\sum_{i=1}^{K_l} \beta_{li}\big(\mathbf{x}_l(k)\big)\boldsymbol{\eta}_{li,j}}{\sum_{i=1}^{K_l} \beta_{li}\big(\mathbf{x}_l(k)\big)}. \tag{3.19}$$

$$l = 1, \ldots, p, \quad j = 1, \ldots, m.$$

Then the necessary and sufficient condition for the existence of an input-output decoupling law amounts to rank $\mathcal{G} = p$, provided that $p \leq m$, see Theorem B.1. As a result,

a necessary (but not sufficient) condition for decoupling of an affine TS model is that at least a single rule has to be active at one moment in time. This implies that in every row of $\mathcal{G}$ there will be at least a single non-zero element.

Provided that the necessary and sufficient condition is satisfied, the input-output decoupling is given through

$$\mathbf{u}_{\text{ref}}(k) = \mathcal{G}^{-1}\big(\mathbf{x}(k)\big)\big(\mathbf{y}_{\text{ref}}(k) - \mathcal{F}(\mathbf{x}(k))\big), \tag{3.20}$$

where $\mathbf{y}_{\text{ref}}(k)$ is some desired reference signal to be followed.

3.3.2 Decoupling of general (non-affine) TS fuzzy models

When the TS model is a non-affine one, we can on-line at each sampling instant find an analytic decoupler that corresponds to the linear time-varying model (2.9) currently in use. Another possibility is to seek a numerical solution by solving a nonlinear optimization problem, which is a particular case of the one utilized in model predictive control.

Analytic decoupling. Regarding the first approach, the problem is that the control input non-linearly enters the TS model (see Example 3.4), so that the model cannot be straightforwardly inverted. To design a decoupler, we consider the fuzzy model to be linear time varying, rather than a nonlinear time-invariant one. In such a setup, the decoupler is redesigned at each sampling instant using the current linear model (Fig. 3.5). The decoupling algorithm is summarized in Algorithm 3.1.

Numeric decoupling. The numeric decoupling law is obtained by solving a nonlinear optimization problem. The cost function used is a particular case of the cost function utilized in Model Predictive Control (MPC) in Chapter 4

$$\min_{\mathbf{u}} J = \sum_{i=1}^{N} \big\|\mathbf{v}(k+i) - \mathbf{y}(k+i)\big\|_{\mathbf{P}}^2, \tag{3.22}$$

where $\mathbf{y}(k+i)$ is the output of the fuzzy model, $\mathbf{v}(k+i)$ is the output reference (of length N) to be followed. If a suitably designed reference is applied, the only deviations of $\mathbf{y}(k+i)$ from $\mathbf{v}(k+i)$ are due to loop interactions. In this way, the optimal sequence $\mathbf{u} = \arg\min J$ will eliminate the coupling as much as possible. The relative importance of the outputs is determined by the weighting matrix $\mathbf{P}$. For $N = 1$ a static decoupler results, while $N > 1$ leads to a dynamic one (Isidori, 1995). The term dynamic decoupler is used here in the sense that for $N > 1$ the control signal $\mathbf{u} = \arg\min J$ depends on the model prediction, where the model states are initialized with previous and current outputs and previous inputs.

An additional computational problem arises when constraints on the process inputs are present. In such a case the constraints first have to be propagated back through the fuzzy model to the primary controller that generates $\mathbf{v}$. Different approaches have been proposed to address the problem in this situation, see (Nevistić and Morari, 1995; de Oliveira et al., 1995; Botto et al., 1999):

Algorithm 3.1 (Time-varying input-output decoupling)

Step 1: **Compute** the degree of fulfillment $\beta_{li}(\boldsymbol{\chi}_l(k))$ according to (2.7). Note that $\boldsymbol{\chi}_l(k) = \left[\mathbf{x}_l^T(k), \mathbf{u}^T(k-1)\right]^T$, since $\mathbf{u}(k)$ is not available yet.

Step 2: **Compute** the current linear model's parameters $\boldsymbol{\zeta}_l(\boldsymbol{\chi}_l(k))$ and $\boldsymbol{\eta}_l(\boldsymbol{\chi}_l(k))$ according to (2.10).

Step 3: **Set up** the transfer function matrix $\mathbf{G}(k)$ (recall (3.4))

$$\mathbf{G}(k) = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1m} \\ g_{21} & g_{22} & \cdots & g_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ g_{p1} & g_{p2} & \cdots & g_{pm} \end{bmatrix}, \tag{3.21}$$

where

$$g_{lj} = \frac{\boldsymbol{\eta}_{l,j}(\boldsymbol{\chi}_l(k))u_j(k-1)}{1 - \boldsymbol{\zeta}_l(\boldsymbol{\chi}_l(k))\boldsymbol{\chi}_l(k)} \quad l = 1,\ldots,p,\ j = 1,\ldots,m.$$

Step 4: **The decoupling law** is

$$\mathbf{u}(k) = \mathbf{G}^{-1}(k)\mathbf{v}(k),$$

where $\mathbf{v}(k)$ is some desired signal, e.g., desired reference trajectory, or controller output as in Fig. 3.5.

- Constraints on the first move only. This is the simplest approximation which leads to suboptimal control moves. The future control moves may violate the constraints but they are not applied.

- Identical constraints on the first and future moves. Its disadvantage is that unnecessarily conservative control moves may result.

- Use first- or second-order approximations of the nonlinear map between $\mathbf{u}$ and $\mathbf{y}_{\text{ref}}$ over the complete prediction horizon. To do this, compute the complete expansion of the nonlinear constraint relation over the prediction horizon and then perform linearization around the different points. In this way, feasible control moves are guaranteed without the conservatism of the identical constraints.

By comparison, the input-output decoupling is an inherent characteristic of the MPC controller (Chapter 4), which depends on (*i*) the model dynamics and (*ii*) constrains on the change rates of the control inputs.

3.4 An example with simulated and real-time liquid level control in a MIMO cascaded-tanks setup

The laboratory-scale cascaded-tanks setup shown in Fig. 3.6 is used as an example to illustrate the discussed methods and demonstrate their real-time performance. The control objective is to follow set-point changes for the liquid levels $\mathbf{h} = [h_1, h_2]^T$ in the lower two tanks by adjusting the flow rates $\mathbf{q} = [q_1, q_2]^T$ of the liquid entering the upper tanks. Coupling is introduced by a non-symmetric cross-connection between the upper and lower tanks: the cross-sectional area of the pipe connecting tank three with tank two is 10% larger than the one between tank four and tank one, respectively 15.21 mm^2 and 12.57 mm^2. The cross-sectional areas of the pipes connecting tank three with tank one and tank four with tank two are 36.32 mm^2. All tanks have a length of 0.5 m and the total height of the setup is 1.5 meters. The liquid pumps are located at the ground level of the setup and the liquid is pushed up to the top of the upper tanks via plastic pipes. As a result, the pumps have to overcome the weight of the liquid in the pipes. This introduces directionality to the step response of the process.

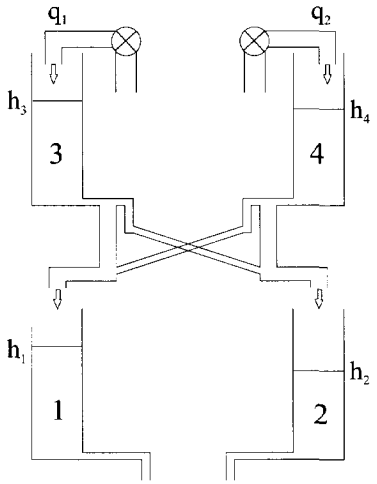


Figure 3.6. Four cascaded-tanks setup.

3.4.1 *Fuzzy modeling*

A TS model is obtained by using experimental input-output data, sampled with period $T_s = 5$ s. The structure of the model is defined using the physical structure of the system: second-order models for both outputs with one sampling-time delay in the inputs during which the water flows down through the tanks. The fuzzy rules derived by using the clustering algorithm presented in Appendix A are given in (3.23). The antecedent membership functions $A_{li,h}$ are Gaussian, with membership functions centers shown in Tab. 3.1. The performance of the model on a validation data set can be seen in Fig. 3.7.

Output 1

1. **If** $h_1(k)$ is $\mathcal{A}_{11,1}$ **and** $h_1(k-1)$ is $\mathcal{A}_{11,2}$ **and** $q_1(k)$ is $\mathcal{A}_{11,3}$ **and** $q_1(k-1)$ is $\mathcal{A}_{11,4}$ **and** $q_2(k)$ is $\mathcal{A}_{11,5}$ **then**
$h_1(k+1) = 0.919h_1(k) - 0.156h_1(k-1) + 0.024q_1(k) + 0.075q_1(k-1) + 0.015q_2(k) + 0.007$
2. **If** $h_1(k)$ is $\mathcal{A}_{12,1}$ **and** $h_1(k-1)$ is $\mathcal{A}_{12,2}$ **and** $q_1(k)$ is $\mathcal{A}_{12,3}$ **and** $q_1(k-1)$ is $\mathcal{A}_{12,4}$ **and** $q_2(k)$ is $\mathcal{A}_{12,5}$ **then**
$h_1(k+1) = 1.13h_1(k) - 0.314h_1(k-1) + 0.032q_1(k) + 0.171q_1(k-1) + 0.048q_2(k) - 0.0512$
3. **If** $h_1(k)$ is $\mathcal{A}_{13,1}$ **and** $h_1(k-1)$ is $\mathcal{A}_{13,2}$ **and** $q_1(k)$ is $\mathcal{A}_{13,3}$ **and** $q_1(k-1)$ is $\mathcal{A}_{13,4}$ **and** $q_2(k)$ is $\mathcal{A}_{13,5}$ **then**
$h_1(k+1) = 1.85h_1(k) - 0.804h_1(k-1) + 0.01q_1(k) - 0.041q_1(k-1) - 0.003q_2(k) - 0.008$

(3.23a)

Output 2

1. **If** $h_2(k)$ is $\mathcal{A}_{21,1}$ **and** $h_2(k-1)$ is $\mathcal{A}_{21,2}$ **and** $q_1(k)$ is $\mathcal{A}_{21,3}$ **and** $q_2(k)$ is $\mathcal{A}_{21,4}$ **and** $q_2(k-1)$ is $\mathcal{A}_{21,5}$ **then**
$h_2(k+1) = 1.02h_2(k) - 0.251h_2(k-1) + 0.012q_1(k) - 0.002q_2(k) + 0.054q_2(k-1) + 0.0082$
2. **If** $h_2(k)$ is $\mathcal{A}_{22,1}$ **and** $h_2(k-1)$ is $\mathcal{A}_{22,2}$ **and** $q_1(k)$ is $\mathcal{A}_{22,3}$ **and** $q_2(k)$ is $\mathcal{A}_{22,4}$ **and** $q_2(k-1)$ is $\mathcal{A}_{22,5}$ **then**
$h_2(k+1) = 1.35h_2(k) - 0.498h_2(k-1) + 0.023q_1(k) + 0.005q_2(k) + 0.059q_2(k-1) - 0.0104$
3. **If** $h_2(k)$ is $\mathcal{A}_{23,1}$ **and** $h_2(k-1)$ is $\mathcal{A}_{23,2}$ **and** $q_1(k)$ is $\mathcal{A}_{23,3}$ **and** $q_2(k)$ is $\mathcal{A}_{23,4}$ **and** $q_2(k-1)$ is $\mathcal{A}_{23,5}$ **then**
$h_2(k+1) = 1.86h_2(k) - 0.929h_2(k-1) - 0.005q_1(k) - 0.006q_2(k) + 0.015q_2(k-1) + 0.0302$

(3.23b)

Table 3.1. Four cascaded-tanks setup. Centers of the fuzzy membership functions.

output, rule (l,j)	$\mathcal{A}_{lj,1}$	$\mathcal{A}_{lj,2}$	$\mathcal{A}_{lj,3}$	$\mathcal{A}_{lj,4}$	$\mathcal{A}_{lj,5}$
1,1	0.103	0.103	0.164	0.160	0.304
1,2	0.200	0.200	0.379	0.379	0.293
1,3	0.256	0.255	0.429	0.432	0.338
2,1	0.089	0.088	0.305	0.159	0.151
2,2	0.173	0.175	0.313	0.363	0.365
2,3	0.267	0.267	0.371	0.454	0.461

To give a quantitative measure of the model accuracy, two performance indices are used: Root Mean Square (RMS) error and Variance Accounted For (VAF) in %, which values are RMS = $[0.0129, 0.0217]^T$ and VAF = $[99.0191, 97.6697]^T$, respectively.

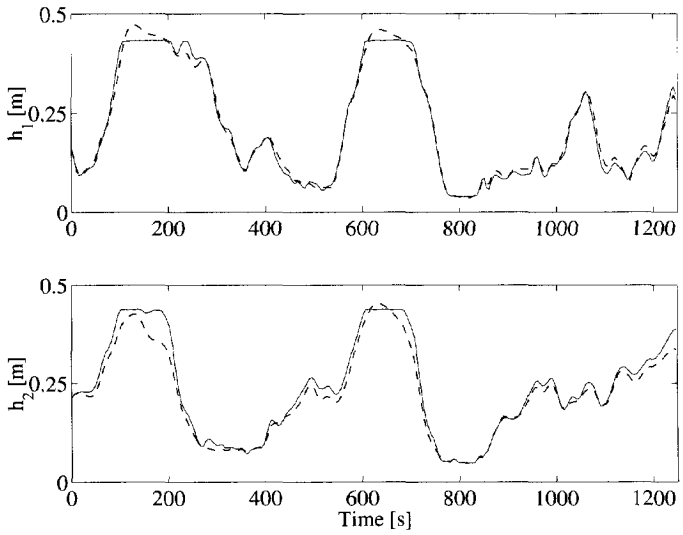


Figure 3.7. Four cascaded-tanks setup. Validation of the fuzzy model. Top: level h_1, bottom: level h_2. Solid line: process output, dashed line: model prediction. The performance indices are RMS $= [0.0129,\ 0.0217]^T$ and VAF $= [99.0191,\ 97.6697]^T$.

3.4.2 Analysis of interactions

RGA analysis. Since the antecedent structure of the fuzzy model (3.23) differs for the two outputs, it is not possible to match the corresponding rules. Therefore it is necessary to provide the points for which the RGA has be computed. Three such points are given in Tab. 3.2. The output values are used later as a desired reference to be followed by the control system, which is used to test the methods developed in the current and subsequent chapters.

To illustrate the RGA computation, consider the operating point that corresponds to the first fuzzy rules for both outputs. The degrees of fulfillment for the outputs are $\beta_1 = [0.3333, 0.9925, 0.3006]$ and $\beta_2 = [0.0093, 0.9945, 0.1068]$. The local model is (recall (2.9))

Table 3.2. Input and output values for three operating points. The control signal is given in electrical units in the interval $[0, 1]$, and the liquid level is given in meters.

Operating point	1	2	3
h_1	0.15	0.25	0.35
h_2	0.12	0.15	0.18
q_1	0.36	0.38	0.40
q_2	0.27	0.36	0.40

$$h_1(k+1) = 1.2189h_1(k) - 0.3719h_1(k-1) + 0.0264q_1(k) + 0.1122q_1(k-1) + 0.0319q_2(k) - 0.0314$$
$$h_2(k+1) = 0.9549h_2(k) - 0.3667h_2(k-1) + 0.0140q_1(k) + 0.0028q_2(k) + 0.0377q_2(k-1) - 0.0044$$

Both MISO submodels are stable, with poles $z_{1,2} = 0.6095 \pm i0.0217$ and $z_{3,4} = 0.6095 \pm i0.0217$, respectively. The steady-state transfer matrix $\mathbf{G}$ is

$$\mathbf{G} = \begin{pmatrix} 1.1047 & 0.3801 \\ 0.2270 & 0.5717 \end{pmatrix}$$

and the resulting Relative Gain Array

$$\Lambda = \mathbf{G} .* \left(\mathbf{G}^{-1}\right)^T = \begin{pmatrix} 1.1583 & -0.1583 \\ -0.1583 & 1.1583 \end{pmatrix}.$$

The RGAs for the different operating points are shown in Tab. 3.3. Looking at the λs, deviations are encountered that show varying coupling at different levels. Nevertheless, all RGAs are diagonal dominant, indicating pairing between the left pump and the left tank, and between right pump and right tank. The presence of negative λs at the first two operating points indicates that the two control loops are mutually destabilizing. For the third point, the interaction leads to extra oscillations, see Fig. 3.10a where the system performance achieved through decentralized PI controllers is shown. At the second point, the coupling (-0.0962) can be neglected, i.e., no decoupler is necessary here. For the two other points, however, such a decoupler should be designed.

Table 3.3. RGA for the operating points of Tab. 3.2.

rule	Λ	
1	1.1583	-0.1583
	-0.1583	1.1583
2	1.0962	-0.0962
	-0.0962	1.0962
3	0.8947	0.1053
	0.1053	0.8947

Sensitivity analysis. Figure 3.8 shows the static input-output interactions for a 50 steps-ahead simulation after a change in either input with an amplitude of 0.0345. At the moment of the input variation, the process is in a steady-state operation for the corresponding combination of q_1 and q_2. The following can be observed from the figure: (*i*) the influence of q_2 on h_1 is significantly less than the influence of q_1 (Fig. 3.8a),

and the influence of q_1 on h_2 is significantly less than the one of q_2 (Fig. 3.8b); (ii) there are some differences between the influence of the q_1 on h_1 and that of q_2 on h_2, which result from the non-symmetric coupling. Tab. 3.4 gives the summarized values for the sensitivity analysis presented in Fig. 3.8.

Note that in all the four figures shown, there is a relative sensitivity peak at the low levels. This can be explained through the observation that at the low levels, a change in one of the outputs requires (relatively) big input changes that influence the other output, compare with Tab. 3.2. There are also a couple of peaks visible in the figures. In Fig. 3.8a top, a mound in $s_{q_1(k)}^{h_1(k+50)}$ is present that shows higher sensitivity of h_1 to combinations of q_1 and q_2. The phenomenon is due to the flow of the liquid in the tanks: at certain levels in the upper tanks, vortices appear that swiftly suck the liquid in the lower tanks. Note that the strength of this phenomenon varies in the different tanks.

Table 3.4. Aggregated values for the steady-state sensitivity analysis of the cascaded-tanks setup.

$s_{q_j}^{h_i}$	$Mean$	Max	Min
$s_{q_1}^{h_1}$	1.1	3.23	0.17
$s_{q_1}^{h_2}$	0.52	2.76	−0.57
$s_{q_2}^{h_1}$	0.46	4.56	−0.62
$s_{q_2}^{h_2}$	1.12	4.38	−0.59

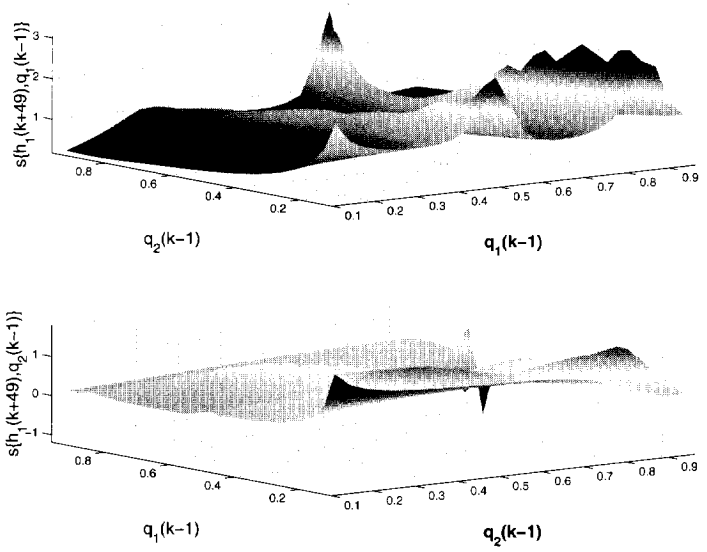
To give an impression of the dynamic coupling, Fig. 3.9 shows the series $s_{q_j(k)}^{h_i(k+1)}$, $\ldots, s_{q_j(k)}^{h_i(k+50)}$ for $i,j = 1,2$ at level $\mathbf{h} = [0.209, 0.187]^T$m (which corresponds to $\mathbf{q} = [0.5, 0.5]^T$) for variations in the inputs with amplitude of 0.0345. The non-symmetric steady coupling is clearly visible, as the influence of q_1 on h_2 $\left(s_{q_1}^{h_2}\right)$ is greater than the influence of q_2 on h_1 $\left(s_{q_2}^{h_1}\right)$. The relative type of the sensitivity measure used makes the quantities independent from the absolute values at which it is computed.

3.4.3 Decoupling control

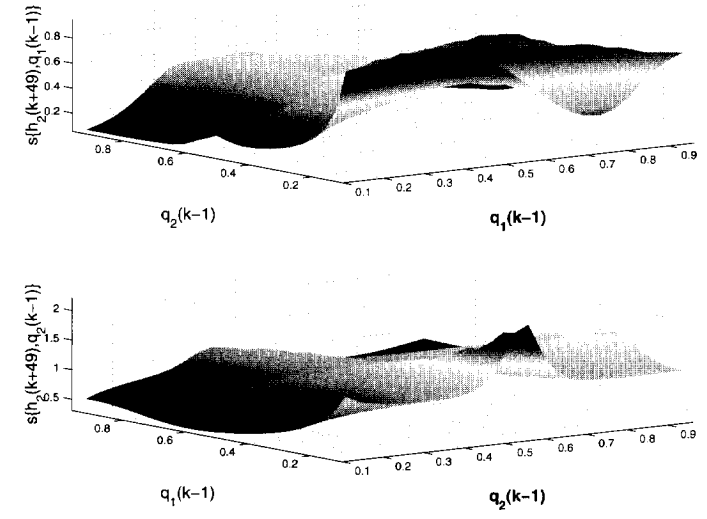
Decoupling control with a non-affine fuzzy model. Despite being only a 2×2 system, the cascaded-tanks setup is difficult to control by means of a decentralized control structure. Figure 3.10 shows the performance achieved by using two detached digital PI controllers (with parameters[2] given in Tab. 3.5)

$$u(k) = \left(K_p + \frac{T_s K_i}{z-1}\right) e(k),$$

[2]The controller parameters are initially tuned with no coupling present, i.e., without the cross connection between the channels. Then, with the cross connection, they are fine-tuned to avoid an agitated response.



(a) Lower left tank.



(b) Lower right tank.

Figure 3.8. Four cascaded-tanks setup. Static coupling for 50-step ahead simulation of the TS model: (a) Lower left tank, (b) Lower right tank.

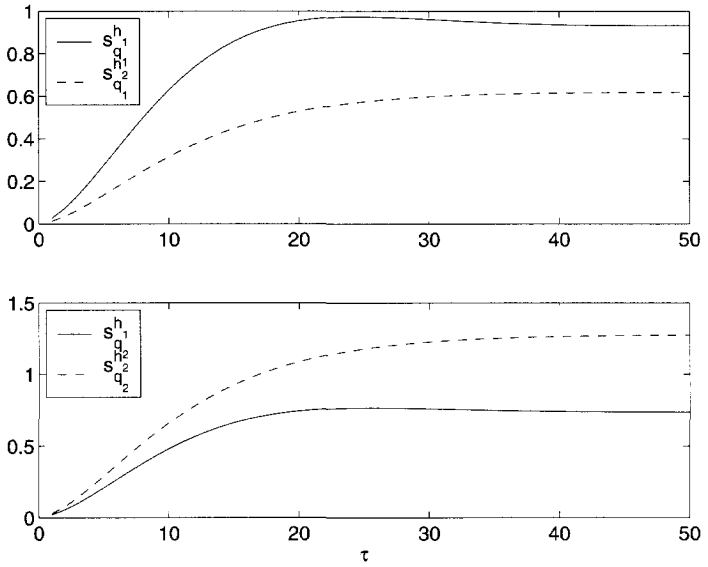


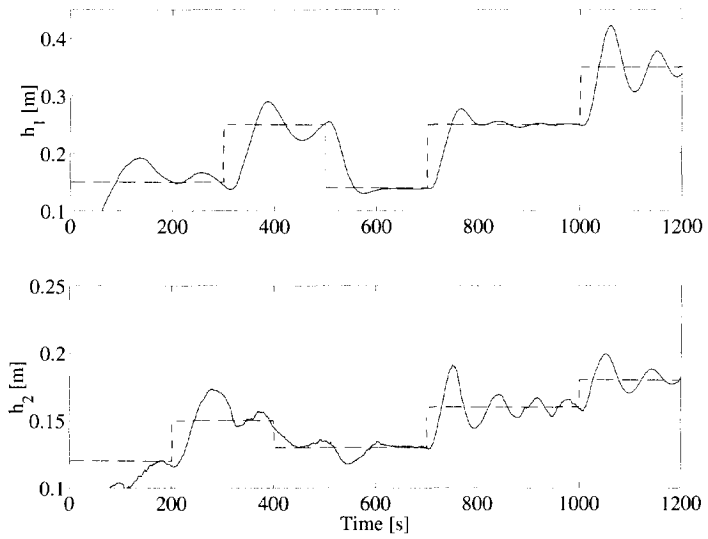
Figure 3.9. Four cascaded-tanks setup. Dynamic coupling for inputs changes with amplitude 0.0345 at steady level $\mathbf{h} = [0.209, 0.187]^T$m, corresponding to $\mathbf{q} = [0.5, 0.5]^T$.

where K_p and K_i are the proportional and integral gains, respectively, and $T_s = 5$ s is the sampling period. Although the controllers are tuned for a smooth rather than fast response, they cannot compensate for the coupling effects following a change in either output reference. The observed behaviour corresponds to the one indicated by the RGA analysis.

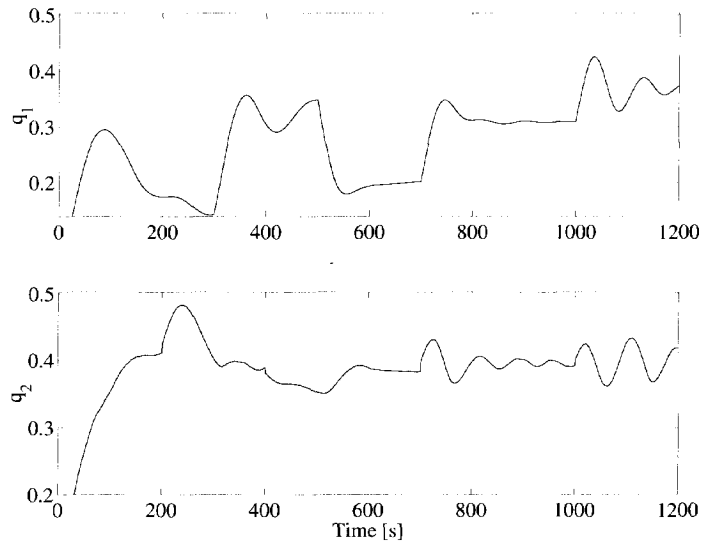
The decoupling scheme depicted in Fig. 3.5 is used to eliminate the coupling. The outputs of the detached PI controllers are fed to a time-varying decoupler based on the fuzzy model (3.23). Using the model, at each sampling instant the input-output gains are calculated, the corresponding transfer matrix is inverted and placed between the PIs and the system. Note that here different tuning is used for the controller parameters (Tab. 3.5), since the decoupler changes the system's properties. In this sense the PI controllers are tuned with respect to the augmented "decoupler & setup" system.

The system outputs and the control signals are given in Fig. 3.11. Comparing the results achieved without a decoupler (Fig. 3.10a) with those achieved with a decoupler (Fig. 3.11a), one can see that the improvement is considerable. While one may be tempted to attribute such a performance gain to the tuning of the controllers, it is necessary to realize that the decoupler does not only cancel the coupling but also compensates for the nonlinearities in the setup, making the augmented system linear.

Decoupling control with an affine fuzzy model. To illustrate the decoupling scheme for affine TS models, we constructed an affine model of the cascaded-tanks setup

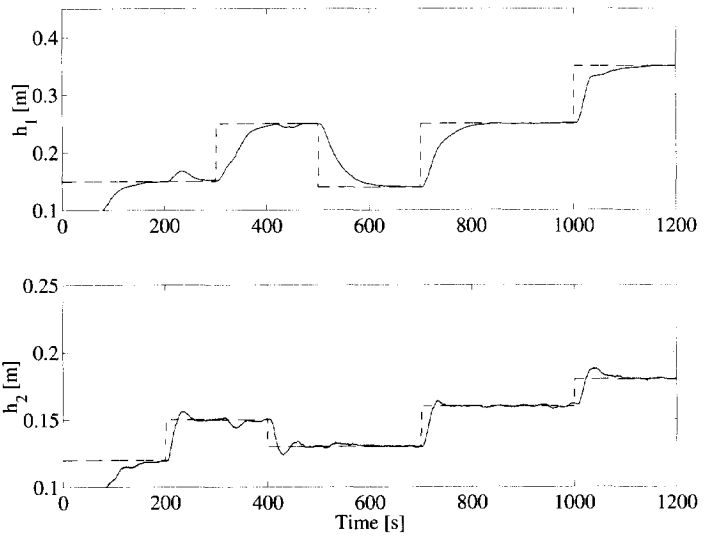


(a) System outputs. Top: level h_1, bottom: level h_2. Dashed line: output reference, solid line: setup output.

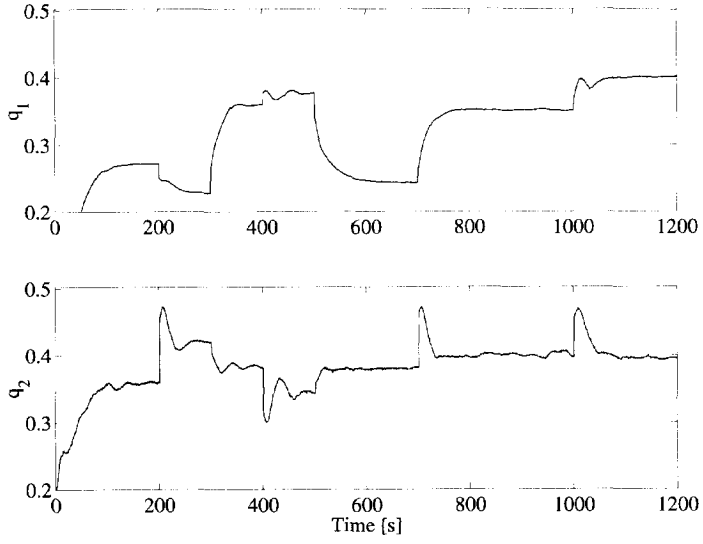


(b) Control inputs. Top: q_1, bottom: q_2.

Figure 3.10. Real-time decentralized PI control of the cascaded-tanks setup.



(a) System outputs. Top: level h_1, bottom: level h_2. Dashed line: output reference, solid line: setup output.



(b) Control inputs. Top: q_1, bottom: q_2.

Figure 3.11. Real-time decoupled PI control of the cascaded-tanks setup.

Table 3.5. PI parameters. The subscripts *ND* and *D* stand for *Not Decoupled* and *Decoupled*, respectively.

PI	$K_{ND,p}$	$K_{ND,i}$	$K_{D,p}$	$K_{D,i}$
Left	0.02	0.0025	0.05	0.045
Right	0.105	0.00425	0.45	0.0750

based on the input-output data. The affineness of the fuzzy model means that inputs of the system appear only in the consequence part, but not in the premises of the fuzzy model.

The general (non-affine) TS fuzzy model (3.23) is used to simulate the setup. Although structurally identical to the model employed in the controller, the presence of the current inputs in the rule premises of the process model results in a considerable difference between the two.

The decoupling method described in Section 3.3.1 is applied in combination with a linear model predictive controller (Mollov, Babuška and Verbruggen, 2000a). The predictive controller is used to provide a desired reference, taking into account the input and output constraints (see Chapter 4). The outputs are limited to the interval of $[0, 0.5]$ m imposed by the real process. The input constraints of $[0, 1]$ reflect the capacity of the water pumps. The linear model used in the predictive controller is derived at $(h_1, h_2) = (0.2, 0.15)$ m and is used throughout the simulation. A prediction horizon of $H_\mathrm{p} = 11$ and a control horizon of $H_\mathrm{c} = 1$ are used and the results are presented in Fig. 3.12 through Fig. 3.14. To illustrate the influence of the input constraints on the achievable decoupling, we consider three cases discussed in Section 3.3.2: (i) no input constraints; (ii) input constraints on the first control move only, and (iii) constant input constraints over the complete prediction horizon.

As can be observed from the figures, when the input is not restricted, full decoupling is achieved. The changes in one of the outputs do not reflect on the other. This, however, requires quite fast changes in both inputs. As a consequence, in the presence of input constraints, slight coupling remains (Fig. 3.13) (Note that the influence of the input constraints on the coupling is also the same in the MPC controller, see Chapter 4). Similar performance can be achieved by penalizing the control changes in the optimized cost function. The influence of the constraints can be assessed by comparing controller performance with constraints on the first move only and constant constraints over the complete prediction interval (Fig. 3.14). In the latter case the response is much more sluggish, but again almost complete decoupling is achieved.

Note that the computational complexity in the presented analytic decoupling methods is very low. The decoupling algorithm used for a non-affine TS model (Algorithm 3.1) results in a matrix inversion which can be calculated either off-line or on-line at each sampling instant. Comparable, although computationally a bit more demanding (because of the way in which the matrix-to-be-inverted is composed) is the decoupling method for an affine TS model. Also here the matrix $\mathcal{G}(\mathbf{x}(k))$ (in

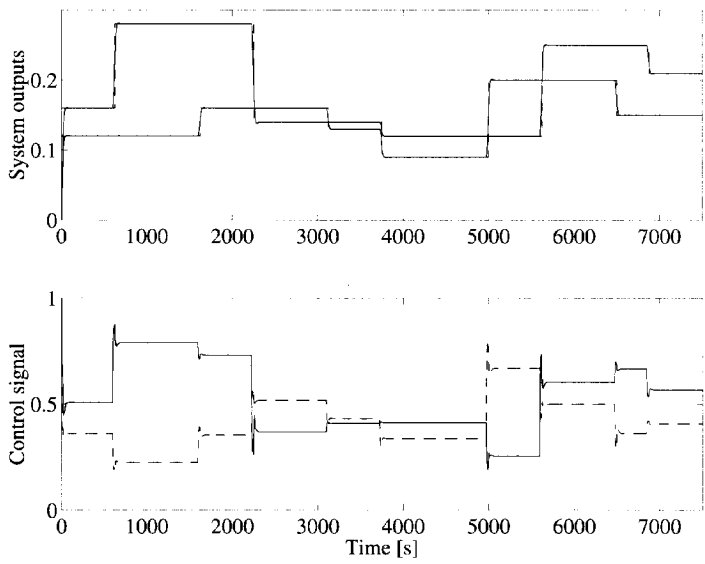


Figure 3.12. Simulated decoupled control of the cascaded-tanks setup. System outputs and control signals when no input constraints are present.

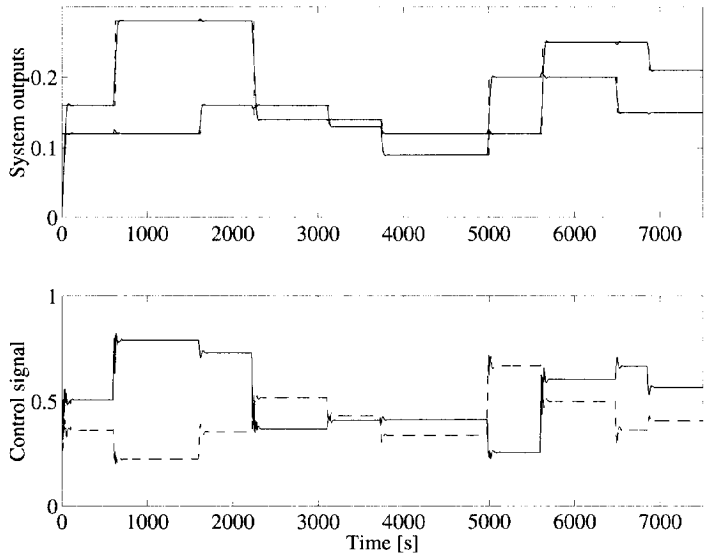


Figure 3.13. Simulated decoupled control of the cascaded-tanks setup. System outputs and control signals when there are only constraints on the first control move.

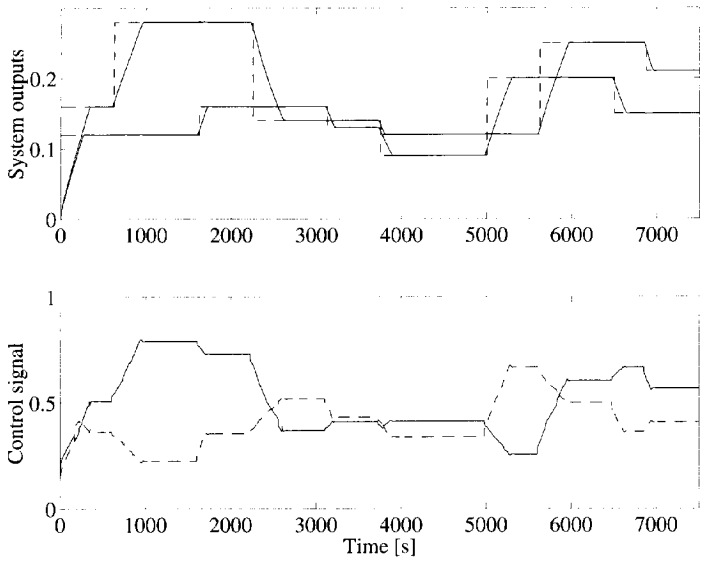


Figure 3.14. Simulated decoupled control of the cascaded-tanks setup. System outputs and control signals when there are constant constraints on the input.

(3.19)) can be hard coded or on-line inverted. The computational complexity of the numeric decoupling method, however, is much greater – the numerical decoupling law is obtained by solving a nonlinear optimization problem which is, in general, a time-consuming task with no guarantee that the optimal solution will be obtained (see Chapter 4).

3.5 Summary and concluding remarks

This chapter presented an extension of the Relative Gain Array (RGA) that facilitates the analysis of interactions in MIMO TS fuzzy models (Section 3.2.1). The specific TS structure is explored in order to obtain a small number of RGAs that indicate the interactions throughout the model. Depending on the antecedent structure, it may be possible to compute the RGAs for the separate rules. These RGAs, however, cannot provide information about the coupling when two or more rules are active. To obtain that information for a specific point in between in the rules, we first derive a linear model valid around that point.

Another tool to analyse the interactions is the output sensitivity function proposed in Section 3.2.2. It measures how (in)dependent a process output is on variations in one or more of its inputs. The function is computed as partial derivative of the output with respect to a given input while the remaining inputs are kept invariant.

The insight obtained by both the RGA and sensitivity analyses should be used in the design of the control system. While within a certain operation region with weak interactions SISO controllers may suffice, in a different region where strong

coupling is present a full or partial decoupler should be designed. In Section 3.3, three methods were introduced that achieve such decoupling, taking advantage of the particular structure of the TS model. Essentially all of them aim to invert the model and the difference between them is shown to be in the means used to achieve this inversion. If the fuzzy model is an affine one, it is possible to obtain an analytic decoupling law by applying tools from the differential geometry theory (Appendix B). For non-affine fuzzy models we can either compute the input-output gains at each sampling instant, or numerically invert the model by solving a nonlinear optimization problem.

The developed techniques were applied to a TS fuzzy model of a laboratory-scaled cascaded-tanks setup (Section 3.4). The analysis based on both RGAs and output sensitivity functions indicated strong input-output coupling that varies for the different operating regions. Since detached controllers of PI type could not provide satisfactory performance, a decoupler based on the fuzzy model was designed. It not only cancelled the coupling, but also compensated for the nonlinearities in the setup. It also rendered the resulting "decoupler & setup" system linear (input-output feedback linearization), simplifying the control problem.

Finally, it should be noted that the proposed decoupling methods have different computational complexities. The decoupling of affine TS models can be achieved computationally very efficiently. This allows for their use in situations where, due to short sampling times, some of the more advanced MIMO control techniques cannot be properly utilized. The same holds for the analytic decoupling method for the non-affine (general) TS model. In contrast, the numeric decoupling method is computationally demanding, which limits its application in fast processes. However, it is the most general method that can be applied regardless of the model structure.

4 FUZZY MODEL PREDICTIVE CONTROL

This chapter focuses on the optimization problem in fuzzy model predictive control. The success of linear model predictive control in controlling constrained linear processes is mainly due to the fact that the on-line optimization problem is convex. When the process model is nonlinear fuzzy model, non-convex, time-consuming optimization is necessary, with no guarantee of finding the optimal solution (Section 4.1). A possible way around this problem is to linearize the fuzzy model at the current operating point and use the linear predictive control (i.e., quadratic programming). For predictive control with a long prediction horizon, however, the influence of the linearization error may significantly deteriorate the performance. This is remedied by linearizing the fuzzy model along the predicted input and output trajectories (Section 4.3). One can further improve the model prediction by iteratively applying the optimized control sequence to the fuzzy model and linearizing along the so obtained simulated trajectories. Four different methods for the construction of the optimization problem are proposed, making difference between the cases when a single linear model or a set of linear models is used. By choosing an appropriate method, one can achieve a desired tradeoff between the control performance and the computational load. Section 4.4 shows how the optimization problem is formulated based on linear time-varying state-space models.

The proposed techniques are tested and evaluated using two examples: control of pH in a simulated continuous stirred tank reactor and real-time liquid level control of a laboratory-scale tanks setup (Section 4.5).

4.1 Problem formulation

In model predictive control (MPC), the control action is obtained by solving on-line, at each sampling instant, an optimization problem in order to minimize the tracking error (and possibly also the control effort). Nonlinear MPC should be applied in situations where the controlled process is inherently nonlinear, large changes in the operating conditions can be anticipated during routine operation, such as in batch processes, or during the start-up and shutdown of continuous processes. The use of nonlinear models in MPC is motivated by the need to improve the performance; these models can predict the process behaviour better. TS fuzzy models proved to be suitable for the use in nonlinear MPC because of their ability to accurately approximate complex nonlinear systems by using data combined with prior knowledge.

The main problem in the real-time application of fuzzy MPC is that the convexity of the optimization problem is lost, and hence time-consuming optimization is necessary, with no guarantee of finding an optimal solution. This hampers the application to fast processes, where iterative optimization techniques cannot be properly used for short sampling periods. Here, methods are presented that avoid non-convex optimization by employing a single state-space local linear model or a set of these that approximates the fuzzy model along the predicted trajectories. The structure of the optimization problem is explored in order to arrive at a suboptimal solution which is as close as possible to the optimum in a limited amount of time, and this approach makes nonlinear MPC also suitable for fast processes. Our approach is based on linear time-varying (LTV) prediction models derived by freezing the parameters of the fuzzy model at a given operating point[1]. (Note that other means for obtaining linear models from the fuzzy model are also possible, for example Kavsek et al. (1997) extract a step response model from the fuzzy one, while Abonyi et al. (2001) apply Jacobian linearization to the fuzzy model.) The control signal is obtained by solving a constrained quadratic programme (QP). To account for errors introduced by the linearization, an iterative optimization scheme is proposed. In such a setting, the QP solution provides a search direction toward the minimum of the optimization problem. Convergence is guaranteed through a line search mechanism that considers reduction both in the cost function and in the constraints. The method belongs to the general class of sequential quadratic programming methods (Powell, 1978; Pshenichnyj, 1994). However, a specific feature of our approach is the way we define (and update) the Hessian and gradient, using the LTV interpretation of the TS fuzzy model. The advantage is its generic form that does not depend on the structure of the fuzzy model used. It is similar to (but more general than) the one presented by Gerkšič et al. (2000), who designed a nonlinear model predictive control method for Wiener type models. To predict the process output, the authors combine the linear dynamic part with a linear approximation of the static nonlinearity derived at each sampling instant.

[1]Nonlinear models can be considered as linear-time varying (LTV) models around a given trajectory. From such a point of view, the term *linear parameter-varying* (LPV) models would be better suited. However, for consistency reasons we use the term LTV throughout the thesis.

The considered fuzzy model predictive controller computes the optimal control sequence $\mathbf{u}(k)$ with respect to the following cost function

$$\min_{\mathbf{u}} J = \sum_{i=H_{\min}}^{H_p} \left\|\mathbf{r}_y(k+i|k) - \hat{\mathbf{y}}(k+i|k)\right\|_{\mathbf{P}_i}^2 + \sum_{i=H_{\min}}^{H_p} \left\|\Delta\hat{\mathbf{y}}(k+i-1|k)\right\|_{\Delta\mathbf{P}_i}^2 + \sum_{j=1}^{H_c-1} \left\|\mathbf{r}_u(k+j|k) - \hat{\mathbf{u}}(k+j|k)\right\|_{\mathbf{Q}_j}^2 + \sum_{j=1}^{H_c-1} \left\|\Delta\hat{\mathbf{u}}(k+j|k)\right\|_{\Delta\mathbf{Q}_j}^2. \quad (4.1)$$

Here, $\hat{\mathbf{y}}(k+i|k)$ is the output of the fuzzy model (2.6) predicted at time k, $\Delta\hat{\mathbf{y}}(k+i|k)$ is the predicted output increment, $\mathbf{r}_y(k+i|k)$ and $\mathbf{r}_u(k+i|k)$ are the output and input references, available at time k, $\hat{\mathbf{u}}(k+i|k)$ and $\Delta\hat{\mathbf{u}}(k+i|k)$ are the control signal and its increment, assumed at time k, respectively, and $\|\cdot\|$ represents the inner-product norm. The cost function J penalizes: (a) the deviations of the predicted controlled outputs $\hat{\mathbf{y}}(k+i|k)$ from the output reference trajectory $\mathbf{r}_y(k+i|k)$, (b) the predicted output increment $\Delta\hat{\mathbf{y}}(k+i|k)$, (c) the deviations of the manipulated inputs $\hat{\mathbf{u}}(k+i|k)$ from the input reference trajectory $\mathbf{r}_u(k+i|k)$ (usually this is done when there are more inputs than outputs), and (d) the input increment $\Delta\mathbf{u}(k+j|k)$.

The parameters H_p, H_c and $H_{\min}$, called the *prediction*, *control* and *minimum cost* horizon, respectively, define the intervals over which the optimization is carried out. The control horizon cannot be longer that the prediction horizon, $H_c \leq H_p$, and $\hat{\mathbf{u}}(k+j|k) = \hat{\mathbf{u}}(k+H_c-1|k)$ for all $j \geq H_c$, i.e., the control signal is manipulated only within the control horizon and remains constant afterwards. The weights $\mathbf{P}_i$, $\Delta\mathbf{P}_i$, $\mathbf{Q}_j$ and $\Delta\mathbf{Q}_j$ determine the relative importance of the different terms in the cost function. Some of the above parameters may be dictated by economic objectives of the control system, but essentially they are *tuning parameters* which can be adjusted to give satisfactory performance in terms of reference tracking, disturbance rejection and robustness against model-plant mismatch (Soeterboek, 1992; Maciejowski, 2002).

The inputs and outputs are subject to (time-varying) level and rate constraints

$$\mathbf{u}_{\min}(k+j|k) \leq \mathbf{u}(k+j|k) \leq \mathbf{u}_{\max}(k+j|k), \quad (4.2a)$$

$$\Delta\mathbf{u}_{\min}(k+j|k) \leq \Delta\mathbf{u}(k+j|k) \leq \Delta\mathbf{u}_{\max}(k+j|k), \quad (4.2b)$$

$$\mathbf{y}_{\min}(k+i|k) \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}(k+i|k), \quad (4.2c)$$

$$\Delta\mathbf{y}_{\min}(k+i|k) \leq \Delta\mathbf{y}(k+i|k) \leq \Delta\mathbf{y}_{\max}(k+i|k). \quad (4.2d)$$

Using the fuzzy model (2.6) and propagating the model output backward, one can transform the output constraints into constraints on the control signal; hence (4.2) can be written as

$$G_j(\mathbf{u}) \leq 0, \qquad j = 1, \ldots, c, \quad (4.3)$$

where c is the total number of constraints[2] (Boggs and Tolle, 1995; Boggs et al., 1999).

Given the optimization problem (4.1) – (4.3) for which we seek an optimal control sequence $\mathbf{u}_{\text{opt}}$, the idea is to "adjoin " the constraints (4.3) to the performance index

[2]Here it is assumed that the constraints are linearly independent, as the *rank* of the set of constraints, rather than the number of constraints is the important value.

(4.1) by a set of c "undetermined multipliers," $\lambda_1, \ldots, \lambda_c$, as follows (Bryson and Ho, 1981)

$$\begin{aligned} L(\mathbf{u}, \lambda) &= J(\mathbf{u}) + \sum_{j=1}^{c} \lambda_j G_j(\mathbf{u}) \\ &= J(\mathbf{u}) + \boldsymbol{\lambda}\mathbf{G}(\mathbf{u}). \end{aligned} \tag{4.4}$$

The function $L(\mathbf{u}, \lambda)$ and the constants $\lambda_1, \ldots, \lambda_c$ are often referred to as the Lagrange function and Lagrange multipliers, respectively.

Then a QP sub-problem is formulated using a second-order approximation of the Lagrange function (4.4). If the optimization problem is a convex programming problem, that is, both the cost function $J(\mathbf{u})$ and the constraints $G_j(\mathbf{u})$, $j = 1, \ldots, c$ are convex functions of $\mathbf{u}$, then the necessary and sufficient condition for a global minimum is the existence of a solution of the Kuhn-Tucker (KT) equation (Kuhn and Tucker, 1951)

$$\nabla J(\mathbf{u}_{\text{opt}}) + \sum_{j=1}^{c} \lambda_j^{\text{opt}} \cdot \nabla G_j(\mathbf{u}_{\text{opt}}) = 0. \tag{4.5}$$

The underlying idea is that if $\mathbf{u}_{\text{opt}}$ is to be an optimal point subject to $G_j(\mathbf{u}_{\text{opt}}) = 0$, $j = 1, \ldots, c$, i.e., when all constraints are active, then $\nabla J(\mathbf{u}_{\text{opt}})$ must lie between the negative gradients of $G_j(\mathbf{u}_{\text{opt}})$. Analytically this means that $\nabla J(\mathbf{u}_{\text{opt}})$ can be expressed as a *negative* linear combination of $\nabla G_j(\mathbf{u}_{\text{opt}})$. In other words, "the gradient $\nabla J(\mathbf{u}_{\text{opt}})$ must be pointed in such a way that decrease of J can only come by violating the constraints" (Bryson and Ho, 1981). Therefore, for the gradients to be canceled, the Lagrange multipliers ($\lambda_j \geq 0$, $j = 1, \ldots, c$), are necessary to balance the deviations in magnitude of the cost function and of the constraint gradients. Since only active constraints are included in this canceling operation, the Lagrange multipliers corresponding to non-active constraints ($G_j(\mathbf{u}_{\text{opt}}) < 0$) are set to zero.

If we use the nonlinear TS model, however, the cost function is non-convex with respect to the control signal. For the resulting non-convex optimization problem, the existence of a solution of the KT equation is then only necessary but sufficient condition to guarantee that a global minimum is attained. With the method we propose, the optimization problem is approached in a different manner. The method has the same structure as the Sequential Quadratic Programme (SQP) (given in Appendix C), which is used to solve optimization problems for which the cost function and/or the constraints are nonlinear functions of the control signal $\mathbf{u}$ in the following way:

1. Update the gradient and the Hessian of the Lagrange function (4.4).
2. Solve a QP subproblem based on the updated gradient and Hessian.
3. Perform a line search to ensure convergence in an iterative scheme.

The major difference lies in the procedure used to update the gradient and the Hessian of the Lagrange function (Step 1 above). In most SQP implementations, the gradient and the Hessian are updated at each iteration through a quasi-Newton approximation (Appendix C.1). However, such an approximation introduces errors that can slow

down the convergence of the solution and even lead to a local rather than to the global minimum. Therefore instead of using a quasi-Newton approximation to update the gradient and the Hessian, we first linearize the original fuzzy model $\mathcal{R}$ at the current operating point or along a predicted trajectory (Appendix D). Having transformed $\mathcal{R}$ to a local linear model (or a set of such models), we can apply the results from linear MPC (García et al., 1989) to get a direct – *not* an approximated – expression for the gradient and the Hessian (Mollov, Babuška, Abonyi and Verbruggen, 2002). The convergence of the optimized control sequences to the optimal one is guaranteed through a line search routine that considers reduction both in the cost function and the constraints (Appendix C.3).

4.2 Internal model control scheme

To compensate for the disturbances acting on the process, measurement noise and model-plant mismatch, the internal model control (IMC) scheme is applied (Economou et al., 1986). The purpose of the fuzzy model working in parallel with the process is to subtract the effect of the control action from the process output (Fig. 4.1). If the predicted and the measured process outputs are equal, the feedback signal is zero and the controller works in an open-loop configuration. If a disturbance acts on the process output, the feedback signal is equal to the influence of the disturbance and is not affected by the effects of the control action. This signal is filtered and subtracted from the reference. With a perfect process model, the IMC scheme is hence able to cancel the effect of unmeasurable output-additive disturbances. Two basic properties of the ideal IMC are inherent stability and perfect control. Inherent stability means that if the controller and the process are input-output stable and a perfect model of the process is available, the closed-loop system is input-output stable. If the system is not input-output stable, but it can be stabilized by feedback, IMC can still be applied. Perfect control means that if the controller is an exact inverse of the model, and the closed-loop system is stable, then the control is error-free, i.e., $y(k) = r(k)$, $\forall k$. However, in practice, the model is never an exact representation of the process. The feedback signal then contains both the effect of unmeasurable disturbances and of the modeling errors, and it becomes a true feedback. For large modeling errors, it deteriorates the performance of the control system and may introduce stability problems. The feedback filter is introduced in order to filter out the measurement noise and to reduce the loop gain (Morari and Zafiriou, 1989). The IMC scheme thus provides the nominal stability, i.e., stability for the case when the model-plant mismatch can be neglected.

4.3 Schemes for obtaining linear models

The LTV models for the QP sub-problem are obtained as follows. The fuzzy model is used to predict the future process behaviour. Then the linear models are derived as the fuzzy model is linearized along the predicted input and output trajectories. Figure 4.1 presents the flow diagram of the fuzzy model predictive controller.

Let $\mathcal{U} = [\mathbf{u}(k+1), \ldots, \mathbf{u}(k+H_\mathrm{p})] \in \mathbb{R}^{m \times H_\mathrm{p}}$ and $\mathcal{Y} = [\mathbf{y}(k+1), \ldots, \mathbf{y}(k+H_\mathrm{p})] \in \mathbb{R}^{p \times H_\mathrm{p}}$ be some general input and output model trajectories, $\mathcal{U}_\mathrm{opt} \in \mathbb{R}^{m \times H_p}$ and $\mathcal{Y}_\mathrm{opt} \in \mathbb{R}^{p \times H_p}$ be the (unknown) optimal trajectories obtained by applying $\mathcal{U}_\mathrm{opt}$ to the nonlin-

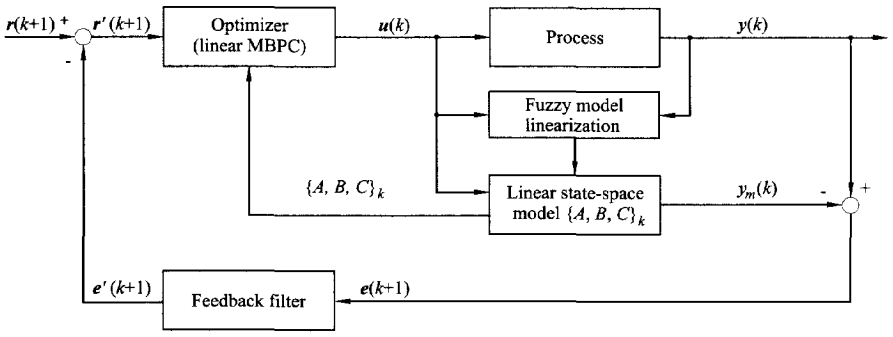


Figure 4.1. Fuzzy model predictive control.

ear fuzzy model, and $\mathcal{U}^* \in \mathbb{R}^{m \times H_p}$ and $\mathcal{Y}^* \in \mathbb{R}^{p \times H_p}$ the trajectories computed by the linear predictive controller (Quadratic Programme (4.22)–(4.25)).

Denote the LTV model extracted from the fuzzy model at the $(k+i)$th step by

$$M(k+i) = \{\mathbf{A}(k+i), \mathbf{B}(k+i), \mathbf{C}(k+i)\} \quad i = 0, \ldots, H_p, \tag{4.6}$$

the LTV system comprising the models $M(k+i)$ by

$$\mathcal{M} = \{M(k), M(k+1), \ldots, M(k+H_p)\}.$$

If $\mathcal{U}^* = \mathcal{U}_{opt}$ and $\mathcal{Y}^* = \mathcal{Y}_{opt}$, then the solution of (4.1) – (4.3) is identical to the solution of linear MPC, based on the LTV models. However, $\mathcal{U}_{opt}$ and $\mathcal{Y}_{opt}$ are not available, and the problem is how to generate $\mathcal{U}^*$, "close" to the optimal solution $\mathcal{U}_{opt}$ that would lead to $\mathcal{Y}_{opt}$, without solving (4.1) – (4.3).

$\mathcal{M}$ is generated by linearizing $\mathcal{R}$ along the trajectory $\mathcal{U}$ and $\mathcal{Y}$, therefore the problem is the estimation of $\mathcal{U}$ and $\mathcal{Y}$. Several methods can be distinguished depending on the way $\mathcal{M}$ is obtained. Generally they can be classified into two groups: non-iterative and iterative ones (Mollov, van der Veen, Babuška, Abonyi, Roubos and Verbruggen, 1999; Mollov, Babuška, Abonyi and Verbruggen, 2002).

4.3.1 Non-iterative methods

In the non-iterative methods, a single local model of a set of local models, obtained at the current time instant k is used in the quadratic programme and the obtained QP solution is directly applied to the controlled process.

Single model. This method is the basic one. The model set $\mathcal{M}$ comprises a single linear model $M(k)$, extracted at $\{\mathbf{u}(k-1), \mathbf{y}(k)\}$. This model is used throughout the entire prediction horizon. Even if this model is very accurate at the linearization point, its accuracy decreases over the prediction horizon. As a consequence, there may be a significant prediction error at $k + H_p$. To reduce the error, a set of local linear models can be extracted from the fuzzy one along the trajectories found at the previous time instant $k-1$. This is the basis of the multiple model method.

Multiple model based on trajectory computed at the previous sampling instant. Let $\mathcal{U}^*(k-1)$ and $\mathcal{Y}^*(k-1)$ be the input and output sequences obtained by optimization at time step $k-1$ and by subsequent simulation of the nonlinear fuzzy model:

$$\mathcal{U}^*(k-1) = \left[\mathbf{u}^*(k-1), \mathbf{u}^*(k), \ldots, \mathbf{u}^*(k+H_\mathrm{p}-1)\right]$$
$$\mathcal{Y}^*(k-1) = \left[\mathbf{y}^*(k-1), \mathbf{y}^*(k), \ldots, \mathbf{y}^*(k+H_\mathrm{p}-1)\right].$$

We use this information to approximate the yet unknown input and output sequences at time k by shifting the elements one step forward and replacing the last element:

$$\mathcal{U}(k) = \left[\mathbf{u}^*(k), \mathbf{u}^*(k+1), \ldots, \mathbf{u}^*(k+H_\mathrm{p}-1), \mathbf{u}^*(k+H_\mathrm{p}-1)\right]$$
$$\mathcal{Y}(k) = \left[\mathbf{y}^*(k), \mathbf{y}^*(k+1), \ldots, \mathbf{y}^*(k+H_\mathrm{p}-1), \mathbf{y}^*(k+H_\mathrm{p}-1)\right].$$

The local linear models $\mathcal{M} = \left\{M(k), M(k+1)\ldots, M(k+H_p)\right\}$ extracted from the fuzzy model along $\left\{\mathcal{U}(k),\ \mathcal{Y}(k)\right\}$ are then used to construct the quadratic programme and to compute $\mathcal{U}^*(k)$. In this way, the prediction error is reduced.

4.3.2 Iterative methods

With the non-iterative methods, the performance still may be suboptimal due to the fact that the control sequence, along which the fuzzy model is linearized, is from the previous time step. To improve the performance, iterative methods can be introduced, in which the optimized control sequence is not directly applied to the process, but is first used to simulate the fuzzy model. The resulting model outputs (and control inputs) then provide more accurate trajectories along which a new set of local models is obtained and the whole procedure is iteratively repeated.

Iterative version of the *Single model* method. In the first iteration, a linear model $M(k)$ is obtained at $\{\mathbf{u}^*(k-1), \mathbf{y}(k)\}$. This model is employed to compute $\mathcal{U}^*(k)$. Thereafter, in the following iterations $\{\mathbf{u}^*(k), \mathbf{y}(k)\}$ is used as a linearization point.

To further reduce the error, a set of local linear model can be extracted from the fuzzy model along the trajectories in an iterative manner. This is the basis of the following iterative multiple model method.

Multiple model method. Rather than using a single model over the whole prediction horizon, one can use a separate model for each step within the prediction horizon (see Section 4.4), (Mollov et al., 1998a; Mollov et al., 1999b). The algorithm is summarized on the following page.

Although, according to the receding-horizon principle, only the first control action is used to control the process, the complete sequence is available. This can be used to initialize the iterative routine at the next sample instant, starting from a point close to the optimum.

Table 4.1 on page 67 summarizes how the linear models are obtained from the fuzzy one in the different methods. In the iterative methods, at each iteration the fuzzy model is linearized around the corresponding sequences $\mathcal{Y}^*(k)$ and $\mathcal{U}^*(k)$.

Algorithm 4.1 (Multiple model algorithm)

Use the already obtained linear model $M(k)$ and compute the control sequence $\mathcal{U}_1^*(k)$ for the whole prediction horizon. Choose the termination tolerance ϵ.

Repeat for $i = 2,3,\ldots$

Step 1: **Simulate** the fuzzy model over the prediction horizon.

Step 2: **Linearize** the fuzzy model along the predicted trajectory $\{\mathcal{U}_{i-1}^*(k),\ \mathcal{Y}_{i-1}^*(k)\}$ and obtain $\mathcal{M}$.

Step 3: **Use $\mathcal{M}$ to compute the new control sequence** $\mathcal{U}_i^*(k)$ for the whole prediction horizon.

Until: $\|\mathcal{U}_i^*(k) - \mathcal{U}_{i-1}^*(k)\| < \epsilon$ or the maximum number of iterations is reached.

4.4 Optimization problem based on a linear time-varying model

Let the process be locally represented by a linear time-varying (LTV) model $M(k) = \{\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)\}$,

$$\begin{aligned} \mathbf{x}_{\text{lin}}(k+1) &= \mathbf{A}(k)\mathbf{x}_{\text{lin}}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}_{\text{lin}}(k) &= \mathbf{C}(k)\mathbf{x}_{\text{lin}}(k). \end{aligned} \tag{4.7}$$

To ensure an offset-free reference tracking, we define the optimization problem with respect to the increment in the control signal $\Delta\mathbf{u}$, rather than with respect to the control signal $\mathbf{u}$. The state-space description is extended correspondingly

$$\begin{bmatrix} \mathbf{x}_{\text{lin}}(k+1) \\ \mathbf{u}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(k) & \mathbf{B}(k) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{lin}}(k) \\ \mathbf{u}(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{B}(k) \\ \mathbf{I} \end{bmatrix} \Delta\mathbf{u}(k) \tag{4.8}$$

$$\mathbf{y}_{\text{lin}}(k) = \begin{bmatrix} \mathbf{C}(k) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{lin}}(k) \\ \mathbf{u}(k-1) \end{bmatrix}$$

$$\Updownarrow$$

$$\begin{aligned} \bar{\mathbf{x}}_{\text{lin}}(k+1) &= \bar{\mathbf{A}}(k)\bar{\mathbf{x}}_{\text{lin}}(k) + \bar{\mathbf{B}}(k)\Delta\mathbf{u}(k) \\ \mathbf{y}_{\text{lin}}(k) &= \bar{\mathbf{C}}(k)\bar{\mathbf{x}}_{\text{lin}}(k). \end{aligned} \tag{4.9}$$

For the sake of simplicity, in the sequel we drop the bars from the modified state-space form. Assuming that at time instant k the state vector and the future control sequence are known, the future process outputs can be predicted through a successive

Table 4.1. Schemes for obtaining local linear models. In the iterative methods these local linear models are obtained as the fuzzy model is linearized in each iteration along the corresponding sequences $\mathcal{Y}^*_{\text{Iter}}(k)$ and $\mathcal{U}^*_{\text{Iter}}(k)$.

Method, *Iteration*	Model used at time instant				
	k	$k+1$	$k+2$	...	$k+H_\text{p}$
Single model	$M(k)$	$M(k)$	$M(k)$	...	$M(k)$
Multiple model based on trajectory computed at the previous instant	$M(k)$	$M(k+1)$	$M(k+2)$	...	$M(k+H_\text{p})$
Iterative version of the *Single model*					
Iteration 1	$M_1(k)$	$M_1(k)$	$M_1(k)$	...	$M_1(k)$
Iteration 2	$M_2(k)$	$M_2(k)$	$M_2(k)$	...	$M_2(k)$
⋮	⋮	⋮	⋮	⋱	⋮
Iteration N	$M_N(k)$	$M_N(k)$	$M_N(k)$	...	$M_N(k)$
Multiple model					
Iteration 1	$M_1(k)$	$M_1(k+1)$	$M_1(k+2)$	...	$M_1(k+H_\text{p})$
Iteration 2	$M_2(k)$	$M_2(k+1)$	$M_2(k+2)$	...	$M_2(k+H_\text{p})$
⋮	⋮	⋮	⋮	⋱	⋮
Iteration N	$M_N(k)$	$M_N(k+1)$	$M_N(k+2)$	...	$M_N(k+H_\text{p})$

substitution:

$$\begin{aligned}\mathbf{x}_{\text{lin}}(k+2) &= \mathbf{A}(k+1)\mathbf{x}_{\text{lin}}(k+1)+\mathbf{B}(k+1)\Delta\mathbf{u}(k+1)\\ &= \mathbf{A}(k+1)\left[\mathbf{A}(k)\mathbf{x}_{\text{lin}}(k)+\mathbf{B}(k)\Delta\mathbf{u}(k)\right]+\mathbf{B}(k+1)\Delta\mathbf{u}(k+1)\\ &= \mathbf{A}(k+1)\mathbf{A}(k)\mathbf{x}_{\text{lin}}(k)+\mathbf{A}(k+1)\mathbf{B}(k)\Delta\mathbf{u}(k)+\mathbf{B}(k+1)\Delta\mathbf{u}(k+1)\\ &\ldots\end{aligned}$$

$$\mathbf{x}_{\text{lin}}(k+H_\text{p}) = \prod_{i=H_\text{p}-1}^{0}\mathbf{A}(k+i)\mathbf{x}_{\text{lin}}(k)+\sum_{i=0}^{H_\text{p}-1}\prod_{\substack{j=H_\text{p}-1,\\ j\geq i+1}}^{i+1}\mathbf{A}(k+j)\mathbf{B}(k+i)\Delta\mathbf{u}(k+i),$$

where $\prod_{i=H_\text{p}-1}^{0}\mathbf{A}(k+i)=\mathbf{A}(k+H_p-1)\mathbf{A}(k+H_\text{p}-2)\cdots\mathbf{A}(k+1)\mathbf{A}(k)$. The predicted output follows directly

$$\hat{\mathbf{y}}_{\text{lin}}(k+H_{\text{p}}) = \mathbf{C}(k+H_{\text{p}}) \prod_{i=H_{\text{p}}-1}^{0} \mathbf{A}(k+i)\mathbf{x}_{\text{lin}}(k) + \mathbf{C}(k+H_{\text{p}}) \sum_{i=0}^{H_{\text{p}}-1} \prod_{\substack{j=H_{\text{p}}-1,\\ j>i+1}}^{i+1} \mathbf{A}(k+j)\mathbf{B}(k+i)\Delta\mathbf{u}(k+i). \tag{4.10}$$

The complete output sequence over the prediction horizon for $H_{\min} = 1$ is given as

$$\begin{bmatrix} \hat{\mathbf{y}}(k+1) \\ \hat{\mathbf{y}}(k+2) \\ \vdots \\ \hat{\mathbf{y}}(k+H_{\text{p}}) \end{bmatrix} = \mathbf{R}_{\text{x}}\mathbf{A}(k)\mathbf{x}_{\text{lin}}(k) + \mathbf{R}_{\text{u}} \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+H_{\text{c}}-1) \end{bmatrix}, \tag{4.11}$$

where

$$\mathbf{R}_{\text{x}} = \begin{bmatrix} \mathbf{C}(k) \\ \mathbf{C}(k+1)\mathbf{A}(k) \\ \vdots \\ \mathbf{C}(k+H_{\text{p}}) \prod_{i=H_{\text{p}}-1}^{0} \mathbf{A}(k+i) \end{bmatrix}, \tag{4.12}$$

$$\mathbf{R}_{\text{u}} = \begin{bmatrix} \mathbf{C}(k+1)\mathbf{B}(k) & \dots & \mathbf{0} \\ \mathbf{C}(k+2)\mathbf{A}(k+1)\mathbf{B}(k) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}(k+H_{\text{p}}) \prod_{i=H_{\text{p}}-1}^{1} \mathbf{A}(k+i)\mathbf{B}(k) \dots \mathbf{C}(k+H_{\text{p}}) \prod_{i=H_{\text{p}}-1}^{H_{\text{c}}} \mathbf{A}(k+i)\mathbf{B}(k+H_{\text{c}}-1) \end{bmatrix}. \tag{4.13}$$

The input sequence for the input level constraints (4.2a) can be defined as

$$\begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+H_{\text{c}}-1) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I}_{\text{m}} \\ \mathbf{I}_{\text{m}} \\ \vdots \\ \mathbf{I}_{\text{m}} \end{bmatrix}}_{\mathbf{I}_{\text{u}}} \mathbf{u}(k-1) + \underbrace{\begin{bmatrix} \mathbf{I}_{\text{m}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}_{\text{m}} & \mathbf{I}_{\text{m}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{\text{m}} & \mathbf{I}_{\text{m}} & \dots & \mathbf{I}_{\text{m}} \end{bmatrix}}_{\mathbf{I}_{\Delta\text{u}}} \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+H_{\text{c}}-1) \end{bmatrix}, \tag{4.14}$$

where $\mathbf{I}_{\text{m}}$ is an $m \times m$ identity matrix. Inserting (4.14) in (4.2a) gives

$$\begin{bmatrix} -\mathbf{I}_{\Delta\text{u}} \\ \mathbf{I}_{\Delta\text{u}} \end{bmatrix} \Delta\mathbf{u} \leq \begin{bmatrix} \mathbf{I}_{\text{u}} \left(-\mathbf{u}_{\min}(k) + \mathbf{u}(k-1) \right) \\ \mathbf{I}_{\text{u}} \left(\mathbf{u}_{\max}(k) - \mathbf{u}(k-1) \right) \end{bmatrix}. \tag{4.15}$$

The input rate constraints (4.2b) are respectively

$$\begin{bmatrix} -\mathbf{I}_{H_{\text{p}}\text{m}} \\ \mathbf{I}_{H_{\text{p}}\text{m}} \end{bmatrix} \Delta\mathbf{u} \leq \begin{bmatrix} -\mathbf{I}_{H_{\text{p}}\text{m}} \Delta\mathbf{u}_{\min}(k) \\ \mathbf{I}_{H_{\text{p}}\text{m}} \Delta\mathbf{u}_{\min}(k) \end{bmatrix}, \tag{4.16}$$

where $\mathbf{I}_{H_{\text{p}}\text{m}}$ is an $(H_{\text{p}} \cdot m \times H_{\text{p}} \cdot m)$ identity matrix.

The output level constraints (4.2c) can be derived through the prediction equation (4.11)

$$\begin{bmatrix} -\mathbf{R}_u \\ \mathbf{R}_u \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} -\mathbf{y}_{\min}(k) + \mathbf{R}_x \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) \\ \mathbf{y}_{\max}(k) - \mathbf{R}_x \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) \end{bmatrix}. \tag{4.17}$$

The output rate constraints (4.2d) are also derived through (4.11), although the expression is more involved. Since $\mathbf{y}(k)$ is not predicted but measured, for the prediction of $\Delta\hat{\mathbf{y}}(k+1)$ we have

$$\begin{bmatrix} -\mathbf{dR}_{u1} \\ \mathbf{dR}_{u1} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} -\Delta \mathbf{y}_{\min}(k) + \mathbf{dR}_{x1} \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) - \mathbf{y}_{\text{lin}}(k) \\ \Delta \mathbf{y}_{\max}(k) - \mathbf{dR}_{x1} \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) + \mathbf{y}_{\text{lin}}(k) \end{bmatrix}, \tag{4.18}$$

where the matrices $\mathbf{dR}_{x1}$ and $\mathbf{dR}_{u1}$ are defined as $\mathbf{dR}_{x1} = \mathbf{C}(k)$ and $\mathbf{dR}_{u1} = \mathbf{C}(k)\mathbf{B}(k) - \mathbf{C}(k)$. The constraints on the predicted outputs $\Delta\hat{\mathbf{y}}(k+2), \ldots, \Delta\hat{\mathbf{y}}(k+H_p)$ are

$$\begin{bmatrix} -\mathbf{dR}_u \\ \mathbf{dR}_u \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} -\mathbf{y}_{\min}(k) + \mathbf{dR}_x \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) \\ \mathbf{y}_{\max}(k) - \mathbf{dR}_x \mathbf{A}(k) \mathbf{x}_{\text{lin}}(k) \end{bmatrix}, \tag{4.19}$$

where

$$\mathbf{dR}_x = \begin{bmatrix} \mathbf{C}(k+1)\mathbf{A}(k) - \mathbf{C}(k) \\ \mathbf{C}(k+2)\mathbf{A}(k+1)\mathbf{A}(k) - \mathbf{C}(k+1)\mathbf{A}(k) \\ \vdots \\ \mathbf{C}(k+H_p)\prod_{i=H_p-1}^{0} \mathbf{A}(k+i) - \mathbf{C}(k+H_p-1)\prod_{i=H_p-2}^{0} \mathbf{A}(k+i) \end{bmatrix}, \tag{4.20}$$

$$\mathbf{dR}_u = \begin{bmatrix} \mathbf{C}(k+2)\mathbf{A}(k+1)\mathbf{B}(k) - \mathbf{C}(k+1)\mathbf{B}(k) & \cdots \\ \mathbf{C}(k+3)\mathbf{A}(k+2)\mathbf{A}(k+1)\mathbf{B}(k) - \mathbf{C}(k+2)\mathbf{A}(k+1)\mathbf{B}(k) & \cdots \\ \vdots & \ddots \\ \mathbf{C}(k+H_p)\prod_{i=H_p-1}^{1} \mathbf{A}(k+i)\mathbf{B}(k) - \mathbf{C}(k+H_p-1)\prod_{i=H_p-2}^{1} \mathbf{A}(k+i)\mathbf{B}(k) & \cdots \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{C}(k+H_p)\prod_{i=H_p-1}^{H_c+1} \mathbf{A}(k+i)\mathbf{B}(k+H_c) - \mathbf{C}(k+H_p-1)\prod_{i=H_p-2}^{H_c+1} \mathbf{A}(k+i)\mathbf{B}(k+H_c) \end{bmatrix} \tag{4.21}$$

Given the LTV model (4.9), the cost function (4.1) and the constraints (4.2), the optimization problem for the constrained linear MPC can be cast as a quadratic programme (QP)

$$\min_{\Delta \mathbf{u}} \left\{ \frac{1}{2} \Delta \mathbf{u}^T \cdot \mathbf{Hs} \cdot \Delta \mathbf{u} + \mathbf{f}^T \cdot \Delta \mathbf{u} \right\}, \tag{4.22}$$

where **Hs** and **f** are the Hessian and the gradient of the Lagrangian (4.4). The advantage of the optimization problem stated in this way is that now the **Hs** and **f** can be

straightforwardly derived from (4.1) using (4.12)–(4.14)

$$\mathbf{Hs} = 2\left\{\mathbf{R}_u^T \cdot \bar{\mathbf{P}} \cdot \mathbf{R}_u + \left(\mathbf{R}_u - \mathbf{R}_{u2}\right)^T \cdot \Delta\bar{\mathbf{P}} \cdot \left(\mathbf{R}_u - \mathbf{R}_{u2}\right) + \mathbf{I}_{\Delta u}^T \cdot \bar{\mathbf{Q}} \cdot \mathbf{I}_{\Delta u} + \Delta\bar{\mathbf{Q}}\right\} \quad (4.23)$$

$$\mathbf{f} = 2\left\{\left(\mathbf{R}_x \cdot \mathbf{A}(k) \cdot \mathbf{x}_{lin}(k) - \mathbf{r}_y\right)^T \cdot \bar{\mathbf{P}} \cdot \mathbf{R}_u + \left(\mathbf{I}_u \cdot \mathbf{u}(k) - \mathbf{r}_u\right)^T \cdot \bar{\mathbf{Q}} \cdot \mathbf{I}_{\Delta u} + \left(\mathbf{R}_x \cdot \mathbf{A}(k) \cdot \mathbf{x}_{lin}(k) - \mathbf{R}_x \cdot \mathbf{x}_{lin}(k)\right)^T \cdot \Delta\bar{\mathbf{P}} \cdot \left(\mathbf{R}_u - \mathbf{R}_{u2}\right)\right\}, \quad (4.24)$$

where

$$\mathbf{R}_{u2} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}(k+1)\mathbf{B}(k) & \cdots & \mathbf{0} \\ \mathbf{C}(k+21)\mathbf{A}(k+1)\mathbf{B}(k) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}(k+H_p-1)\mathbf{A}(k+H_p-2)\cdots\mathbf{A}(k+1)\mathbf{B}(k)\ldots & & \mathbf{C}(k+H_p-1)\mathbf{A}(k+H_p-1)\cdots\mathbf{A}(k+H_c)\mathbf{B}(k+H_c-1) \end{bmatrix}$$

and $\bar{\mathbf{P}}$, $\Delta\bar{\mathbf{P}}$, $\bar{\mathbf{Q}}$ and $\Delta\bar{\mathbf{Q}}$ in (4.24) are block-diagonal matrices of suitable dimensions, where the diagonal blocks are respectively the weights $\mathbf{P}_i$, $\Delta\mathbf{P}_i$, $\mathbf{Q}_j$ and $\Delta\mathbf{Q}_j$ in the cost function (4.1).

Using (4.15) through (4.18), we combine the constraints as follows

$$\begin{bmatrix} -\mathbf{I}_{\Delta u} \\ \mathbf{I}_{\Delta u} \\ -\mathbf{I}_{H_p m} \\ \mathbf{I}_{H_p m} \\ -\mathbf{R}_u \\ \mathbf{R}_u \\ -\mathbf{dR}_{u1} \\ \mathbf{dR}_{u1} \\ -\mathbf{dR}_u \\ \mathbf{dR}_u \end{bmatrix} \Delta\mathbf{u} \leq \begin{bmatrix} -\mathbf{I}_u\left(\mathbf{u}_{min} + \mathbf{u}(k-1)\right) \\ \mathbf{I}_u\left(\mathbf{u}_{max} - \mathbf{u}(k-1)\right) \\ -\mathbf{I}_{H_p m}\Delta\mathbf{u}_{min} \\ \mathbf{I}_{H_p m}\Delta\mathbf{u}_{max} \\ -\mathbf{y}_{min} + \mathbf{R}_x\mathbf{A}(k)\mathbf{x}_{lin}(k) \\ \mathbf{y}_{max} - \mathbf{R}_x\mathbf{A}(k)\mathbf{x}_{lin}(k) \\ -\Delta\mathbf{y}_{min1} + \mathbf{dR}_{x1}\mathbf{A}(k)\mathbf{x}_{lin}(k) - \mathbf{y}_{lin}(k) \\ \Delta\mathbf{y}_{max1} - \mathbf{dR}_{x1}\mathbf{A}(k)\mathbf{x}_{lin}(k) + \mathbf{y}_{lin}(k) \\ -\Delta\mathbf{y}_{min} + \mathbf{dR}_x\mathbf{A}(k)\mathbf{x}_{lin}(k) \\ \Delta\mathbf{y}_{max} - \mathbf{dR}_x\mathbf{A}(k)\mathbf{x}_{lin}(k) \end{bmatrix}. \quad (4.25)$$

4.5 Examples

Two examples are given to illustrate and evaluate the methods: control of pH in a simulated SISO continuous stirred tank reactor and real-time liquid level control in a MIMO laboratory-scale cascaded-tanks setup.

4.5.1 pH control in a simulated continuous stirred tank reactor

The control of the pH (the concentration of hydrogen ions) in a continuous stirred tank reactor (CSTR) is a well-known problem that presents difficulties due to the nonlin-

earity of the process dynamics (McAvoy et al., 1972). The CSTR, given in Fig. 4.2 has two input streams: one containing sodium hydroxide and the other acetic acid. The acid neutralizes sodium hydroxide of concentration C_2, which flows into the reactor at a rate F_2. The volume of the reactor is constant and equal to V. The variable of interest is the concentration of hydrogen ions pH ($pH = -\log_{10}[H^+]$) of the outlet stream, which can be expressed by

$$[\dot{H}^+] = \frac{1}{V} \cdot \frac{-[H^+]^2\{F_2C_2-(F_1+F_2)\nu\}-[H+]K_a\{F_2C_2-F_1C_1-(F_1+F_2)(\nu-\rho)\}}{3[H^+]^2+2[H^+]\{K_a+\nu\}+\{K_a(\nu-\rho)-K_w\}} \tag{4.26}$$

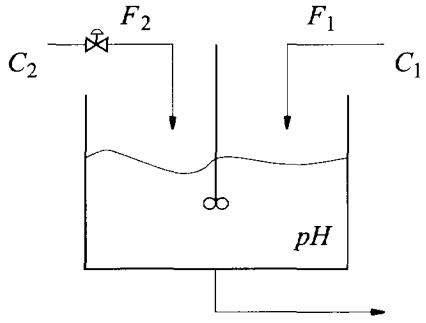


Figure 4.2. Scheme of a continuous stirred tank reactor.

The parameters used in the simulation are given in Tab. 4.2. For simplicity the acetic acid stream F_1 is considered to be constant at its nominal value. Thus we end up with a single-input, single-output process with the sodium hydroxide stream F_2 as input and the pH as output.

Table 4.2. CSTR: Parameters used in the simulation study.

Parameter	Description	Nominal Value
V	Volume of the tank	1000 [l]
F_1	Flow rate of acetic acid	81 [l/min]
F_2	Flow rate of NaOH	515 [l/min]
C_1	Inlet concentration of acetic acid	0.32 [mol/l]
C_2	Inlet concentration of NaOH	0.05 [mol/l]
$\nu = [Na^+]$	Initial concentration of sodium in the CSTR	0.0432 [mol/l]
$\rho = [HAC + AC^-]$	Initial concentration of acetate in the CSTR	0.0432 [mol/l]
K_a	Acid equilibrium constant	1.75310^{-5}
K_w	Water equilibrium constant	10^{-14}

Fuzzy modeling. A TS fuzzy model is obtained from simulated input-output data through fuzzy identification. A sampling time of $T_s = 12\,\mathrm{s}$ is used. This sampling time was chosen since the settling time between two set points for the greater part of the operating range is approximately two minutes. A first-order fuzzy model with one sample time delay in the input appeared to give satisfactory results. Three fuzzy **If - then** rules are used

$$\begin{aligned}
&\mathcal{R}_1: \textbf{If } pH(k) \text{ is LOW } \textbf{and } F_2(k) \text{ is LOW} \\
&\quad \textbf{then } pH_1(k+1) = 0.868pH(k) + 0.046F_2(k) - 22.9, \\
&\mathcal{R}_2: \textbf{If } pH(k) \text{ is MEDIUM } \textbf{and } F_2(k) \text{ is MEDIUM} \\
&\quad \textbf{then } pH_2(k+1) = 0.909pH(k) + 0.187F_2(k) - 96.0, \\
&\mathcal{R}_3: \textbf{If } pH(k) \text{ is HIGH } \textbf{and } F_2(k) \text{ is HIGH} \\
&\quad \textbf{then } pH_3(k+1) = 0.817pH(k) + 0.122F_2(k) - 61.4.
\end{aligned} \tag{4.27}$$

The cluster centers and the membership functions are given in Tab. 4.3 and Fig. 4.3, respectively.

Table 4.3. CSTR: Cluster centers.

rule	$pH(k)$	$F_2(k)$
1	7.42	517
2	8.28	518
3	9.66	519

The model output is computed as weighted average of the consequents in the individual rules

$$pH(k+1) = \frac{\sum_{i=1}^{3} \beta_i(pH, F_2) \cdot pH_i(k+1)}{\sum_{i=1}^{3} \beta_i(pH, F_2)},$$

where

$$\begin{aligned}
\beta_1(pH, F_2) &= \mu_{\text{Low}}(pH) \cdot \mu_{\text{Low}}(F_2), \\
\beta_2(pH, F_2) &= \mu_{\text{Medium}}(pH) \cdot \mu_{\text{Medium}}(F_2), \\
\beta_3(pH, F_2) &= \mu_{\text{High}}(pH) \cdot \mu_{\text{High}}(F_2).
\end{aligned} \tag{4.28}$$

The model performance on a validation data set is shown in Fig. 4.4. To give a quantitative measure of the model accuracy, we use two performance indices: the Root Mean Square (RMS) error and the Variance Accounted For (VAF), in %. For the validation data, the obtained measures are RMS $= 0.1024$ and VAF $= 98.02$. For a comparison, a linear model has correspondingly RMS $= 0.4225$ and VAF $= 63.41$ on the same validation data set, so the fuzzy model is significantly more accurate than the linear one.

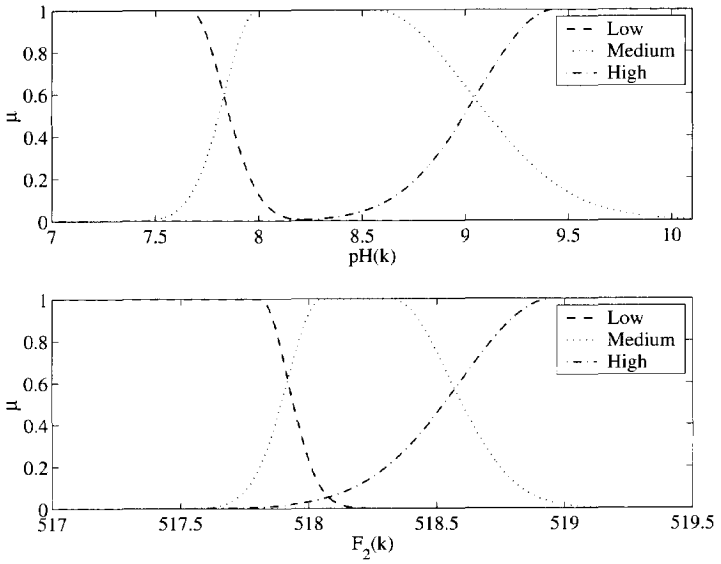


Figure 4.3. CSTR: Membership functions.

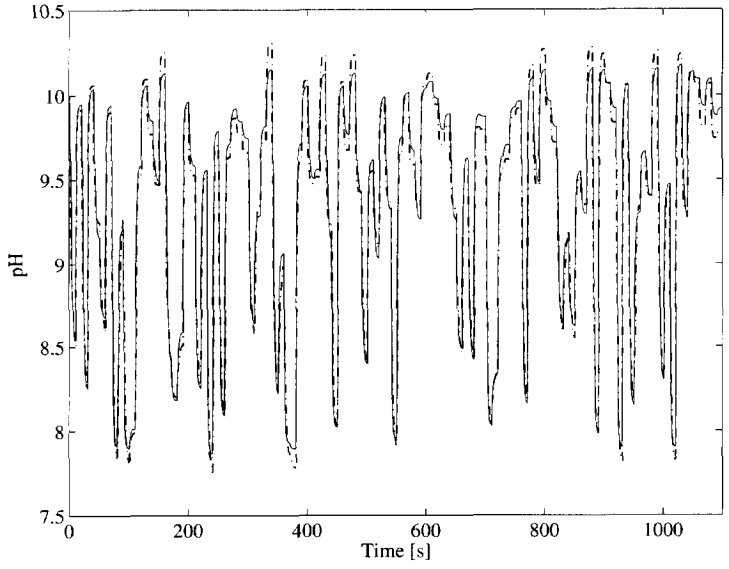


Figure 4.4. CSTR: validation of the fuzzy model (solid line: process output, dashed line: model prediction).

Controller configurations. The control objective is to follow set-point changes for the pH in the interval $[7.8, 8.8]$. Four different controller configurations are used to evaluate the performance of the optimization methods:

1. A linear MPC using a model obtained from the nonlinear process model at ($pH = 8.3$, $F_2 = 518.28$). This point is selected because it is the mean of the operating range for the pH;

2. A single model (SM) method (Section 4.3.1);

3. A multiple model (MM) method (Section 4.3.2);

4. A nonlinear optimization method, referred to as Nlinear, that directly uses the fuzzy model. It is based on the MATLAB[3] implementation of the Sequential Quadratic Programming (SQP) method (Coleman et al., 1999).

The methods are compared with identical tuning parameters set as follows. Since there is a delay of one sample, the minimum cost horizon $H_{\text{min}} = 1$ (Soeterboek, 1992). The process is well damped, therefore a prediction horizon of $H_{\text{p}} = 5$ is used to speed up the response (the settling time is about 120 s). For the same reason a control horizon of $H_{\text{c}} = 2$ is selected even though the fuzzy model is of order one. The acetic acid flow rate is kept at its nominal value $F_1 = 81$ l/min. The sodium hydroxide flow rate F_2 is subject to input level and rate constraints $510 \leq F_2 \leq 525$ [l/min] and $-2 \leq \Delta F_2 \leq 2$ [l/min^2]. The non-zero weights in the cost function (4.1) are $\mathbf{P} = 1.0$ and $\Delta\mathbf{Q} = 0.2$, as the latter is used to prevent aggressive actions. When the line search (LS) is applied, five iterations are used with the maximum merit function iterations set to 50.

Note that although better tuning parameters could be found, the above were selected on purpose to show the advantage of the successive linearization methods over the linear MPC.

State-space model extraction. To illustrate the extraction of a local linear state-space model from the fuzzy one, the first iteration in the MM controller is outlined. The initial process input and output are $F_2 = 517.5$ l/min and $pH = 7$, respectively. The fuzzy model state is $\boldsymbol{\xi} = 7$. To obtain an initial linear model $M(k)$, $k = 1$, the vector of degrees of fulfillment βs for each rule $\mathcal{R}_i$, $i = 1, 2, 3$, are calculated according to (4.28) using the cluster centers given in Tab. 4.3

$$\beta(517.5, 7) = [1.0,\ 0.0,\ 0.0].$$

These are then combined with the rules' consequents in (4.27) to get ζ^*, η^* and θ^*

[3] MATLAB is a registered trade mark of The Mathworks Inc., Natick, MA.

$$\zeta^* = \frac{\sum_{i=1}^{3} \beta_i \cdot \zeta_i}{\sum_{i=1}^{3} \beta_i} = \frac{[1.0\ 0.0\ 0.0] * \begin{bmatrix} 0.8677 \\ 0.9095 \\ 0.8165 \end{bmatrix}}{1.0} = 0.8677$$

$$\eta^* = \frac{\sum_{i=1}^{3} \beta_i \cdot \eta_i}{\sum_{i=1}^{3} \beta_i} = \frac{[1.0\ 0.0\ 0.0] * \begin{bmatrix} 0.0461 \\ 0.1867 \\ 0.1217 \end{bmatrix}}{1.0} = 0.0461$$

$$\theta^* = \frac{\sum_{i=1}^{3} \beta_i \cdot \theta_i}{\sum_{i=1}^{3} \beta_i} = \frac{[1.0\ 0.0\ 0.0] * \begin{bmatrix} -22.8847 \\ -96.0340 \\ -61.3953 \end{bmatrix}}{1.0} = -22.8847.$$

The corresponding $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$ matrices are

$$\mathbf{A}(k) = \begin{bmatrix} 0.8677 & -22.8847 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k) = \begin{bmatrix} 0.0461 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k) = [1\ 0].$$

The obtained matrices are used in the quadratic programme (4.22)–(4.25). The optimized control sequence $\mathcal{U}^* = [519.8612, 518.6208]$ is then applied to the fuzzy model in order to provide new linearization points. The degrees of fulfillment, βs, for the individual rules at these points are

$$\begin{aligned}
\beta(k+1) &= [0.0015,\ 0.2277,\ 0.7707] \\
\beta(k+2) &= [0.0000,\ 0.9081,\ 0.0918] \\
\beta(k+3) &= [0.0000,\ 0.9860,\ 0.0139] \\
\beta(k+4) &= [0.0000,\ 0.9872,\ 0.0127] \\
\beta(k+5) &= [0.0000,\ 0.9600,\ 0.0399]
\end{aligned}$$

for which the local state-space models $M(k+i)$, $k = 1$, $i = 1,...,5$ are

$$\mathbf{A}(k+1) = \begin{bmatrix} 0.8378 & -69.2250 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k+1) = \begin{bmatrix} 0.1364 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k+1) = [1\ 0]$$

$$\mathbf{A}(k+2) = \begin{bmatrix} 0.9009 & -92.8540 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k+2) = \begin{bmatrix} 0.1808 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k+2) = [1\ 0]$$

$$\mathbf{A}(k+3) = \begin{bmatrix} 0.9082 & -95.5506 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k+3) = \begin{bmatrix} 0.1858 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k+3) = [1\ 0]$$

$$\mathbf{A}(k+4) = \begin{bmatrix} 0.9083 & -95.5907 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k+4) = \begin{bmatrix} 0.1859 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k+4) = [1\ 0]$$

$$\mathbf{A}(k+5) = \begin{bmatrix} 0.9057 & -94.6492 \\ 0 & 1.0000 \end{bmatrix}, \quad \mathbf{B}(k+5) = \begin{bmatrix} 0.1841 \\ 0 \end{bmatrix}, \quad \mathbf{C}(k+5) = [1\ 0]$$

This set of linear models $\mathcal{M} = \{M(k+1), M(k+1), \ldots, M(k+5)\}$ again forms the quadratic programme. The optimized control sequence is $\mathcal{U}^* = [518.6760, 519.1455]$. After three more iterations $\mathcal{U}^* = [518.8833, 519.0402]$. The convergence of the optimized control sequence is shown in Fig. 4.5. The absolute variations in the control signal computed in the different iterations are rather small, but note that the range of F_2 is very narrow: $F_2 \in [516.5, 519.5]$. Therefore, the relative differences amounts up to 40%, see Tab. 4.4.

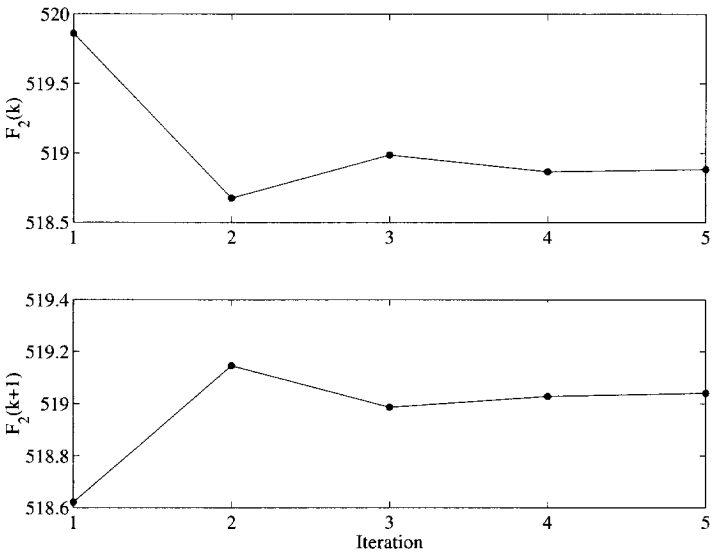


Figure 4.5. CSTR: Convergence of the control sequence.

Performance comparison. For performance comparison, two criteria are used that calculate output and input variations. The sum square error (SSE) criterion gives a quantitative measure, how close the process output $y(k)$ is to the reference $y_{\text{ref}}(k)$

$$SSE_y = \sum_{k=1}^{N} \left(y(k) - y_{\text{ref}}(k)\right)^2, \tag{4.29}$$

where N is the simulation length. To have a measure of how "large" the process input $u(k)$ is, we use the following criterion

$$V_u = \frac{1}{N} \sum_{k=1}^{N} u^2(k). \tag{4.30}$$

The criteria values for the different controllers are given in Table 4.5 on the facing page, where the ones obtained through the linear MPC are taken as 100%. A lower percentage means a better performance.

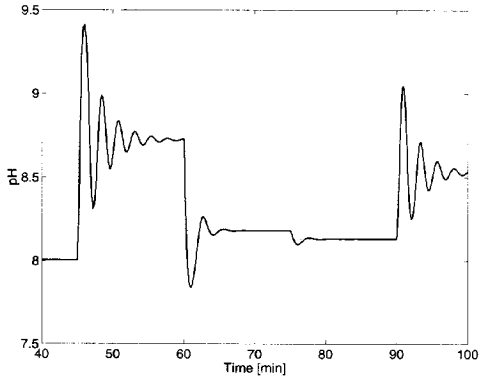
Table 4.4. CSTR: Variations in the sodium hydroxide stream at different iterations.

Iteration	$F_2(k)$			$F_2(k+1)$		
	Value	*Deviation*		*Value*	*Deviation*	
		Abs.	*Rel., %*		*Abs.*	*Rel., %*
1	519.8612	–	–	518.6208	–	–
2	518.6760	1.1852	39.50	519.1455	0.5247	17.49
3	518.9860	0.3100	10.33	518.9860	0.1595	5.32
4	518.8661	0.1199	3.99	519.0283	0.0423	1.41
5	518.8833	0.0172	0.57	519.0402	0.0119	0.40

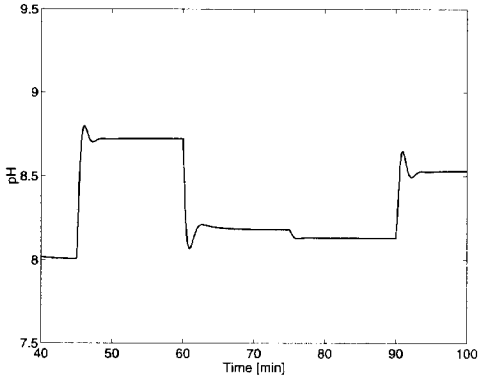
Table 4.5. CSTR: Relative values of the input and output variation criteria for the different controllers, in %. The linear MPC is taken as 100%. A lower percentage means a better performance.

Controller type	V_{pH}	$V_{\mathrm{F_2}}$	FLOPS
Linear MPC	100.0	100.0	100.0
SM	88.89	99.98	118
SM+LS	85.05	99.87	123
MM	79.11	99.98	149
MM+LS	79.11	99.97	153
Nlinear	162.44	102.4	374
Nlinear & SM	68.55	99.97	381
Nlinear & Linear	71.91	99.96	378

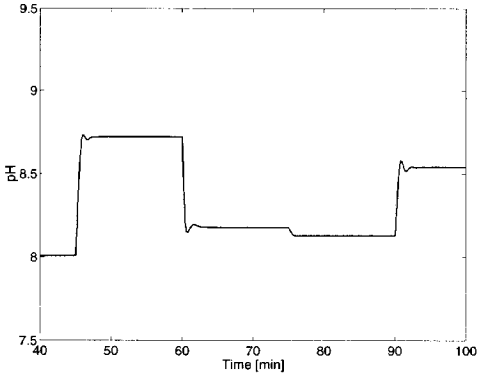
To assess the influence of the line search, the SM and MM performance is compared with and without the line search. As expected, the initial guess for the nonlinear optimization (Nlinear controller) turned out to be of crucial importance for the accuracy of the computed control action. A good initial guess can be obtained with the SM method (Section 4.3.1), see 'Nlinear & SM'. One can also use the linear MPC to provide such an initial guess, see 'Nlinear & Linear'.



(a) System performance with linear MPC.



(b) System performance with SM controller.



(c) System performance with Nlinear&SM controller.

Figure 4.6. CSTR: system performance with different controllers. Top: Linear MPC, middle: SM controller, bottom: Nlinear&SM controller.

One can see that the SM and MM lead to a significant improvement in the performance criteria over the values achieved through linear MPC. This improvement is, however, achieved at the expense of a higher computational load (FLOPS, floating-point operations). The introduction of the line search slightly improves the controller performance. When the nonlinear fuzzy model is used in the optimization routine, the computational load is approximately four times higher than for the linear MPC and the performance is poor. However, if the control sequence, obtained through a single iteration with SM or Linear MPC is used as an initial guess to the nonlinear optimization, a superior performance is achieved. Comparing this performance to the one obtained with MM, the improvement of about 10% is paid for with nearly 2.5 times as many FLOPS, increasing the time necessary for computation in a similar scale.

Figure 4.6 on the preceding page depicts the process behaviour for a part of the considered trajectory. When Linear MPC is used, the process behaviour is fairly oscillatory, while both SM and Nlinear & SM provide satisfactory performance with only a marginal superiority of the latter (achieved at the expense of a computational load approximately three times higher than the one for the SM).

4.5.2 Real-time liquid level control in a MIMO cascaded-tanks setup

We demonstrate the real-time performance of the presented technique with the setup described in Section 3.4. Again the control objective is to follow set-point changes in the levels in the lower two tanks by adjusting the flow rates of the liquid entering the upper tanks.

The single-model (SM) method and the multiple-model (MM) method are used to assess the influence of the models employed in the optimization on the achieved performance. The MPC parameters are selected according to the tuning rules proposed by Soeterboek (1992). Since there is a delay of one sample, the minimum cost horizon $H_{\text{min}} = 1$. The process is a well-damped second-order one, so a prediction horizon $H_{\text{p}} = 7$ and a control horizon $H_{\text{c}} = 2$ are used. The non-zero weighting matrices are $\mathbf{P} = \mathbf{I}$ and $\Delta\mathbf{Q} = 0.1\mathbf{I}$. The constraints on the inputs and outputs are set to

$$\begin{array}{llll} q_1 \in [0,\ 1.0] & q_2 \in [0,\ 1.0] & \Delta q_1 \in [-0.3,\ 0.3] & \Delta q_2 \in [-0.3,\ 0.3] \\ h_1 \in [0,\ 0.5] & h_2 \in [0,\ 0.5] & \Delta h_1 \in [-0.1,\ 0.1] & \Delta h_2 \in [-0.1,\ 0.1] \end{array} \quad (4.31)$$

Two first-order Butterworth filters with cut-off frequency $f_{\text{cut}} = 0.05$ Hz are used as feedback filters (Fig. 4.1). The cut-off frequency is one fourth of the sampling frequency ($f_{sampling} = 1/T_s = 0.2\,\text{Hz}$).

The process behaviour for the two controllers is presented in Fig. 4.7 and Fig. 4.9, respectively. While linear MPC (not shown in the figures) cannot stabilize the process, both SM and MM provide satisfactory performance, as the latter is slightly superior. The different degrees of coupling, noticeable in both figures, are due to the non-symmetric cross-connection. The cross-section area of the tube from tank 3 to tank 2 is larger than the corresponding area between 4 and 1. Therefore a change in h_1 level will have a larger affect on h_2 level than a change in h_2 on h_1. In the latter case the inputs have enough freedom to compensate for the change in the level, while in the former, the constraints and the weights imposed on the input increments prevent that.

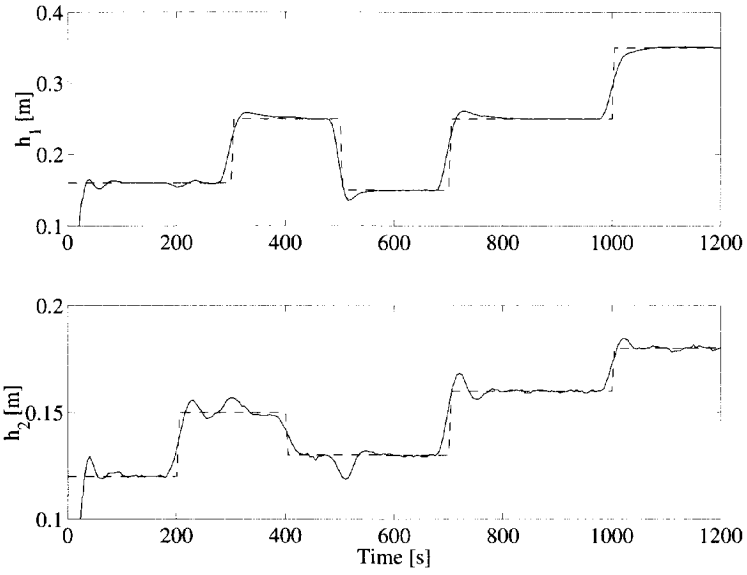


Figure 4.7. MIMO cascaded-tanks setup: real-time system performance with SM. Top: level h_1 bottom: level h_2. Dashed line: output reference, solid line: setup output.

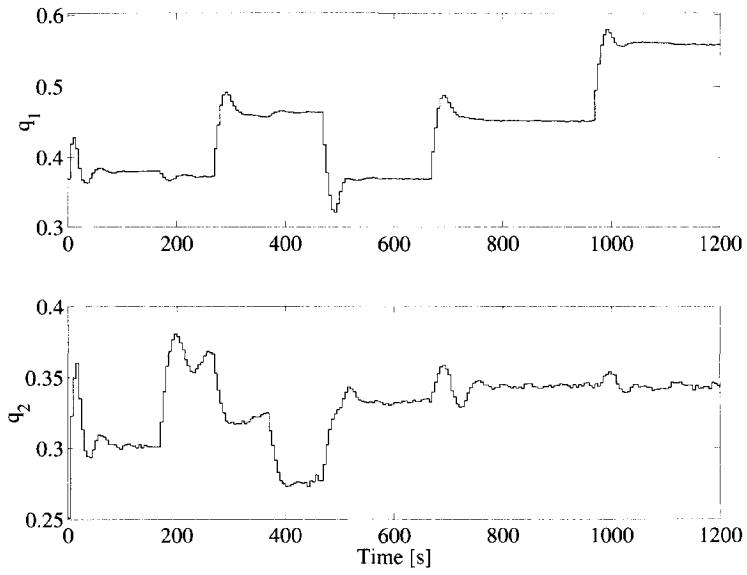


Figure 4.8. MIMO cascaded-tanks setup: control signal with SM. Top: q_1 bottom: q_2.

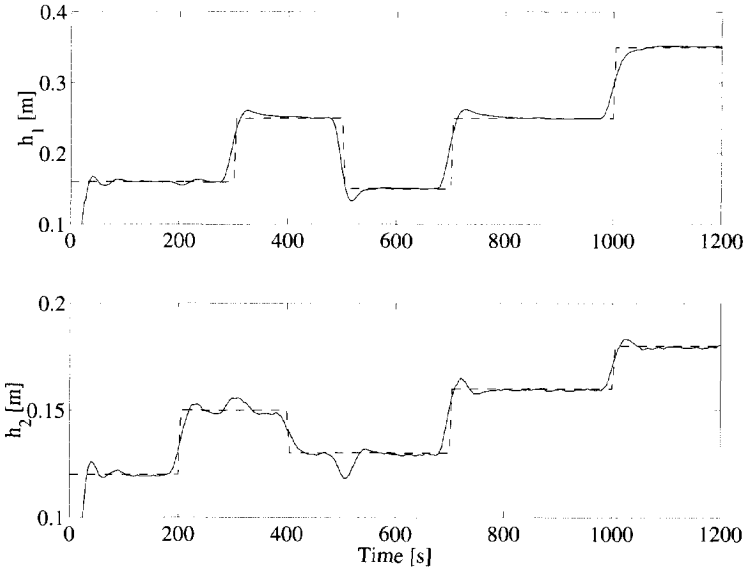


Figure 4.9. MIMO cascaded-tanks setup: real-time system performance with MM. Top: level h_1, bottom: level h_2. Dashed line: output reference, solid line: setup output.

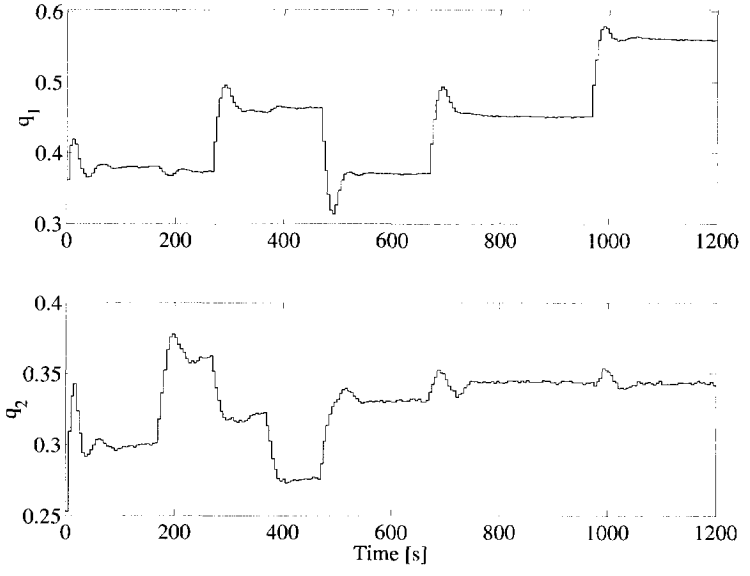


Figure 4.10. MIMO cascaded-tanks setup: control signal with MM. Top: q_1, bottom: q_2.

4.6 Summary and concluding remarks

Model predictive control is a powerful technique for the control of constrained nonlinear multi-variable processes. Various modifications of the original concept have been developed and also successfully applied in the industry. Nevertheless, there are two main problems that hamper a wider application of MPC: (i) the derivation of an accurate prediction model, which is not a trivial task, and (ii) the computational cost of the optimization problem that must be solved at each sampling period.

Fuzzy models of the Takagi–Sugeno (TS) type proved to be suitable for the use in nonlinear MPC because of their ability to accurately approximate complex nonlinear systems. This chapter presented an approach for using the TS fuzzy models in the MPC framework that avoids the time-consuming, non-convex optimization.

The proposed approach is based on linear parameter-varying (LTV) prediction models derived by fixing the parameters of the fuzzy model at a given operating point or a trajectory. In Section 4.3, algorithms for extracting a single model or a set of models along the predicted trajectory using the fuzzy model have been discussed. In Section 4.4 we showed how these linear models are used to construct the optimization problem in a straightforward manner, contrary to the cases where a nonlinear process model is used directly in the optimization problem.

Two benchmarks were presented (in Section 4.5) to illustrate the method. In the first one, the superiority of the fuzzy predictive controller over a linear MPC was demonstrated for pH control in a simulated continuous stirred tank reactor. The performance achieved by the latter is comparable to the performance obtained through nonlinear optimization, for which the computational load is considerably higher. The second benchmark showed the real-time performance of the fuzzy predictive controller.

The methods for model predictive control presented in this chapter are especially suited for high-quality control of nonlinear multivariable processes with short sampling time. For such processes, linear model predictive control does not give satisfactory results, while the optimization based on a nonlinear process model is too time consuming.

Among the proposed model predictive controllers, the best results were obtained using the multiple-model controller (Section 4.3), although the results were only slightly better than the results obtained by the single-model controller. The calculation time used by the latter is comparable to the time necessary for the linear model predictive controller and it would therefore be preferred for processes where the calculation time, compared to the sampling period, is a problem.

5 ROBUST STABILITY CONSTRAINTS FOR FUZZY MODEL PREDICTIVE CONTROL

This chapter addresses the robustness of the fuzzy model predictive controller discussed in Chapter 4. The goal is to obtain constraints on the control signal and its increment that guarantee closed-loop robust asymptotic stability for open-loop BIBO stable processes with an additive l_1-norm bounded model uncertainty. The next section states the robust stability problem in model predictive control. Section 5.2 presents the theoretical background. The idea is closely related to (small-gain-based) l_1-control theory, but due to the time-varying approach, the resulting robust stability constraints are less conservative. Bounds on the model uncertainty for models of the Takagi–Sugeno type are derived in Section 5.3. The fuzzy model is viewed as a linear time-varying model rather than as a nonlinear one. The incorporation of the robust stability constraints in the model predictive control scheme is described in Section 5.4, which also addresses nominal and robust performance and offset-free reference tracking. Robust asymptotic stability and offset-free reference tracking are achieved for asymptotically constant reference trajectories and disturbances. In Section 5.5, a simulation example and a real-time example are presented.

5.1 Problem statement

Although the TS model usually yields a reasonably accurate approximation of the process, one must keep in mind that a certain model-plant mismatch will always be present. The mismatch can be due to unmodeled dynamics, time-varying and/or aging phenomena, etc., and it will not only deteriorate the control performance; it may even destabilize the closed-loop system. The availability of tools for the design of a robustly stable predictive controller is thus of critical importance. So far, this aspect of MPC has only been addressed for linear time-invariant (LTI) process models, typically by using techniques based on infinite prediction horizon (Rawlings and Muske, 1993) or end-point constraints (Clarke and Scattolini, 1991; Mosca and Zhang, 1992).

The method presented in this chapter is an extension of the method proposed in (de Vries and van den Boom, 1997), where conditions are given that guarantee robust asymptotic stability for open-loop stable linear systems with an additive ∞-norm bounded model uncertainty. Here, similar conditions are derived for open-loop stable *nonlinear* systems with an additive ∞-norm bounded model uncertainty. Based on the uncertainty description, we derive level and rate constraints for the control signal that guarantee stability for any model-plant mismatch within the given bounds (Mollov, van den Boom, Cuesta, Ollero and Babuška, 2002).

First, we derive time-varying level and rate constraints for a predictive controller based on a general (i.e., possibly non-fuzzy) nonlinear model. These constraints guarantee bounded-input bounded-output (BIBO) stability for any model-plant mismatch within certain bounds. Then an algorithm is presented that estimates the uncertainty bounds when the nonlinear model is a fuzzy model of the Takagi–Sugeno type. The method makes use of the fact that at each sampling instant, new measurements become available and thus linear time-varying constraints can easily be incorporated in the design. Input constraints are recalculated at each sampling instant, based on a linear time-varying model derived from the fuzzy model. The resulting constraints are similar to (small-gain-based) l_1-control theory, but they are much less conservative as they are based on the uncertainty that actually occurs in the system, instead of on the "allowed worst-case model uncertainty." The uncertainty can also be viewed as a fuzzy number having for support the "allowed worst-case model uncertainty" (Setnes et al., 1998). Then, the uncertainty calculated at each sampling instant corresponds to a certain α-cut.

Similar ideas have also been investigated within the gain-scheduling framework (Shamma and Athans, 1990, 1991; Rugh, 1991). Shamma and Athans (1990, 1991) give sufficient conditions that guarantee that the overall gain-scheduled system will retain the feedback properties of the local designs when the process model is linear parameter varying or when it is nonlinear. If the gain-scheduling methodology is used, however, stability can only be guaranteed for sufficiently slow parameter variations. Masubuchi et al. (1998) proposed a method based on a (gain scheduling) parameters-dependent Lyapunov function that takes into account the rate of the parameter variations. While the conditions constructed in this way are both necessary and sufficient to guarantee stability, they may become conservative because of the fixed number of

off-line derived models. Our approach does not suffer from such conservatism, as the linear models are derived locally around the current operation point.

Although a well-tuned predictive controller is usually quite robust with respect to a model-plant mismatch (Soeterboek, 1992; Lee and Yu, 1994), a general theory dealing with the stability and robustness issues in predictive control has been missing. Therefore, the goal in this chapter is to provide the predictive controller with constraints on the control signal that ensure stability of the closed-loop control system. Usually stability is analyzed with respect to a given process model. However, often the process is complicated and/or varies over time, hence it is difficult to obtain a perfect model. A reasonable approach is to consider the process deviation from the available model as model uncertainty, such that the process behavior is always contained within the set of behaviors described by the model together with the associated uncertainty. Based on the current model uncertainty, at each sample constraints on the control signal and its increment are computed such that the control signal is in a (hyper)region where stability is guaranteed for all possible perturbations in the process (Fig. 5.1).

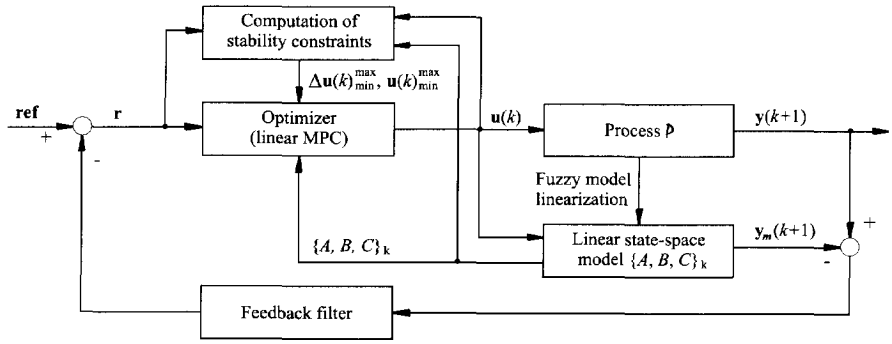


Figure 5.1. Fuzzy model-based predictive control with robust stability constraints.

The process $\mathcal{P}$ is considered to be Linear Time Varying (LTV) rather than nonlinear time invariant

$$\mathcal{P}: \mathbf{y}(k) = G(k)\mathbf{u}(k) + \mathbf{g}(k). \tag{5.1}$$

Denote the "true" process by $\mathcal{P}_t$

$$\mathcal{P}_t: \mathbf{y}_t(k) = G_t(k)\mathbf{u}(k) + \mathbf{g}_t(k), \tag{5.2}$$

and the nominal process and available model by $\mathcal{P}$. The offsets in the nominal and the "true" process $\mathbf{g}_t$ and $\mathbf{g}$ respectively, are assumed to be bounded. The model uncertainty Ω, defined as the deviation of the true process from its nominal model, is then given by (Fig. 5.2)

$$\left.\begin{array}{rcl} \Omega(k) & = & G_t(k) - G(k) \\ \omega(k) & = & \mathbf{g}_t(k) - \mathbf{g}(k) \end{array}\right\} \Rightarrow \qquad \mathbf{y}_t(k) - \mathbf{y}(k) = \Omega(k)\mathbf{u}(k) + \omega(k).$$

The TS fuzzy models belong to this class; compare (5.1) to the expression for the fuzzy model output (2.9). To remove the explicit dependence of the model output on its previous values, we express the fuzzy model as a time-varying convolution operator.

5.2 Derivation of robust stability constraints

This section presents *general* conditions that ensure robust stability of the control system despite variations in the controlled process. These conditions are valid for the class of open-loop BIBO stable processes that can be described through (5.1).

First, some definitions for l_p-stability are recalled (Vidyasagar, 1993), where the symbol l_p stands for the Lebesque spaces. For $p \in [1, \infty)$, l_p denotes the set of all measurable functions, $f\colon \mathbb{R}^+ \to \mathbb{R}$, whose pth powers are absolutely integrable over $[0, \infty)$, i.e., $\int_0^\infty |f(t)|^p dt < \infty$, while l_∞ denotes the set of essentially bounded measurable functions f.

Definition 5.1 *Let G be an input-output mapping on l_p. Then G is said to be l_p* **stable** *if $\mathbf{y} = G(\mathbf{u})$, $\mathbf{u} \in l_p \Rightarrow \mathbf{y} \in l_p$.*

Definition 5.2 *G is l_p stable* **with finite gain (wfg)** *if it is l_p stable, and in addition there exist finite real constants γ_p and b_p, such that*

$$\|\mathbf{y}\|_p \leq \gamma_p \|\mathbf{u}\|_p + b_p, \tag{5.3}$$

where $\|\cdot\|_p = \int_0^\infty |\cdot|^p dt$ is the p-norm of the corresponding signal.

Remarks:

1. l_p-stability **wfg** implies l_p-stability.
2. If G is an l_p-stable wfg input-output mapping, then the l_p-gain $\gamma_p(G)$ of G is the minimal value for which there exists a non-negative parameter b_p, such that (5.3) holds: $\gamma_p(G) = \inf\{\gamma_p : \exists b_p \geq 0 \text{ such that (5.3) holds}\}$.
3. The 1-norm (or the induced ∞-norm) of G is defined as
$$\|G\|_1 = \|G\|_{i,\infty} = \max_{\mathbf{u}} \frac{\|\mathbf{y}\|_\infty}{\|\mathbf{u}\|_\infty}. \tag{5.4}$$
4. Below BIBO stability and l_∞-stability, **wfg** has the same meaning.

Figure 5.2 on the facing page depicts a scheme which is solely used to compute the stability bounds on the control signal and its increment. It is a general two-degree-of-freedom control scheme in which all the blocks are input-output mappings. These mapping are used to obtain the signal relations that are necessary to derive the stability constraints.

It is assumed that the process is described by l_∞-stable wfg mappings $G, \Omega\colon \mathbb{R}^m \to \mathbb{R}^p$, which denote the nominal model and the additive model uncertainty, respectively. The mappings $R, Q : \mathbb{R}^p \to \mathbb{R}^m$ are the controllers to be designed. The signal $\mathbf{y}$ is the output of the process, $\mathbf{u}_{\text{fb}}$ is the output of the feedback controller Q, $\mathbf{u}_{\text{ff}}$ is the output of the feedforward controller R, which filters the reference trajectory $\mathbf{r}$, and $\mathbf{u}$ is the output of the two-degree-of-freedom controller. The signals $\mathbf{d}$ and ω are additive disturbances on the input and the output of the process, respectively.

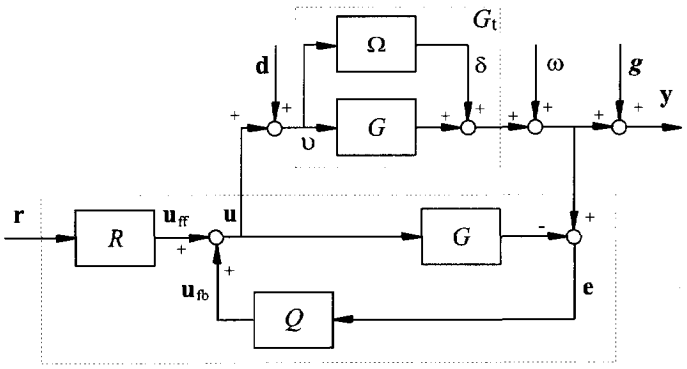


Figure 5.2. A general IMC scheme which is *only* used to compute the stability constraints on the control signal and on its increment.

Recall that ω is the difference between the offsets of the true (perturbed) process and of the nominal model, $\omega = \mathbf{g}_t - \mathbf{g}$, such that the parallel connection of the mappings G and Ω is offset-free. To retain the input-output mapping (5.1), we add the model offset $\mathbf{g}$ outside the feedback loop. It is also assumed that the model exactly represents the nominal process G, i.e., the model-plant mismatch is contained in the additive uncertainty Ω.

Denote $\mathbf{u}_2(k) = \omega(k) + G\big(\mathbf{d}(k)\big)$. It can be seen that the only feedback loop in the controlled system is the one depicted in Fig. 5.3. Hence, the problem of guaranteeing robust internal stability of the controlled system given in Fig. 5.2 can be reduced to the general problem of guaranteeing that the feedback system of Fig. 5.3 is robustly internally stable.

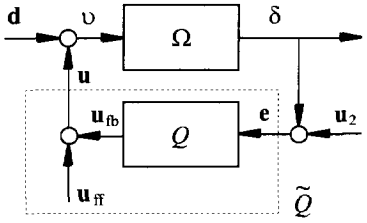


Figure 5.3. Feedback system for internal robust stability analysis, $\mathbf{u}_2(k) = \omega(k) + G(\mathbf{d}(k))$.

In nonlinear systems, the incremental input-output mappings are different from the non-incremental ones (contrary to linear systems). Therefore, to describe the feedback system Fig. 5.3, both the incremental (denoted through Δ) and the non-incremental mappings are necessary:

$$\begin{aligned} \boldsymbol{v}(k) &= \mathbf{d}(k)+\mathbf{u}(k) & \mathbf{e}(k) &= \mathbf{u}_2(k)+\boldsymbol{\delta}(k) \\ \boldsymbol{\delta}(k) &= \Omega(k)\big(\boldsymbol{v}(k)\big) & \mathbf{u}_{\text{fb}}(k) &= Q(k)\big(\mathbf{e}(k)\big) \\ \mathbf{u}(k) &= \tilde{Q}(k)\big(\mathbf{e}(k)\big) = \mathbf{u}_{\text{ff}}(k)+\mathbf{u}_{\text{fb}}(k) & & \end{aligned} \tag{5.5a}$$

$$\begin{aligned} \Delta\boldsymbol{v}(k) &= \Delta\mathbf{d}(k)+\Delta\mathbf{u}(k) & \Delta\mathbf{e}(k) &= \Delta\mathbf{u}_2(k)+\Delta\boldsymbol{\delta}(k) \\ \Delta\boldsymbol{\delta}(k) &= \Omega_\Delta(k)\big(\Delta\boldsymbol{v}(k)\big) & \Delta\mathbf{u}_{\text{fb}}(k) &= Q_\Delta(k)\big(\Delta\mathbf{e}(k)\big) \\ \Delta\mathbf{u}(k) &= \tilde{Q}_\Delta(k)\big(\Delta\mathbf{e}(k)\big) = \Delta\mathbf{u}_{\text{ff}}(k)+\Delta\mathbf{u}_{\text{fb}}(k). & & \end{aligned} \tag{5.5b}$$

The stability constraints proposed in (de Vries and van den Boom, 1997) were derived for convolution systems

$$\mathbf{y}(k) = G\big(\mathbf{u}(k)\big) = \sum_{\tau=-\infty}^{\infty} \mathbf{H}_G(k-\tau)\mathbf{u}(\tau), \quad k \in \mathbb{Z} \tag{5.6}$$

where the input-output mapping G is a MIMO LTI system. The matrix $\mathbf{H}_G(k-\tau)$ contains the time-invariant *Markov parameters* or *kernels* of the system (Kwakernaak and Sivan, 1991). The system is called causal (or strictly causal) if $\mathbf{H}_G(k-\tau) = 0, \forall\tau > k$ (or $\tau \geq k$).

The following theorem gives the constraints on the controller mapping Q (and $\tilde{Q}$) in the feedback system (5.5), which guarantee that this feedback system is globally internally asymptotically stable for different assumptions on Ω and/or Q.

Theorem 5.1 *Consider the feedback system (5.5), where Q and Ω are l_∞-stable LTI convolution operators, Q is strictly causal and $\|\Omega\|_{i,\infty} \leq \epsilon_\Omega < \infty$. Under these conditions the system is asymptotically stable if $\|Q\|_{i,\infty} < 1/\epsilon_\Omega$.*

Proof: See Section 6.6 in (Vidyasagar, 1993). □

As the process model is time varying and contains an offset term, both the uncertainty mapping Ω and its increment $\Delta\Omega$ are time varying, and cannot be written directly in the form (5.6). To guarantee stability under these conditions, we first introduce some necessary concepts and then give a new theorem.

Assume the process model (5.2) to be l_∞-stable with finite gain (wfg)

$$\begin{aligned} \gamma_\infty &= \max_{\mathbf{u}} \frac{\|G(k)\mathbf{u}(k)\|_\infty}{\|\mathbf{u}(k)\|_\infty} & \text{and} \quad \gamma_\infty < \infty \\ b_\infty &= \max_{\mathbf{u}} \|\mathbf{g}(k)\|_\infty & \text{and} \quad b_\infty < \infty, \end{aligned}$$

that is,

$$\|\mathbf{y}(k)\|_\infty \leq \gamma_\infty \|\mathbf{u}(k)\|_\infty + b_\infty.$$

Usually the offset $\mathbf{g}$ is regarded as a constant (Vidyasagar, 1993). However, for the sake of generality, here it may be time varying.

Thus for the uncertainty Ω and its increment Ω_Δ in (5.5), and the offset ω and its increment ω_Δ it holds that

$$\|\Omega(k)\|_{i,\infty} = \max_{\boldsymbol{v}} \frac{\|\Omega(k)\boldsymbol{v}(k)\|_\infty}{\|\boldsymbol{v}(k)\|_\infty} = \gamma_\infty \qquad \text{and} \quad |\omega(k)| < b_\infty$$

$$\|\Omega_\Delta(k)\|_{i,\infty} = \max_{\Delta\boldsymbol{v}} \frac{\|\Omega_\Delta(k)\Delta\boldsymbol{v}(k)\|_\infty}{\|\Delta\boldsymbol{v}(k)\|_\infty} = \gamma_{\Delta,\infty} \qquad \text{and} \quad |\omega_\Delta(k)| < b_{\Delta,\infty}\,.$$

The corresponding theorem states the conditions for BIBO stability of the LTV system:

Theorem 5.2 *Consider the feedback system (5.5), where $Q(k)$ and $\Omega(k)$ are l_∞-stable wfg LTV convolution operators, $Q(k)$ is strictly causal and $\|\Omega(k)\|_{i,\infty} \le \epsilon_\Omega < \infty$. Under these conditions, the system (5.5) is BIBO stable if $\|Q(k)\|_{i,\infty} < 1/\epsilon_\Omega$.*

Proof: The proof is similar to the proof of Theorem 5.1 (Section 6.6 in (Vidyasagar, 1993)). □

The constraint $\|Q(k)\|_{i,\infty} < 1/\epsilon_\Omega$ is sufficient to guarantee stability for all possible perturbations which satisfy $\|\Omega(k)\|_{i,\infty} \le \epsilon_\Omega$. However, if the true perturbation is only a small subset of all possible perturbations like, e.g., one fixed (but unknown) convolution operator, this constraint is very conservative. Less conservative stability constraints can be derived using the fact that all the signals in (5.5) are known up to time k.

Theorem 5.3 *Consider the feedback system (5.5), where $Q(k)$, $Q_\Delta(k)$, $\Omega(k)$ and $\Omega_\Delta(k)$ are finite-order l_∞-stable wfg mappings of the form (5.6). Further, let $Q(k)$ and $Q_\Delta(k)$ be causal, $\Omega(k)$ and $\Omega_\Delta(k)$ be strictly causal, and $\|\Omega(k)\|_{i,\infty} \le \epsilon_\Omega < \infty$ and $\|\Omega_\Delta(k)\|_{i,\infty} \le \epsilon_{\Omega_\Delta} < \infty$. Under these conditions the system (5.5) is BIBO stable if at least one of the following three constraints is satisfied.*

Ca: $\|Q(k)\|_{i,\infty} < 1/\epsilon_\Omega$ *and* $\|Q_\Delta(k)\|_{i,\infty} < 1/\epsilon_{\Omega_\Delta}$.

Cb: $\Big|Q(k)\Big(\Omega(k)\big(\boldsymbol{v}(k)\big)\Big)\Big| < \max_{0<i\le N} |\boldsymbol{v}(k-i)|$ *and* $\Big|Q_\Delta(k)\Big(\Omega_\Delta\big(\Delta\boldsymbol{v}(k)\big)\Big)\Big| < \max_{0<i\le N_\Delta} |\Delta\boldsymbol{v}(k-i)|\ \forall k$ *where N and N_Δ are integers such that $0 < N \le N(Q) + N(\Omega)$ and $0 < N_\Delta \le N(Q_\Delta) + N(\Omega_\Delta)$, where $N(\cdot)$ denotes the order of the corresponding mapping.*

Cc: $\Big|Q(k)\Big(\Omega(k)\big(\boldsymbol{v}(k)\big)\Big)\Big| \le \bar{P}|\boldsymbol{v}(k)|$ *and* $\Big|Q_\Delta(k)\Big(\Omega_\Delta\big(\Delta\boldsymbol{v}(k)\big)\Big)\Big| \le \bar{P}_\Delta|\boldsymbol{v}(k)|\ \forall k$ *where $\bar{P}$ and $\bar{P}_\Delta$ are arbitrary strictly causal l_∞-stable wfg operators of the form (5.6) in which the the Markov parameters are greater than or equal to zero. $\bar{P}$ and $\bar{P}_\Delta$ have finite orders $N(\bar{P}) = N(Q) + N(\Omega)$ and $N(\bar{P}_\Delta) = N(Q_\Delta) + N(\Omega_\Delta)$, and induced ∞-norms which satisfy $\|\bar{P}\|_{i,\infty} < 1$ and $\|\bar{P}_\Delta\|_{i,\infty} < 1$.*

Proof:

Constraint Ca. Let $\|\Omega(k)\|_{i,\infty} = \alpha_1 < \epsilon_\Omega$ and $\|Q(k)\|_{i,\infty} = \alpha_2 < 1/\epsilon_\Omega$, thus $\alpha_1\alpha_2 < 1$. To prove that system (5.5) is l_∞ stable wfg (BIBO stable) when **Ca** is used, consider

the external signal $\boldsymbol{v}(k)$

$$\begin{aligned}
\|\boldsymbol{v}(k)\|_\infty &= \big\|\mathbf{d}(k)+\mathbf{u}_{\mathrm{ff}}(k)+Q(k)\big(\mathbf{e}(k)\big)\big\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \big\|Q(k)\big(\mathbf{e}(k)\big)\big\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \|Q(k)\|_{i,\infty}\|\mathbf{e}(k)\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \|Q(k)\|_{i,\infty} + \big\|\Omega(k)\big(\boldsymbol{v}(k)\big)+\mathbf{u}_2(k)\big\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \|Q(k)\|_{i,\infty}\big(\|\Omega(k)\|_{i,\infty}\|\boldsymbol{v}(k)\|_\infty + \|\mathbf{u}_2(k)\|_\infty\big) \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \alpha_1\alpha_2\|\boldsymbol{v}(k)\|_\infty + \alpha_2\|\mathbf{u}_2(k)\|_\infty.
\end{aligned}$$

The above is equivalent to

$$(1-\alpha_1\alpha_2)\|\boldsymbol{v}(k)\|_\infty \le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \alpha_2\|\mathbf{u}_2(k)\|_\infty$$

or

$$\|\boldsymbol{v}(k)\|_\infty \le \frac{1}{1-\alpha_1\alpha_2}\|\mathbf{d}(k)\|_\infty + \frac{1}{1-\alpha_1\alpha_2}\|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \frac{\alpha_2}{1-\alpha_1\alpha_2}\|\mathbf{u}_2(k)\|_\infty.$$

Since $\frac{1}{1-\alpha_1\alpha_2} < \infty$ and $\frac{\alpha_2}{1-\alpha_1\alpha_2} < \infty$, and $\mathbf{d}, \mathbf{u}_{\mathrm{ff}} \in l_\infty^m$ and $\mathbf{u}_2 \in l_\infty^n$, we have that $\boldsymbol{v} \in l_\infty^m$. This in turn implies that $\mathbf{u}, \mathbf{u}_{\mathrm{fb}} \in l_\infty^m$, and from the l_∞-stability wfg of Ω we have that $\boldsymbol{\delta}, \mathbf{e} \in l_\infty^m$, which leads to the conclusion that the system (5.5) is l_∞-stable.

The proof for the incremental relations proceeds along the same lines.

Constraint Cb. The constraint **Cb** is a special case of **Cc** when the Markov parameter $p_i(k)$ (in $\bar{P}$), corresponding to the maximum value of $|\boldsymbol{v}(k-i)|$ for $0 < i \le N(\bar{P})$ $\big(N(\bar{P}) = N(Q) + N(\boldsymbol{\Omega})\big)$ is set to ρI with $0 \ll \rho < 1$,
$\bar{P}|\boldsymbol{v}(k)| = \max_{0<i\le N(P)} |\boldsymbol{v}(k-i)|$.

Constraint Cc. The proof that system (5.5) is l_∞ stable wfg when **Cc** is used follows the same reasoning as the proof for the **Ca** case

$$\begin{aligned}
\|\boldsymbol{v}(k)\|_\infty &= \Big\|\mathbf{d}(k)+\mathbf{u}_{\mathrm{ff}}(k)+Q(k)\big(\mathbf{u}_2(k)\big)+Q(k)\Big(\Omega\big(\boldsymbol{v}(k)\big)\Big)\Big\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \big\|Q(k)\big(\mathbf{u}_2(k)\big)\big\|_\infty + \Big\|Q(k)\Big(\Omega(k)\big(\boldsymbol{v}(k)\big)\Big)\Big\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \|Q(k)\|_{i,\infty}\|\mathbf{u}_2(k)\|_\infty + \|\bar{P}|\boldsymbol{v}(k)|\|_\infty \\
&\le \|\mathbf{d}(k)\|_\infty + \|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \alpha_2\|\mathbf{u}_2(k)\|_\infty + \|\bar{P}\|_{i,\infty}\|\boldsymbol{v}(k)\|_\infty.
\end{aligned}$$

Letting $\|\bar{P}\|_{i,\infty} \le \epsilon_{\bar{P}} < 1$, we get

$$\|\boldsymbol{v}(k)\|_\infty \le \frac{1}{1-\epsilon_{\bar{P}}}\|\mathbf{d}(k)\|_\infty + \frac{1}{1-\epsilon_{\bar{P}}}\|\mathbf{u}_{\mathrm{ff}}(k)\|_\infty + \frac{\alpha_2}{1-\epsilon_{\bar{P}}}\|\mathbf{u}_2(k)\|_\infty.$$

Since $\frac{1}{1-\epsilon_{\bar{P}}} < \infty$ and $\frac{\alpha_2}{1-\epsilon_{\bar{P}}} < \infty$, and $\mathbf{d}, \mathbf{u}_{\mathrm{ff}} \in l_\infty^m$ and $\mathbf{u}_2 \in l_\infty^n$, we have that $\boldsymbol{v} \in l_\infty^m$. In turn this implies that $\mathbf{u}, \mathbf{u}_{\mathrm{fb}} \in l_\infty^m$, and from the l_∞-stability wfg of Ω we have that $\boldsymbol{\delta}, \mathbf{e} \in l_\infty^m$, which leads to the conclusion that the system (5.5) is l_∞-stable.

The proof for the incremental relations proceeds along the same lines. □

Constraint **Ca** is just as conservative as the constraint $\|Q(k)\|_{i,\infty} < 1/\epsilon_\Omega$ in Theorem 5.2 and it guarantees that $\|Q\Omega\|_{i,\infty} = \sup_{\mathbf{u}} \|Q(\Omega(\mathbf{u}))\|_\infty / \|\mathbf{u}\|_\infty < 1, \forall \mathbf{u} \in l_\infty^m$. Constraints **Cb** and **Cc** guarantee $\|Q(k)(\delta(k))\|_\infty / \|v(k)\|_\infty < 1$ for the specific realization of the signals $v(k)$ and $\delta(k) = \Omega(k)(v(k))$ that actually occur in the system. They can be less conservative than **Ca**, depending on the exact situation in the system. In other words, any of the constraints **Ca**, **Cb** and **Cc** guarantees stability for all possible perturbations that satisfy $\|\Omega(k)\|_{i,\infty} \leq \epsilon_\Omega$, but **Cb** and **Cc** automatically adapt to the true realization of the output of $\Omega(k)$, which usually makes them less conservative (de Vries and van den Boom, 1997). The following theorems provide more easily implementable versions of the constraints **Ca**, **Cb** and **Cc**. First, the constraints are defined with respect to some upper bounds of Q, Q_Δ, Ω and Ω_Δ since the actual mappings are not known. Then the current "worst-case" upper bounds and input-output disturbances are estimated.

Theorem 5.4 *Consider the feedback system (5.5), where $Q(k)$, $Q_\Delta(k)$, $\Omega(k)$ and $\Omega_\Delta(k)$ are operators that are l_∞-stable wfg $\forall\ k \geq 0$, $\forall p \in [1, \infty)$. Let $Q(k)$ and $Q_\Delta(k)$ be causal and $\Omega(k)$ and $\Omega_\Delta(k)$ be strictly causal, $\|\Omega(k)\|_{i,\infty} \leq \epsilon_\Delta < \infty$ and $\|\Omega_\Delta(k)\|_{i,\infty} \leq \epsilon_{\Omega_\Delta} < \infty$. Further, let $\bar{\Omega}$ and $\bar{\Omega}_\Delta$ be strictly causal, finite-order l_∞-stable wfg operators of the form*

$$\bar{X}(\mathbf{u}(k)) = \sum_{\tau=k-N_X}^{\infty} \mathbf{H}_{\bar{X}}(k,\tau)|\mathbf{u}(\tau)|, \quad k \in \mathbb{Z}, \quad \bar{X} = \bar{\Omega} \text{ or } \bar{\Omega}_\Delta \tag{5.7}$$

that satisfy the following assumptions: $\|\bar{\Omega}\|_{i,\infty} < \epsilon_\Omega$, $\|\bar{\Omega}_\Delta\|_{i,\infty} < \epsilon_{\Omega_\Delta}$ and $|\Omega(k)(v(k))| < \bar{\Omega}(v(k))$, $|\Omega_\Delta(\Delta v(k))| < \bar{\Omega}_\Delta(\Delta v(k)), \forall k$.

Define arbitrary causal l_∞-stable wfg mappings $\bar{Q}$ and $\bar{Q}_\Delta$ of the form (5.7) with $\|\bar{Q}\|_{i,\infty} < 1/\epsilon_\Omega$ and $\|\bar{Q}_\Delta\|_{i,\infty} < 1/\epsilon_{\Omega_\Delta}$. Then the system (5.5) is BIBO stable if at least one of the following two constraints is satisfied:

$$\mathbf{C1}: |\mathbf{u}(k)| \leq \bar{Q}(\mathbf{e}(k)) + |\mathbf{u}_{\text{ff}}(k)| \text{ and } |\Delta\mathbf{u}(k)| \leq \bar{Q}_\Delta(\Delta\mathbf{e}(k)) + |\Delta\mathbf{u}_{\text{ff}}|, \ \forall k.$$

$$\mathbf{C2}: |\mathbf{u}(k)| \leq \bar{Q}\Big(\bar{\Omega}(v(k))\Big) + \bar{Q}(\mathbf{u}_2(k)) + |\mathbf{u}_{\text{ff}}(k)| \quad \text{and}$$

$$|\Delta\mathbf{u}(k)| \leq \bar{Q}_\Delta\Big(\bar{\Omega}_\Delta(\Delta v(k))\Big) + \bar{Q}_\Delta(\Delta\mathbf{u}_2(k)) + |\Delta\mathbf{u}_{\text{ff}}(k)|, \ \forall k$$

where

$$\mathbf{u}(k) = Q(k)\Big(\Omega(k)(v(k) + \mathbf{u}_2(k))\Big) + \mathbf{u}_{\text{ff}}(k) \quad \text{and}$$

$$\Delta\mathbf{u}(k) = Q_\Delta(k)\Big(\Omega_\Delta(k)(\Delta v(k) + \Delta\mathbf{u}_2(k))\Big) + \Delta\mathbf{u}_{\text{ff}}(k).$$

Proof:

The proof of Theorem 5.4 is based on the fact that upper bounds can be found that guarantee stability for all the internal signals in system (5.5), by analyzing $\Delta v(k)$ and $v(k)$ explicitly. □

Constraint **C2** still cannot be implemented as $\bar{\Omega}$, $\bar{\Omega}_\Delta$, $v(k)$ and $\mathbf{u}_2(k)$ are unknown in practice. In the following corollary two constraints are specified which are closely related to **C2** but which *can be implemented.* This is achieved by using the measurable signal $\mathbf{e}(k)$ instead of the non-measurable $v(k)$ and/or $\mathbf{u}_2(k)$ and by using estimates of the worst-case upper bounds $\bar{\Omega}$ and $\bar{\Omega}_\Delta$.

Corollary 5.1 *Let $\bar{\Omega}_e(k)$ and $\bar{\Omega}_{\Delta_e}(k)$ be arbitrary strictly causal l_∞-stable wfg mappings $\mathbb{R}^m \to \mathbb{R}^p$ of the form (5.7) with $\|\bar{\Omega}_e(k)\|_{i,\infty} \le 1/\epsilon_\Omega$ and $\|\bar{\Omega}_{\Delta_e}(k)\|_{i,\infty} \le 1/\epsilon_{\Omega_\Delta}$. The operators $\bar{\Omega}_e(k)$ and $\bar{\Omega}_{\Delta_e}(k)$ can be seen as estimates of $\bar{\Omega}$ and $\bar{\Omega}_\Delta$ at the current time instant. Then constraint* **C2** *in Theorem 5.4 can be replaced by either of the following*

$$\textbf{C3}: |\mathbf{u}(k)| \le \bar{Q}\big(\bar{\mathbf{e}}(k)\big) + |\mathbf{u}_{\text{ff}}(k)| \quad \textit{and} \quad |\Delta\mathbf{u}(k)| \le \bar{Q}_\Delta\big(\Delta\bar{\mathbf{e}}(k)\big) + |\Delta\mathbf{u}_{\text{ff}}(k)|, \ \forall k$$

where

$$\bar{\mathbf{e}}(k) = \max\big\{\bar{\Omega}_e(k)\big(\mathbf{u}(k)\big), |\mathbf{e}(k)|\big\} \quad \textit{and}$$
$$\Delta\bar{\mathbf{e}}(k) = \max\big\{\bar{\Omega}_{\Delta_e}(k)\big(\Delta\mathbf{u}(k)\big), |\Delta\mathbf{e}(k)|\big\}, \ \forall k.$$

$$\textbf{C4}: \ |\mathbf{u}(k)| \le \bar{\mathbf{u}}(k), \qquad \bar{\mathbf{u}}(k) = \bar{Q}(k)\big(\bar{\mathbf{e}}(k)\big) + |\mathbf{u}_{\text{ff}}(k)| \quad \textit{and}$$
$$|\Delta\mathbf{u}(k)| \le \Delta\bar{\mathbf{u}}(k), \quad \Delta\bar{\mathbf{u}}(k) = \bar{Q}_\Delta\big(\Delta\bar{\mathbf{e}}(k)\big) + |\Delta\mathbf{u}_{\text{ff}}(k)|, \ \forall k$$

where

$$\bar{\mathbf{e}}(k) = \max\big\{\bar{\Omega}_e(k)\big(\bar{\mathbf{u}}(k)\big), |\mathbf{e}(k)|\big\} \quad \textit{and}$$
$$\Delta\bar{\mathbf{e}}(k) = \max\big\{\bar{Q}_{\Delta_e}(k)\big(\Delta\bar{\mathbf{u}}(k)\big), |\Delta\mathbf{e}(k)|\big\}, \ \forall k.$$

Proof:

The proof of Corollary 5.1 is based on the same principles as the proof of Theorem 5.4 except that $\Delta\mathbf{u}(k)$ (or $\Delta\bar{\mathbf{u}}(k)$) should be analyzed explicitly for the case that $\Delta\bar{\mathbf{e}}(k) = \bar{\Omega}_{\Delta_e}(k)\big(\Delta\mathbf{u}(k)\big)$ or $\Delta\bar{\mathbf{e}}(k) = \bar{\Omega}_{\Delta_e}(k)\big(\Delta\bar{\mathbf{u}}(k)\big)$, which, together with the result **C2** of Theorem 5.4, form the basis of the proof. □

Remarks:

- The interpretation of **C3** and **C4** is the same as that of **Cb** and **Cc**. There is a close relation between **C1** and **Ca**, but **C1** can be implemented in a much simpler and computationally faster form than **Ca**.
- The upper bounds $\bar{\Omega}$ and $\bar{\Omega}_\Delta$ in Theorem 5.4 only need to exist. If **C1**, **C3** or **C4** is used, they do not need to be known.

  Below we derive such bounds for the nonlinear Takagi–Sugeno fuzzy model.

5.3 Fuzzy model as a convolution operator

In order to remove the explicit dependence of the model output on its previous values, the fuzzy model (2.2) is represented as an LTV, l_1-stable wfg convolution operator of

the form (5.6)

$$\begin{aligned}\mathbf{y}(k) &= \mathcal{R}\big(\mathbf{u}(k)\big) \\ &= G(k)\mathbf{u}(k) + \mathbf{g}(k) \\ &= \sum_{i=0}^{\infty} \mathbf{H}_{\mathrm{y,tv}}(k,i)\mathbf{u}(k-i) + \mathbf{g}(k)\end{aligned} \tag{5.8}$$

where $\mathbf{H}_{\mathrm{y,tv}}(k,i)$ is the time-varying matrix of the system's kernels, which will be obtained later on.

To find an upper bound for the (unstructured) model uncertainty Ω, we use the induced ∞-norm, as it accounts for the ∞-norms of the input and output signals

$$\|\Omega\|_{i,\infty} = \max_{\boldsymbol{\upsilon}} \frac{\|\boldsymbol{\delta}(k)\|_\infty}{\|\mathbf{d}(k)\|_\infty}. \tag{5.9}$$

Recall that (Fig. 5.2)

$$\|\boldsymbol{\delta}(k)\|_\infty = \max_k |\boldsymbol{\delta}(k)| \quad \text{and}$$

$$\boldsymbol{\delta}(k) = \sum_{i=0}^{\infty} \mathbf{H}_{\delta,\mathrm{tv}}(k,i)\boldsymbol{\upsilon}(k-i), \tag{5.10}$$

then

$$\begin{aligned}\|\boldsymbol{\delta}(k)\|_\infty &= \max_k |\boldsymbol{\delta}(k)| \\ &\le \max_k \sum_{i=0}^{\infty} |\mathbf{H}_{\delta,\mathrm{tv}}(k,i)| \cdot |\boldsymbol{\upsilon}(k-i)| \\ &\le \max_k \sum_{i=0}^{\infty} |\mathbf{H}_{\delta,\mathrm{tv}}(k,i)| \\ &\le \sum_{i=0}^{\infty} \max_k |\mathbf{H}_{\delta,\mathrm{tv}}(k,i)| \le \sum_{i=0}^{\infty} |\mathbf{H}_{\delta,\mathrm{tv,max}}(i)|.\end{aligned} \tag{5.11}$$

To find an expression for the kernels $\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$, we write the fuzzy model in a specific state-space form. Since the model is assumed to match exactly the nominal process, we give an extended description including both the process and the model

$$\mathbf{x}_{\mathrm{t}}(k+1) = \mathbf{A}_{\mathrm{t}}(k)\mathbf{x}_{\mathrm{t}}(k) + \mathbf{B}_{\mathrm{t}}(k)\boldsymbol{\upsilon}k) \tag{5.12}$$

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\boldsymbol{\upsilon}(k) \tag{5.13}$$

$$\boldsymbol{\delta}(k) = \mathbf{C}_{\mathrm{t}}(k)\mathbf{x}_{\mathrm{t}}(k) - \mathbf{C}(k)\mathbf{x}(k). \tag{5.14}$$

The vector $\mathbf{x}_{\mathrm{t}}(k)$ denotes the process state, and the model state $\mathbf{x}(k)$ contains the regression vectors $\mathbf{x}_l$ for the separate outputs in (2.2), reordered such that delayed outputs from all $\mathbf{x}_l$, $l = 1,\ldots,p$ come first and then the delayed inputs. Note that $\mathbf{x}(k)$ is identical to the one shown in Appendix D, without the last element of *one*.

The matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$ denote the process model (and at the same time the nominal process G), while the matrices $\mathbf{A}_t(k)$, $\mathbf{B}_t(k)$ and $\mathbf{C}_t(k)$ denote the real process G_t (Fig. 5.2), where

$$\begin{aligned}\mathbf{A}_t(k) &= \mathbf{A}(k) + \Delta\mathbf{A}(k)\\ \mathbf{B}_t(k) &= \mathbf{B}(k) + \Delta\mathbf{B}(k)\\ \mathbf{C}_t(k) &= \mathbf{C}(k).\end{aligned} \tag{5.15}$$

The matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$ are in the controllable canonical form. The model-plant mismatch is taken into account through the variations in the model parameters, given in $\Delta\mathbf{A}(k)$ and $\Delta\mathbf{B}(k)$. These matrices have structures identical to the ones used in $\mathbf{A}(k)$ and $\mathbf{B}(k)$ (without the ones and the θs in $\mathbf{A}(k)$), and the entries are the tripled values of the standard deviations of the corresponding parameters in the fuzzy model consequents. These parameters, together with their standard deviations, are obtained during the derivation of the fuzzy model. Since the matrix $\mathbf{C}$ (and $\mathbf{C}_t$) is constant, and only used to select the necessary outputs from the state vector, we drop the time index k for the sake of simplicity.

Equations (5.12) through (5.14) can be combined into

$$\begin{aligned}\mathbf{x}_c(k+1) &= \mathbf{A}_c(k)\mathbf{x}_c(k) + \mathbf{B}_c(k)\boldsymbol{\upsilon}(k)\\ \boldsymbol{\delta}(k) &= \mathbf{C}_c\mathbf{x}_c(k)\end{aligned} \tag{5.16}$$

with

$$\mathbf{x}_c(k) = \begin{pmatrix}\mathbf{x}_t(k)\\ \mathbf{x}(k)\end{pmatrix},\ \mathbf{A}_c(k) = \begin{pmatrix}\mathbf{A}_t(k) & 0\\ 0 & \mathbf{A}(k)\end{pmatrix},\ \mathbf{B}_c(k) = \begin{pmatrix}\mathbf{B}_t(k)\\ \mathbf{B}(k)\end{pmatrix},\ \mathbf{C}_c = (\mathbf{C}_t, -\mathbf{C}).$$

The output $\boldsymbol{\delta}(k)$ can be obtained through a Volterra series expansion

$$\begin{aligned}\boldsymbol{\delta}(k) = {}& \mathbf{C}_c\mathbf{B}_c(k-1)\boldsymbol{\upsilon}(k-1) + \mathbf{C}_c\mathbf{A}_c(k-1)\mathbf{B}_c(k-2)\boldsymbol{\upsilon}(k-2) +\\ & \mathbf{C}_c\mathbf{A}_c(k-1)\mathbf{A}_c(k-2)\mathbf{B}_c(k-3)\boldsymbol{\upsilon}(k-3) + \ldots\ .\end{aligned} \tag{5.17}$$

Comparing (5.17) with (5.10), we obtain the needed kernels $\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$

$$\mathbf{H}_{\delta,\mathrm{tv}}(k,i) = \begin{cases} i=0 & : \mathbf{H}_{\delta,\mathrm{tv}}(k,0) = 0\\ i=1 & : \mathbf{H}_{\delta,\mathrm{tv}}(k,1) = \mathbf{C}_c\mathbf{B}_c(k-1)\\ i=2 & : \mathbf{H}_{\delta,\mathrm{tv}}(k,2) = \mathbf{C}_c\mathbf{A}_c(k-1)\mathbf{B}_c(k-2)\\ \quad\vdots & \\ i=j & : \mathbf{H}_{\delta,\mathrm{tv}}(k,j) = \mathbf{C}_c\underbrace{\mathbf{A}_c(k-1)\ldots\mathbf{A}_c(k-j+1)}_{j-1}\mathbf{B}_c(k-j)\end{cases}. \tag{5.18}$$

Then, recall (5.11)

$$\begin{aligned}|\mathbf{H}_{\delta,\mathrm{tv}}(k,j)| &= |\mathbf{C}_c\mathbf{A}_c(k-1)\ldots\mathbf{A}_c(k-j+1)\mathbf{B}_c(k-j)|\\ &\le \|\mathbf{C}_c\mathbf{A}_c(k-1)\ldots\mathbf{A}_c(k-j+1)\mathbf{B}_c(k-j)\|\\ &\le \|\mathbf{C}_c\mathbf{D}^{-1}\mathbf{D}\mathbf{A}_c(k-1)\mathbf{D}^{-1}\mathbf{D}\ldots\mathbf{D}^{-1}\mathbf{D}\mathbf{A}_c(k-j+1)\mathbf{D}^{-1}\mathbf{D}\mathbf{B}_c(k-j)\|\end{aligned} \tag{5.19}$$

where $\|\cdot\| = \sigma_{max}(\cdot)$ (maximal singular value). In order to avoid $\|\mathbf{A}_c(k-i)\|$ greater or equal to one, $i = 1, \ldots, j-1$, which would imply instability, we introduce a Lyapunov transformation through a matrix $\mathbf{D}$, such that

$$\|\mathbf{D}\mathbf{A}_c(k-i)\mathbf{D}^{-1}\| < \alpha < 1. \tag{5.20}$$

Then

$$\begin{aligned} |\mathbf{H}_{\delta,tv}(k,j)| &\leq \|\mathbf{C}_c\mathbf{A}_c(k-1)\ldots\mathbf{A}_c(k-j+1)\mathbf{B}_c(k-j)\| \\ &\leq \|\mathbf{C}_c\mathbf{D}^{-1}\|\alpha^j\|\mathbf{D}\mathbf{B}_c(k-j)\|. \end{aligned} \tag{5.21}$$

To find an upper bound on the increment of the output $\Delta\boldsymbol{\delta}(k)$ we follow the same lines:

$$\|\Omega_\Delta\|_{i,\infty} = \max_{\Delta\boldsymbol{v}} \frac{\|\Delta\boldsymbol{\delta}(k)\|_\infty}{\|\Delta\mathbf{d}(k)\|_\infty}. \tag{5.22}$$

Analogously to (5.10) and (5.11)

$$\begin{aligned} \|\Delta\boldsymbol{\delta}(k)\|_\infty &= \max_k |\Delta\boldsymbol{\delta}(k)| \\ \Delta\boldsymbol{\delta}(k) &= \sum_{i=0}^{\infty} \Delta\mathbf{H}_{\delta,tv}(k,i)\Delta\boldsymbol{v}(k-i) \end{aligned} \tag{5.23}$$

and

$$\begin{aligned} \|\Delta\boldsymbol{\delta}(k)\|_\infty &= \max_k |\Delta\boldsymbol{\delta}(k)| \\ &\leq \max_k \sum_{i=0}^{\infty} |\Delta\mathbf{H}_{\delta,tv}(k,i)| \cdot |\Delta\boldsymbol{v}(k-i)| \\ &\leq \max_k \sum_{i=0}^{\infty} |\Delta\mathbf{H}_{\delta,tv}(k,i)| \\ &\leq \sum_{i=0}^{\infty} \max_k |\Delta\mathbf{H}_{\delta,tv}(k,i)| \leq \sum_{i=0}^{\infty} |\Delta\mathbf{H}_{\delta,tv,\max}(i)|. \end{aligned} \tag{5.24}$$

To find an expression for $\Delta\delta(k)$, we rewrite (5.16) in an incremental form

$$\begin{aligned} \Delta\delta(k) &= \delta(k) - \delta(k-1) = \mathbf{C}_c\big(\mathbf{x}_c(k) - \mathbf{x}_c(k-1)\big) \\ &= \mathbf{C}_c\Big\{\big[\mathbf{A}_c(k-1) - \mathbf{I}\big]\mathbf{x}_c(k-1) + \mathbf{B}_c(k-1)\boldsymbol{v}(k-1)\Big\} \\ &= \mathbf{C}_c\Big\{\big[\mathbf{A}_c(k-1) - \mathbf{I}\big]\big[\mathbf{A}_c(k-2)\mathbf{x}_c(k-2) \\ &\qquad + \mathbf{B}_c(k-2)\boldsymbol{v}(k-2)\big] + \mathbf{B}_c(k-1)\boldsymbol{v}(k-1)\Big\} \\ &= \cdots \quad . \end{aligned} \tag{5.25}$$

Thus

$$\begin{aligned} \Delta\boldsymbol{\delta}(k) &= \sum_{i=1}^{\infty} \boldsymbol{\Gamma}(k,i)\boldsymbol{v}(k-i) \\ &= \sum_{i=0}^{\infty} \Delta\mathbf{H}_{\delta,tv}(k,i)(k,i)\Delta\boldsymbol{v}(k-i), \end{aligned} \tag{5.26}$$

with

$$\Delta \mathbf{H}_{\delta,\mathrm{tv}}(k,i)(k,i) = \sum_{j=0}^{i} \mathbf{\Gamma}(k,j) \tag{5.27}$$

where the parameters $\mathbf{\Gamma}(k,i)$ are

$$\mathbf{\Gamma}(k,i) = \begin{cases} i=0: \mathbf{\Gamma}(k,0) = 0 \\ i=1: \mathbf{\Gamma}(k,1) = \mathbf{C}_\mathrm{c}\mathbf{B}_\mathrm{c}(k-1) \\ i=2: \mathbf{\Gamma}(k,2) = \mathbf{C}_\mathrm{c}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\mathbf{B}_\mathrm{c}(k-2) \\ \quad\vdots \\ i=l: \mathbf{\Gamma}(k,l) = \mathbf{C}_\mathrm{c}\Big[\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\underbrace{\mathbf{A}_\mathrm{c}(k-2)\ldots\mathbf{A}_\mathrm{c}(k-l+1)}_{l-2}\Big]\mathbf{B}_\mathrm{c}(k-l) \end{cases}.$$

Analogously to (5.19)

$$\begin{aligned} |\Delta \mathbf{H}_{\delta,\mathrm{tv}}(k,i)| &= \big|\mathbf{C}_\mathrm{c}\mathbf{B}_\mathrm{c}(k-1)+\cdots+\mathbf{C}_\mathrm{c}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\ldots\mathbf{A}_\mathrm{c}(k-i+1)\mathbf{B}_\mathrm{c}(k-i)\big| \\ &\le \big\|\mathbf{C}_\mathrm{c}\mathbf{B}_\mathrm{c}(k-1)+\cdots+\mathbf{C}_\mathrm{c}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\ldots\mathbf{A}_\mathrm{c}(k-i+1)\mathbf{B}_\mathrm{c}(k-i)\big\| \\ &\le \|\mathbf{C}_\mathrm{c}\mathbf{B}_\mathrm{c}(k-1)\|+\cdots+\big\|\mathbf{C}_\mathrm{c}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\ldots\mathbf{A}_\mathrm{c}(k-i+1)\mathbf{B}_\mathrm{c}(k-i)\big\| \\ &\le \|\mathbf{C}_\mathrm{c}\mathbf{B}_\mathrm{c}(k-1)\|+\cdots+ \\ &\quad \big\|\mathbf{C}_\mathrm{c}\mathbf{D}^{-1}\mathbf{D}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\mathbf{D}^{-1}\ldots\mathbf{D}\mathbf{A}_\mathrm{c}(k-i+1)\mathbf{D}^{-1}\mathbf{D}\mathbf{B}_\mathrm{c}(k-i)\big\| \end{aligned} \tag{5.28}$$

where $\|\cdot\| = \sigma_{max}(\cdot)$ is the maximal singular value and the matrix $\mathbf{D}$ is a Lyapunov transformation, such that $\|\mathbf{D}\mathbf{A}_\mathrm{c}(\cdot)\mathbf{D}^{-1}\| < \alpha < 1$ $\Big($and also $\big\|\mathbf{D}\big(\mathbf{A}_\mathrm{c}(k-1)-\mathbf{I}\big)\mathbf{D}^{-1}\big\| < \alpha < 1\Big)$. Then

$$|\Delta \mathbf{H}_{\delta,\mathrm{tv}}(k,i)| \le \sum_{j=1}^{i} \|\mathbf{C}_\mathrm{c}\mathbf{D}^{-1}\|\alpha^j\|\mathbf{D}\mathbf{B}_\mathrm{c}(k-j)\|. \tag{5.29}$$

5.4 Synthesis of a robust fuzzy model predictive controller

This section combines the results presented so far to design a predictive controller with guaranteed robust stability for stable TS models with ∞-norm bounded additive uncertainty. Both offset-free reference tracking and robust stability are guaranteed for asymptotically constant reference trajectories ($\mathbf{r}(k) \to \mathbf{r}_\infty$ as $k \to \infty$) and disturbances. The control system is said to be robustly stable if the process output (Fig. 5.2) goes to the constant value $\mathbf{r}_\infty$ when $\mathbf{u}_\mathrm{ff}$ and $\mathbf{u}_\mathrm{fb}$ become constant: $\mathbf{y}(k) = G(k)\big(\mathbf{u}(k)\big) + \mathbf{e}(k) \to \mathbf{r}_\infty$ as $k - k_0 \to \infty$ and $\Delta\mathbf{u}_\mathrm{ff} = \Delta\mathbf{u}_\mathrm{fb} = 0$, $\forall k \ge k_0$.

5.4.1 Tuning parameters for nominal and robust performance

The robust stability constraints calculated through **C1**, **C3** or **C4** use as parameters $\bar{\Omega}_\mathbf{e}$, $\bar{\Omega}_{\Delta_\mathbf{e}}$, $\bar{Q}$, $\bar{Q}_\Delta$ and the output $\mathbf{u}_\mathrm{ff}$ of the feedforward filter R. Below the effect of these "tuning parameters" of the constraints is explained and guidelines are given on how to select them.

In practice, besides robust stability and nominal performance, robust performance is also desired. For optimal nominal performance the constraints on $\mathbf{u}$ and $\Delta\mathbf{u}$ should be as large as possible, while still guaranteeing stability. For robust performance the upper bounds on δ and $\Delta\delta$ should be as small as possible. Although the MPC method usually determines the compromise between the nominal and robust performance, the robust stability constraints should also influence this compromise.

Nominal performance. When nominal performance is the main objective, **C4** should be used in order to provide sufficient freedom to the constraints on $\mathbf{u}$ (and $\Delta\mathbf{u}$), and $\bar{\Omega}_{\mathbf{e}}$, $\bar{\Omega}_{\Delta_{\mathbf{e}}}$, $\bar{Q}$ and $\bar{Q}_{\Delta}$ should be chosen as large as possible. This would result in good nominal disturbance rejection. The steady-state gain of the feedforward filter R should be the inverse of the steady-state gain of $G(k)$. Usually the choices proposed by de Vries and van den Boom (1997)

$$\begin{aligned} \bar{\Omega}_{\mathbf{e}}(k) &= \epsilon_{\Omega}(k)\frac{1-p}{1-pq^{-1}}I \qquad \text{and} \\ \bar{\Omega}_{\Delta_{\mathbf{e}}}(k) &= \epsilon_{\Omega_{\Delta}}(k)\frac{1-p}{1-pq^{-1}}I \qquad 0 \ll p < 1 \end{aligned} \tag{5.30}$$

ensure that $\bar{\Omega}_{\mathbf{e}}(k)\big(\mathbf{u}(k)\big)$ and $\bar{\Omega}_{\Delta_{\mathbf{e}}}(k)\big(\Delta\mathbf{u}(k)\big)$ are much larger than the true output (and the output increment) of the model uncertainty, even if the actual model uncertainty has a different realization. For the upper bounds of the controller mappings de Vries and van den Boom (1997) suggested

$$\begin{aligned} \bar{Q}(k) &= \frac{\lambda(k)}{\epsilon_{\Omega}(k)}\frac{1-p}{1-pq^{-1}}I \qquad \text{and} \\ \bar{Q}_{\Delta}(k) &= \frac{\lambda_{\Delta}(k)}{\epsilon_{\Omega_{\Delta}}(k)}\frac{1-p}{1-pq^{-1}}I \end{aligned} \tag{5.31}$$

where $\epsilon_{\Omega_{\Delta}}(k)\|G(k,1)^{-1}\|_{i,\infty} \leq \lambda_{\Delta}(k) < 1$ and $\epsilon_{\Omega}(k)\|G(k,1)^{-1}\|_{i,\infty} \leq \lambda(k) < 1$ and $G(k,1)$ is the steady-state gain matrix of the nominal system at time instant k. These choices also guarantee offset-free reference tracking.

Robust performance. When robust performance is required, the feedforward filter R can be designed in a way similar to LTI l_1-control or H_{∞} control. However, due to the LTV nature of the model, R has to be recomputed at each sampling instant reflecting the current situation. In this case the constraint **C1** can be used, where $\bar{Q}(k)$ and $\bar{Q}_{\Delta}(k)$ are computed according to (5.31). This would result in conservative constraints because **C1** only depends on $\epsilon_{\Omega}(k)$ and $\epsilon_{\Omega_{\Delta}}(k)$ and no other information about $\Omega(k)$ and $\Omega_{\Delta}(k)$ is taken into account. Such information could be used when **C3** is applied. For example, $\bar{\Omega}_{\mathbf{e}}(k)$ and $\bar{\Omega}_{\Delta_{\mathbf{e}}}(k)$ can be chosen such that $\bar{\Omega}_{\mathbf{e}}(k)\big(\mathbf{u}(k)\big) = |W(k)\mathbf{u}(k)|$ and $\bar{Q}(k)\big(\mathbf{e}(k)\big) = |\hat{Q}(k)V(k)\mathbf{e}(k)|$, where $V(k)$ and $W(k)$ are the weighting filters used in LTI H_{∞} control ($\|V^{-1}(k)\Omega(k)W^{-1}(k)\|_{i,\infty} \leq 1$), and $\hat{Q}(k)$ is the optimal H_{∞} controller.

5.4.2 Offset-free reference tracking and feedforward filter

Offset-free reference tracking. Although offset-free tracking should be guaranteed by the MPC method, we investigate the assumptions that the robust stability constraints must satisfy in order to make the problem (at least) feasible.

In a steady state, $\mathbf{u}_{\text{ff}}$ must be equal to $G(k,1)^{-1}\mathbf{r}_\infty$ and $\mathbf{u}$ to $G(k,1)^{-1}\big(\mathbf{r}_\infty - \mathbf{e}(k)\big)$, which leads to the well-known result that the steady gain of Q in (5.5) must be equal to $Q(k,1) = G(k,1)^{-1}$ (and $Q_\Delta(k,1) = G(k,1)^{-1}$). This implies that the steady-state gains of $\bar{Q}$ and $\bar{Q}_\Delta$ must be equal to $|G(k,1)^{-1}|$ in order to guarantee offset-free tracking. Hence, in a steady state the 1-norms of $\bar{Q}$ and $\bar{Q}_\Delta$ are equal to $\|G(k,1)^{-1}\|_{i,\infty}$. In the theorems given in Section 5.2, the condition that $\|\bar{Q}\|_{i,\infty} < 1/\epsilon_\Omega$ and $\|\bar{Q}_\Delta\|_{i,\infty} < 1/\epsilon_{\Omega_\Delta}$ must hold for each k and thus also in a steady state. The latter implies that the bound on the induced ∞-norm of the additive model uncertainty must be lower than a specific value, which depends on the steady-state gain matrix of the nominal system – $\epsilon_\Omega < \|G(k,1)^{-1}\|_{i,\infty}^{-1}$ and $\epsilon_{\Omega_\Delta} < \|G(k,1)^{-1}\|_{i,\infty}^{-1}$.

Feedforward filter. As the forward gain of the fuzzy model is not constant, the feedforward filter is recomputed at each sampling instant. Using the optimal control sequence computed in the previous sampling instant, the fuzzy model is simulated over the prediction horizon, and linear models are obtained at ith step in the prediction horizon, $i = 1,\ldots,H_{\text{p}}$. The filter gain is the inverse of the steady-state gain of these models. There are three options:

1) a single gain based on the linear model obtained for $i = 1$;
2) a single gain based on the entire set of linear models. For example, the minimal of the steady-state gains;
3) different gains for each step i, $i = 1,...,H_{\text{p}}$, based on the steady-state gain of the corresponding model.

While the first alternative is the simplest one, the third one leads to the most accurate calculation. The second alternative is a compromise between the two, providing a feasible solution.

The parameters $\lambda(k) = \lambda_\Delta(k)$ in (5.31) are set 10% higher than the product of the uncertainty bound and the inverse of the steady-state gain of the current model, as there is an absolute upper limit of 0.9

$$\lambda(k)\,(=\lambda_\Delta(k)) = \min\big(1.1\epsilon_\Omega(k)\|G(k,1)^{-1}\|_{i,\infty}, 0.9\big).$$

5.5 Examples

Two examples are given to illustrate the method: liquid-level control in a simulated SISO cascaded-tanks process and real-time control of the liquid level in the MIMO laboratory-scale cascaded-tanks setup described in Chapter 3.

5.5.1 Liquid-level control in a simulated SISO cascaded-tanks setup

This example illustrates the computation of the uncertainty bounds, the different degree of conservatism of the three implementable constraints **C1**, **C3** and **C4**, and the

influence of the constraints on the performance of the controller. Consider a laboratory setup consisting of two cascaded tanks (Fig. 5.4). The manipulated variable is the pump flow rate q and the goal is to control the liquid level in the lower tank h_1 such that it follows a prescribed reference.

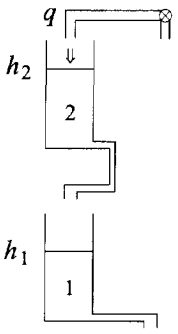


Figure 5.4. SISO cascaded-tanks setup.

The fuzzy model was identified using data from the real process, sampled with the period $T_s = 5\,\mathrm{s}$. The model consists of four rules of the form:

$\mathcal{R}_i$: **If** $h_1(k-1)$ is $\mathcal{A}_{i1}$ **and** $h_1(k-2)$ is $\mathcal{A}_{i2}$ **and** $q(k-1)$ is $\mathcal{A}_{i3}$ **and** $q(k-2)$ is $\mathcal{A}_{i4}$
then $h_1(k) = a_{i1}h_1(k-1) + a_{i2}h_1(k-2) + b_{i1}q(k-1) + b_{i2}q(k-2) + \theta_i$
$i = 1,\ldots,4$.

The identification technique is based on fuzzy clustering and least-squares estimation (Babuška, 1998), see Appendix A. The consequent parameters and their standard variations are given in Tab. 5.1 and Tab. 5.2 on the following page.

Table 5.1. Consequent parameters.

$\mathcal{R}_i$	$a_{i,1}$	$a_{i,2}$	$b_{i,1}$	$b_{i,2}$	θ_i
1	1.52	−0.598	$3.87 \cdot 10^{-4}$	$2.46 \cdot 10^{-2}$	$3.34 \cdot 10^{-3}$
2	1.68	−0.716	$-3.16 \cdot 10^{-3}$	$6.03 \cdot 10^{-2}$	$-1.39 \cdot 10^{-2}$
3	1.66	−0.688	$-1.37 \cdot 10^{-3}$	$3.52 \cdot 10^{-2}$	$-6.57 \cdot 10^{-3}$
4	1.74	−0.779	$-3.92 \cdot 10^{-3}$	$2.91 \cdot 10^{-2}$	$-6.15 \cdot 10^{-3}$

The performance of the model on a validation data set is shown in Fig. 5.5 on the next page. The *Variance Accounted For* performance index is VAF = 90.4%.

This fuzzy model is used to represent the process model in the controller. The process itself is simulated by using a perturbed fuzzy model obtained by randomly varying the consequent parameters of the original fuzzy model. The perturbed parameters are in the interval $\pm 3\sigma$ (Tab. 5.3).

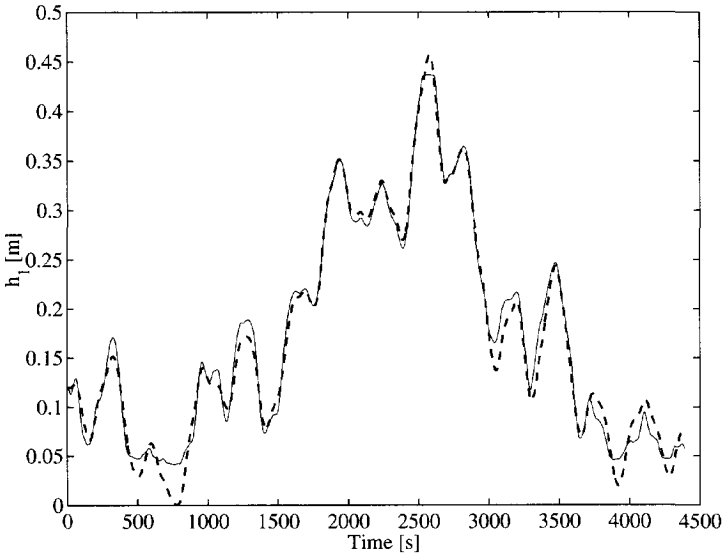


Figure 5.5. SISO cascaded-tanks setup. Validation of the fuzzy model. Solid line: process output, dashed line: model prediction.

Table 5.2. Standard deviations of the consequent parameters.

$\mathcal{R}_i$	$\sigma_{a_{i,1}}$	$\sigma_{a_{i,2}}$	$\sigma_{b_{i,1}}$	$\sigma_{b_{i,2}}$	σ_{θ_i}
1	$3.47 \cdot 10^{-2}$	$2.86 \cdot 10^{-2}$	$4.14 \cdot 10^{-3}$	$4.80 \cdot 10^{-3}$	$4.56 \cdot 10^{-4}$
2	$1.33 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$3.87 \cdot 10^{-3}$	$4.73 \cdot 10^{-3}$	$7.43 \cdot 10^{-4}$
3	$1.46 \cdot 10^{-2}$	$1.37 \cdot 10^{-1}$	$2.97 \cdot 10^{-3}$	$3.57 \cdot 10^{-3}$	$5.95 \cdot 10^{-4}$
4	$1.55 \cdot 10^{-2}$	$1.48 \cdot 10^{-1}$	$3.97 \cdot 10^{-3}$	$4.57 \cdot 10^{-3}$	$7.47 \cdot 10^{-4}$

Table 5.3. Perturbed consequent parameters.

rule	a_1	a_2	b_1	b_2	θ
1	1.51	−0.594	$1.51 \cdot 10^{-4}$	$2.68 \cdot 10^{-2}$	$3.84 \cdot 10^{-3}$
2	1.68	−0.714	$-2.39 \cdot 10^{-3}$	$6.64 \cdot 10^{-2}$	$-1.35 \cdot 10^{-2}$
3	1.67	−0.686	$-1.33 \cdot 10^{-3}$	$3.68 \cdot 10^{-2}$	$-6.45 \cdot 10^{-3}$
4	1.73	−0.750	$-9.61 \cdot 10^{-4}$	$3.10 \cdot 10^{-2}$	$-5.65 \cdot 10^{-3}$

The MPC parameters are selected according to the tuning rules given in Section 5.1. Since there is a delay of one sample, the minimum cost horizon $H_{\text{min}} = 1$. The process is well damped and a prediction horizon of $H_{\text{p}} = 4$ is used to speed up the response

(the settling time is about 25 s). The process order is two, thus control horizon is $H_c = 2$. The weight in the cost function is $P = 1$ and the physical input constraints are $q \in [0,1]$, $\Delta q \in [-1,1]$. The gain of the feedforward filter R (Fig. 5.2) is equal to the inverse of the minimal steady-state gain (2nd option in Section 5.4.2). The parameter p in (5.30) and (5.31) is set to $p = 0.5$, which gives a first-order filter $\frac{0.5}{z-0.5}$. The choice of p determines how the stability constraints will react to a change either in the model gain (λ and λ_Δ) or in the model uncertainty bounds (ϵ_Ω and ϵ_{Ω_Δ}): $p \ll 1$ allows rapid changes for a short period, while with $p \approx 1$ the transient is long with a small amplitude.

The benefits of the proposed method can be appreciated when we compare the result to unconstrained case. Figure 5.6 and Figure 5.7 on the following page demonstrate the difference between controllers with and without stability constraints. Without stability constraints (only the physical constraints $q \in [0,1]$, $\Delta q \in [-1,1]$ are imposed) the system oscillates. The robust constraints smooth out the control signal, which results in a smoother output. Note that offset-free reference tracking is achieved.

The calculation of the uncertainty bounds at time $t = 51$ s is shown below. The bounds $\epsilon_\Omega(k)$ and $\epsilon_{\Omega_\Delta}(k)$ on the model uncertainty $\bar{\Omega}$ and $\bar{\Omega}_\Delta$ are computed according to (5.9) and (5.22), respectively. For the sake of illustration, the considered convolution operators in (5.10) and (5.23) are of fifth order.

At $t = 51$ s, i.e., at $k = 10$ the previous values for the lower tank level are $h_1(k) = 0.1268$ m and $h_1(k-1) = 0.0955$ m; the control input is $q(k) = 0.6066$ and $q(k-1) = 0.6357$. Based on these values, the matrices $\mathbf{A}_c(k)$ and $\mathbf{B}_c(k)$ extracted from the fuzzy model (5.16) are

$$\mathbf{A}_c(k) = \begin{pmatrix} 1.661 & -0.689 & 0.035 & 0 & 0 & 0 \\ 1.000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.706 & -0.648 & 0.046 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{B}_c(k) = \begin{pmatrix} -0.0014 \\ 0 \\ 1.0000 \\ 0.0076 \\ 0 \\ 1.0000 \end{pmatrix}$$

The corresponding matrices obtained in the previous four sampling instants are $\mathbf{A}_c(k-2) = \mathbf{A}_c(k-1) = \mathbf{A}_c(k)$, $\mathbf{B}_c(k-2) = \mathbf{B}_c(k-1) = \mathbf{B}_c(k)$ and

$$\mathbf{A}_c(k-3) = \begin{pmatrix} 1.528 & -0.608 & 0.026 & 0 & 0 & 0 \\ 1.000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.626 & -0.526 & 0.040 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{B}_c(k-3) = \begin{pmatrix} 0.0002 \\ 0 \\ 1.0000 \\ 0.0123 \\ 0 \\ 1.0000 \end{pmatrix}$$

$$\mathbf{A}_c(k-4) = \begin{pmatrix} 1.513 & -0.598 & 0.026 & 0 & 0 & 0 \\ 1.000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.617 & -0.513 & 0.039 \\ 0 & 0 & 0 & 1.000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \mathbf{B}_c(k-4) = \begin{pmatrix} 0.0004 \\ 0 \\ 1.0000 \\ 0.0128 \\ 0 \\ 1.0000 \end{pmatrix}$$

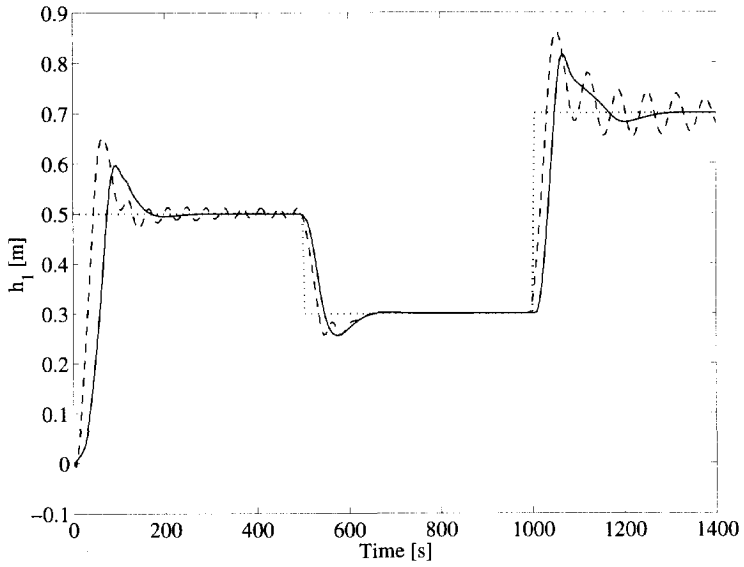


Figure 5.6. SISO cascaded-tanks setup. Reference tracking with (solid) and without (dashed) robust constraints.

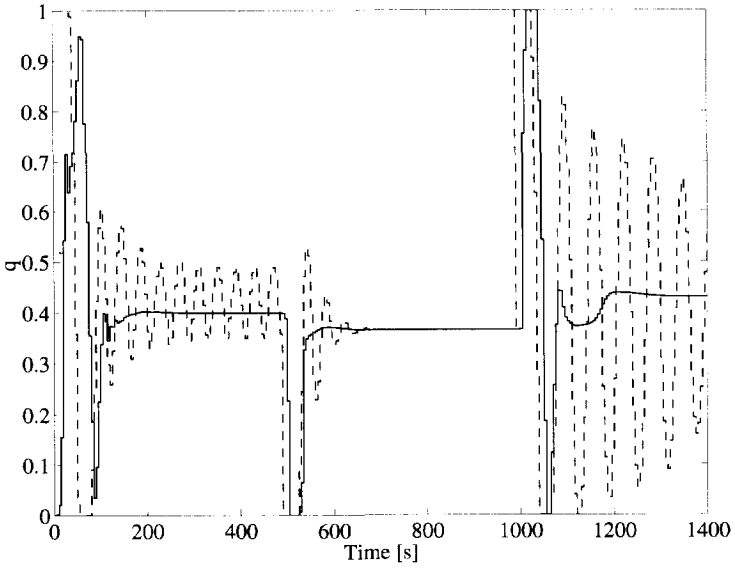


Figure 5.7. SISO cascaded-tanks setup. Control signal with **C1** (solid line) and without robust stability constraints (dashed line).

and the common matrix is $\mathbf{C}_c = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$.

For the kernels $\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$, $i = 0,\ldots,5$ from (5.19) we have

$$|\mathbf{H}_{\delta,\mathrm{tv}}(k,i)| = \begin{cases} i=0 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,0)| = 0 \\ i=1 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,1)| \leq 0.0090 \\ i=2 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,2)| \leq 0.0260 \\ i=3 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,3)| \leq 0.0399 \\ i=4 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,4)| \leq 0.0643 \\ i=5 & : & |\mathbf{H}_{\delta,\mathrm{tv}}(k,5)| \leq 0.0880 \end{cases} \tag{5.32}$$

and using (5.9) through (5.11), $\epsilon_\Omega(k) = 0.2272$.

Analogously for the incremental kernels $\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$, $i = 0,\ldots,5$ from (5.28) we have

$$|\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,i)| = \begin{cases} i=0 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,0)| = 0 \\ i=1 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,1)| \leq 0.0090 \\ i=2 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,2)| \leq 0.0260 \\ i=3 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,3)| \leq 0.0229 \\ i=4 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,4)| \leq 0.0261 \\ i=5 & : & |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,5)| \leq 0.0283 \end{cases} \tag{5.33}$$

and using (5.9) through (5.11), the upper bound of the uncertainty is $\epsilon_{\Omega_\Delta}(k) = 0.1123$. The robust stability constraints on Δq for the first 450 s, computed through **C1**, **C3** and **C4** are given in Fig. 5.8 on the next page. The input increment is guaranteed to stay in the interval determined by the bounds $\Delta q_{\min}$ and $\Delta q^{\max}$ respectively (see Fig. 5.9 for the **C1** case). Since $\Delta q_{\min}$ and $\Delta q^{\max}$ are symmetric around zero (Theorem 5.4 and Corollary 5.1), only $\Delta q^{\max}$ is shown in the figure. Note that the constraint based on **C1** is the most conservative one, while the one using **C4** is the least conservative. The **C3** constraint is in the middle. After a reference change, the constraints based on **C1** swiftly go to zero, while the **C4** constraints allow some variation of the control signal. The **C3** constraints also tend to zero, but more slowly than the constraints based on **C1**.

5.5.2 Real-time liquid-level control in a MIMO cascaded-tanks setup

The real-time performance of the presented technique is demonstrated by using the setup described in Section 3.4. Again the control objective is to follow set-point changes in the levels in the lower two tanks by adjusting the flow rates of the liquid entering the upper tanks.

The MPC parameters are selected as follows. The minimum cost horizon is $H_{\min} = 1$ as in the previous section. The control horizon is set to $H_c = 2$ and the prediction horizon is set only to $H_p = 4$ to improve the disturbance rejection properties. The non-zero weighting matrix is $\mathbf{P} = \mathbf{I}$. The user-provided constraints on the inputs and outputs are

$$\begin{array}{llll} q_1 \in [0,\ 1] & q_2 \in [0,\ 1] & \Delta q_1 \in [-0.3,\ 0.3] & \Delta q_2 \in [-0.3,\ 0.3] \\ h_1 \in [0,\ 0.5] & h_2 \in [0,\ 0.5] & \Delta h_1 \in [-0.1,\ 0.1] & \Delta h_2 \in [-0.1,\ 0.1] \end{array}$$

The process behaviour with and without robust stability constraints (**C1**) is presented in Fig. 5.10 on page 106. The bounds on the control signal and its increment

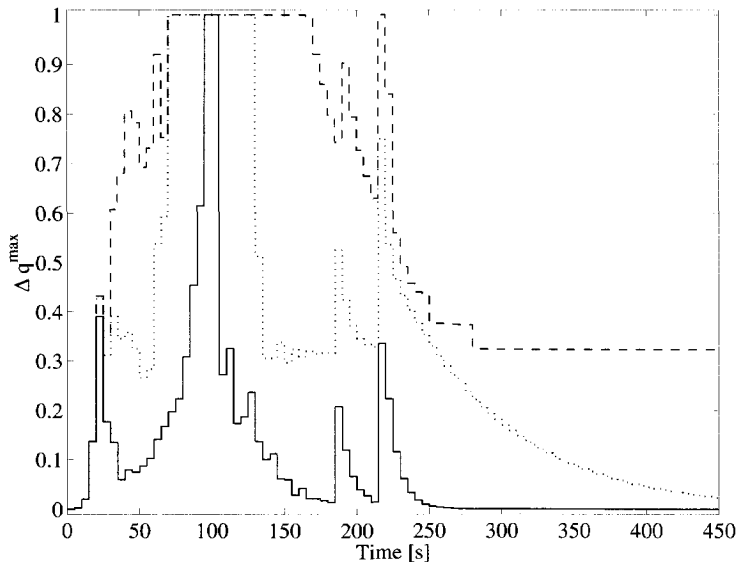


Figure 5.8. SISO cascaded-tanks setup. Robust stability constraints **C1** (solid line), **C3** (dotted line) and **C4** (dashed line) on the control input increment Δq.

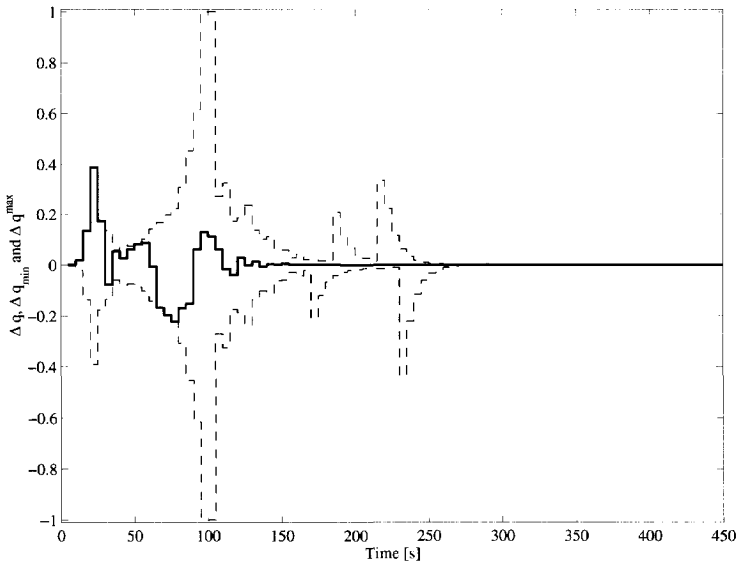


Figure 5.9. SISO cascaded-tanks setup. Control input increment Δq (solid) and the robust stability constraints based on **C1** (dashed).

are not very tight, due to the existing interactions inside the model (Fig. 5.11). Still, the constraints on the control increment go to zero in steady state and when no disturbances are present. Note that since the user-provided lower bounds on the control signals are equal to zero, only the upper bounds are given in Fig. 5.11a.

While in both cases the controller achieves offset-free reference tracking, the introduction of the robust stability constraints reduces the oscillations during a transition between different set-point levels. At $T = 200$ s, a temporary power failure (lasting five seconds) in the left pump is deliberately introduced. Again, the robust constraints smooth out the controller reaction to this external disturbance.

5.6 Summary and concluding remarks

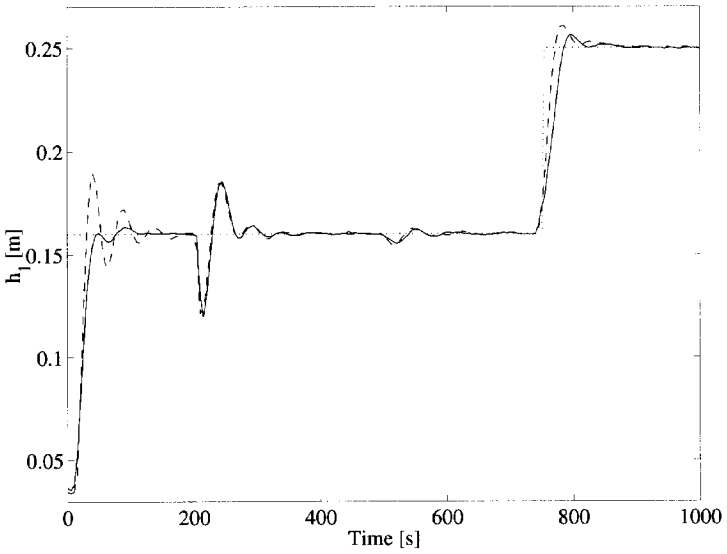
A general scheme for computing robust stability constraints for nonlinear MPC has been presented. Constraints on the control signal and its increment are calculated that guarantee stability for any model-plant mismatch within the given uncertainty for general (also non-fuzzy) nonlinear plants. The underlying assumptions are as follows: (i) the available model perfectly matches the nominal process, (ii) the model-plant mismatch is expressed as an additive uncertainty, and (iii) the offset is bounded.

The stability constraints can be derived using only the small-gain theorem, however, they may be too conservative. The conservatism is reduced by taking into account the model-plant mismatch. The process model is considered to be a linear time-varying model, rather than a nonlinear time-invariant one, which makes possible the use of linear input-output operators. The model-plant mismatch is expressed as an output of such a linear time-varying operator. The stability constraints are recomputed at each sampling instant, based on the current model-plant mismatch.

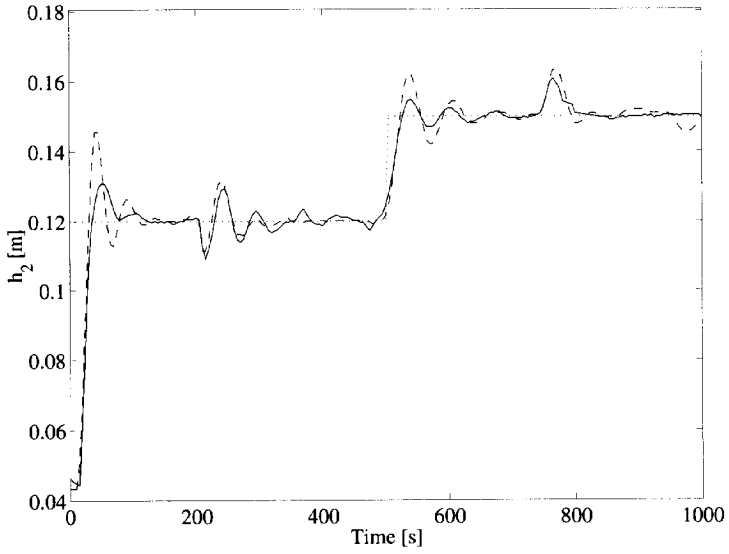
A systematic procedure for the computation of the bounds on the model uncertainty has been proposed for a the process model that is a TS fuzzy model. The TS model is assumed to match the nominal process perfectly and the model-plant mismatch is given through the standard deviations of the fuzzy consequent parameters. To compute the uncertainty bounds, the model-plant mismatch is represented as a linear time-varying convolution operator. The parameters (kernels) of the convolution operator are expressed through the difference between the outputs of the nominal and the perturbed process. Since here the incremental and non-incremental input-output mappings are different, separate convolution operators for the incremental and non-incremental relations are needed.

A simulated SISO cascaded-tanks process was used to illustrate the computation of the uncertainty bounds, the different degrees of conservatism of the different implementable constraints **C1**, **C3** and **C4**, and the influence of the constraints on the performance of the controller. To attain an offset-free reference tracking also in the most conservative (**C1**) case, the feedforward filter has to be properly designed in order to provide extra freedom. The filter gain is recomputed at each sampling instant, using the current fuzzy model gain. The effectiveness of the approach in real time was demonstrated using a MIMO laboratory cascaded-tanks setup.

It was shown that the stability constraints robustify the system performance in the case of a model-plant mismatch without deteriorating the nominal performance. The

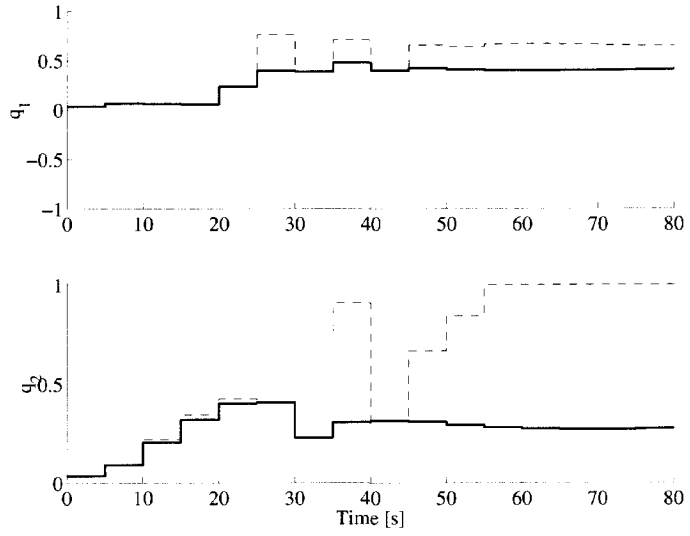


(a) Lower left tank level.

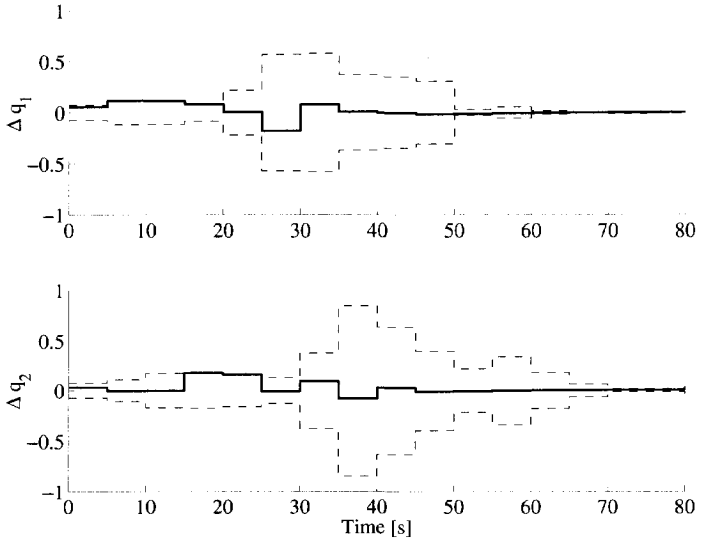


(b) Lower right tank level.

Figure 5.10. MIMO cascaded-tanks setup. Real-time control performance with and without robust stability constraints.



(a) Control signals and upper constraints. Solid line: control signal, dashed line: stability constraints.



(b) Control signal increment and stability constraints. Solid line: control signal, dashed line: stability constraints.

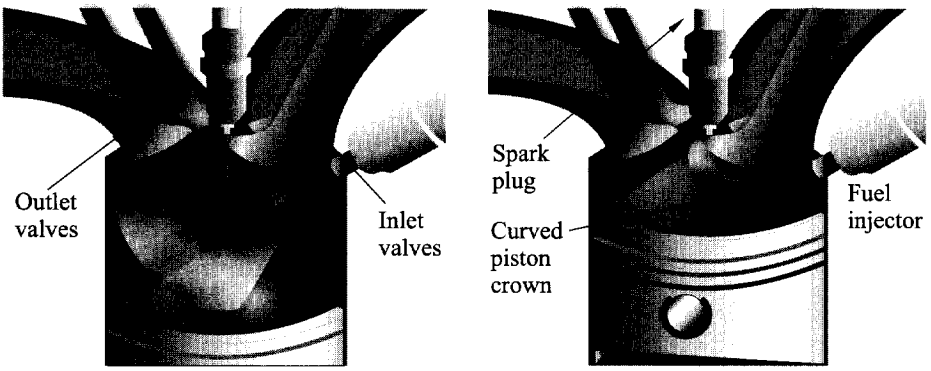
Figure 5.11. MIMO cascaded-tanks setup. Robust stability constraints on the control signals and their increments.

stability constraints, rather than the weights in the cost function smooth out the control signal. An additional advantage is that the stability constraints and the bounds on the model uncertainty are computed in a systematic way, by extracting the necessary information from the fuzzy model, either off-line (where possible) or on-line, at each sampling instant.

The application of the stability constraints, however, has a significant drawback: its complexity and, as a consequence, the computational time. In order to compute the bounds on the model-plant mismatch, we represent the fuzzy model as a convolution operator. The order of the convolution operator depends on the poles of subsystems in the fuzzy model. As a result, when the poles move closer to the unit circle, this order increases, and hence the memory requirements and the processing time as well.

6 FUZZY MODEL PREDICTIVE CONTROL OF A GDI ENGINE

This chapter presents the first of the two real-world applications of the developed methods for fuzzy model predictive control – a *Gasoline Direct Injection* engine. The complexity of this system exceeds that of most of the previously reported applications of MPC, mainly because of the switching of combustion modes and the related adaptation of the cost function and constraints. Section 6.1 introduces the physical and technological principles of this engine. The GDI simulation model, developed by Siemens Automotive SA in Toulouse, is given in Section 6.2. This model is implemented in Simulink and includes the engine management system and the driver and powertrain submodels. Section 6.3 describes the construction of the fuzzy prediction models for MPC. The control design is presented in Section 6.4. The obtained results are presented in Section 6.5. Comparison with other control strategies is given in Section 6.6.



(a) Homogeneous mode *(b) Stratified mode*

Figure 6.1. Overview of the differences between the HM and SM.

6.1 GDI engine

The technology currently used in most modern gasoline combustion engines is indirect multiple point injection (MPI), in which the fuel is injected before the inlet valve outside the cylinder (Ganesan, 1994). MPI, however, shows limitations with respect to the new, more stringent requirements for reduced pollution and fuel consumption.

For years, engineers have known that if they could build a petrol engine that works like a diesel engine – one in which fuel is injected directly into the cylinder and the stratified, rich mixture near the spark plug is ignited – they would have an engine that achieved both the fuel efficiency of a diesel engine and attained the high output of a conventional petrol engine. However, development of such an engine has been impeded by the poor combustibility of petrol.

The *Gasoline Direct Injection* (GDI) engine is a new type of engine which can operate in two different combustion modes: homogeneous and stratified (Fig. 6.1). In the homogeneous mode (HM), the operation of the engine resembles that of an MPI engine. The fuel injection takes place early in the inlet stroke (Fig. 6.2) and the air-fuel mixture is spread evenly throughout the cylinder (Fig. 6.1a). In the stratified mode (SM), the injection takes place late in the compression stroke. A compact spray of fuel is injected through a high-pressure swirl injector over the piston crown, resulting in an optimal stratified air/fuel mixture immediately beneath the spark plug (Fig. 6.1b). The movement of the fuel spray, the piston head's deflection of the spray and the flow of the air within the cylinder cause the spray to vaporize and disperse. The resulting mixture of gaseous fuel and air is then carried up to the spark plug for ignition. The biggest advantage of this system is that it enables precise control (sampling time $T_s = 5$ ms) over the air/fuel ratio at the spark plug at the point of ignition. In this way, higher efficiency can be achieved, which leads to fuel savings up to 30% and lower pollution (Bortolet et al., 1998).

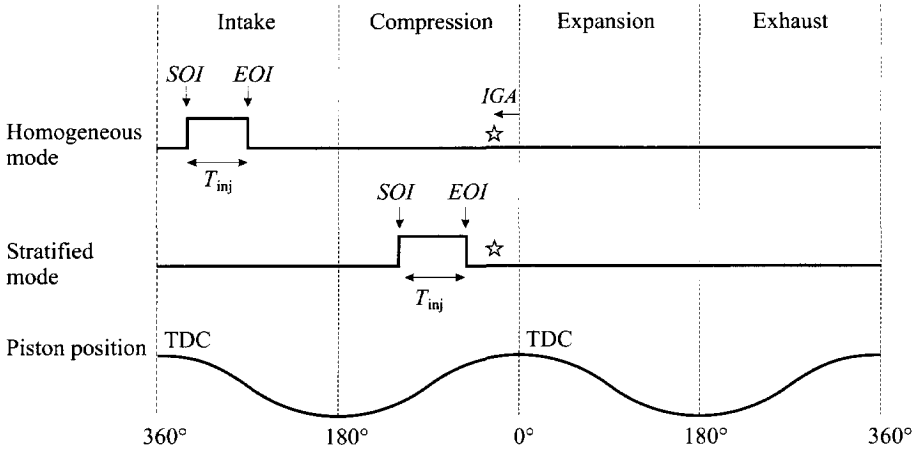


Figure 6.2. Timing in the two different combustion modes. The Start Of Injection (SOI), the injection time (T_{inj}) and the IGnition Advance angle (IGA) are indicated. TDC and EOI stand for Top Dead Center and End Of Injection, respectively. The moment of spark flash is denoted by a star (☆).

Figure 6.2 shows the relation between engine timing and the main control variables: the Start Of Injection (SOI) and IGnition Advance angle (IGA) are expressed in degrees before Top Dead Cgenter (TDC). The time of injection T_{inj} is expressed in milliseconds and can be recalculated to degrees by scaling with a factor of $0.006 \cdot N$ [deg/ms]. The main difference between the homogeneous and stratified mode is the fuel injection timing. In the HM, the injection takes place in the intake stroke (EOI $> 180°$), while in the SM the fuel is injected during the compression stroke (EOI $< 180°$).

In response to driving conditions, the Engine Management System (EMS) changes the timing of the fuel injection, alternating between the combustion modes: stratified charge and homogeneous charge. Under normal driving conditions, when the speed is stable and there is no need for sudden acceleration, the GDI engine operates in the stratified mode (SM). The SM, however, can only be used in a restricted operating range (in terms of engine speed and torque demand), and operation in the homogeneous mode (HM) is necessary in certain situations, e.g., to provide fast acceleration. The GDI engine switches to the HM when the driver accelerates, i.e., when more power is needed.

The task of the EMS is to control the combustion process such that the torque demand is satisfied, while fuel consumption is minimized and the required driving comfort is maintained (switching between the combustion modes can produce excessive torque gradients and thus negatively influence the comfort). The choice of the combustion mode has a large impact on the timing in the cylinder, and hence, on the control strategy used.

The EMS controllers are normally designed using engine maps: look-up tables derived through extensive experiments with an engine prototype (Gäfvert et al., 2000).

This is a laborious and time-consuming approach. Instead, here a model predictive controller (MPC) is utilized, based on TS fuzzy models identified from input-output data. The application of MPC to engine control is a new approach, which can potentially surpass the shortcomings of the traditional control strategies applied to increasingly sophisticated engines (Mollov, van der Veen and Babuška, 2001). MPC is truly multi-variable and can handle constraints in an explicit way. No detailed process knowledge is required to construct the fuzzy models. Instead, relatively simple closed-loop experiments are run on the engine to collect data (Mollov, Babuška and van der Veen, 2001).

The MPC determines optimal settings for the Start Of Injection SOI, the duration of fuel injection T_{inj}, the IGnition Advance angle IGA and the amount of fresh air introduced into the cylinders MTC (Fig. 6.3). The important state and output variables of the engine (such as the engine speed, effective torque or the air/fuel ratio) are predicted by fuzzy models. Different models are used in the different combustion modes. The optimization algorithm is based on quadratic programming, using successive linearization of the fuzzy prediction model (see Chapter 4). Due to the short sampling period ($T_s = 5$ ms), only a single linear model is extracted from the fuzzy one at the current operating point. Mode-dependent constraints and weights in the cost function are introduced in order to minimize the torque bumps during mode switching.

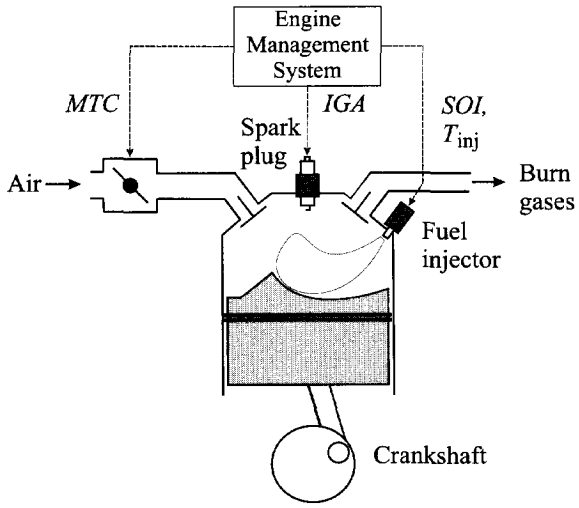


Figure 6.3. A schematic diagram of one cylinder of the GDI engine with the main control variables indicated.

6.2 The GDI simulation model

A dynamic simulation model of the GDI engine was developed by Siemens Automotive SA in Toulouse within the European Esprit research project FAMIMO (Fuzzy Algorithms for the control of MIMO Processes, LTR 219 11). The overall structure of

the simulation model is shown in Fig. 6.4. It consists of three major subsystems: the EMS and the driver and powertrain models.

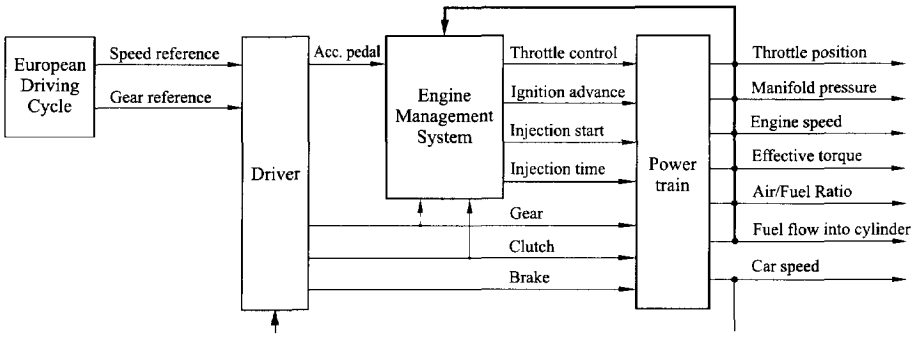


Figure 6.4. The simulation model including the powertrain submodel, the EMS and a submodel of the driver.

The reference inputs are the speed and gear references defined by a driving scenario, extracted from the European Driving Cycle (Commission, EU, 1993). The driver tries to follow these references as closely as possible by operating the acceleration, clutch and brake pedals and the gearshift. Based on the driver commands and the engine states and outputs, the *EMS* provides the four control inputs for the engine. These signals are given in Fig. 6.4 and the corresponding symbols and units are summarized in Table 6.1.

Table 6.1. Control inputs and measured outputs of the engine.

Control inputs	Symbol	Unit
Throttle command	MTC	% open
Ignition advance angle	IGA	deg before TDC
Start of injection	SOI	deg before TDC
Injection time	T_{inj}	ms
Outputs	Symbol	Unit
Throttle position	ϕ_{MTC}	% open
Manifold pressure	P_m	mbar
Engine speed	N	rpm
Effective torque	TQE	Nm
Air/fuel ratio	RAF	—
Fuel flow into cylinder	Q_{fuel}	kg/h
Car speed	V	km/h

The goal is to design an EMS that meets the following control objectives:

1. Control the engine such that the effective torque follows the torque reference. This is necessary in order to allow the car speed to follow the reference signal while maintaining driving comfort (avoiding excessive torque gradients).

2. Operation in the SM requires less fuel and consequently reduces the pollution. However, there are limitations to the range of operation in the SM which are defined with respect to the engine speed and the indicated torque TQI (i.e., the effective torque minus torque losses, TQI = TQE − TQL)

$$\text{TQI} < 50\,\text{Nm} \quad \text{and} \quad \text{N} < 3000\,\text{rpm}. \tag{6.1}$$

   Since the SM operation is restricted, the engine should be able to switch between the combustion modes according to the operating conditions. The commutation between the HM and the SM must be such that any disturbance to the driver is avoided.

3. During HM operation, keep the air/fuel ratio to a constant value RAF = 1 with an allowed tolerance of 2%. During SM operation, the engine must run in lean burn conditions, i.e., 1 < RAF.

4. Minimize the fuel consumption. Due to the low fuel consumption and low pollutant emissions in the SM, the engine should operate whenever possible in this mode.

The *powertrain model* describes the behaviour of the car. Its dynamics are given through three state variables: the fresh-air throttle position ϕ_{MTC}, the pressure of the intake manifold P_m and the engine speed N, and through a number of nonlinear maps. The powertrain model can be divided in two parts: *engine model* and *engine load model* (Fig. 6.5). The engine model contains two of the states: ϕ_{MTC} and P_m, while N is hosted by the engine load model (Sun et al., 1999).

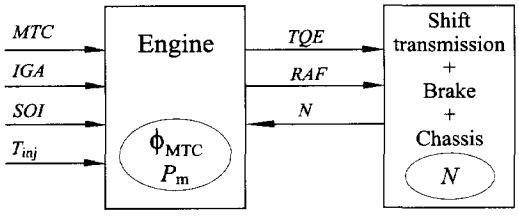


Figure 6.5. The powertrain model. The engine delivers the torque TQE to the transmission, which determines the engine speed N on the basis of the load and the gearshift.

The engine load model includes the transmission, the brakes and the chassis. It describes the car's behaviour for a given gearshift and is assumed to be invariant during the simulations.

The engine model can be decomposed into three subsystems: the air intake, the engine manifold, and the cylinder (Fig. 6.6). The volume of intake air is manipulated through the position of the fresh-air throttle ϕ_{MTC}, with control input MTC. The air intake delivers the air with flow rate Q_{tps} to the engine manifold, affecting the manifold

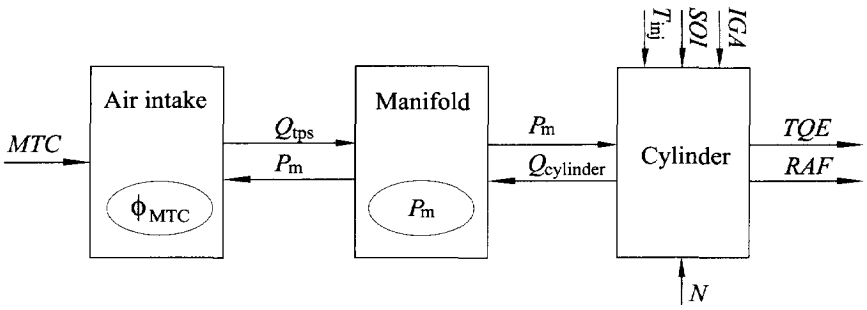


Figure 6.6. The three parts of the engine model: the air intake, the intake manifold and the cylinder.

pressure P_m. The manifold pressure and the engine speed N control the air intake in the cylinder $Q_{cylinder}$, while P_m is also influenced by $Q_{cylinder}$. The inputs T_{inj} and SOI control the fuel intake and the injection timing, respectively (Fig. 6.2). The cylinder subsystem contains no dynamics, hence the effective torque TQE and the air/fuel ratio RAF are determined directly by the inputs IGA, SOI, T_{inj}, P_m and N.

In practice, not all signals inside the engine (states and outputs) can be directly measured with sensors (e.g., the air/fuel ratio). Nevertheless, because observers can be applied to reconstruct the non-measurable signals, it is assumed here that all signals are available.

The approach chosen in the FAMIMO project is to consider the engine benchmark as a "gray box." Therefore the engine model (mathematical equations, look-up tables) cannot be utilized directly for control design. The model can only be used to gain some basic knowledge regarding engine control (e.g., "In the HM, the engine torque depends mainly on the air introduced in the cylinder and can be reduced through the ignition advance and the air/fuel ratio"). Nevertheless, for control purposes some look-up tables can directly be used to *express the various constraints* on the signals and parameters of the engine, such as (Section 6.4.1 – Section 6.4.2):

1. The SM operation domain;

2. IGA constraints both in the HM and in the SM;

3. Start and end of injection (SOI and EOI).

6.3 Construction of the prediction model for MPC

The development of a physical (white-box) model of the powertrain for control purposes is a difficult and time-consuming task. An alternative approach is to build a TS fuzzy model based on measured data.

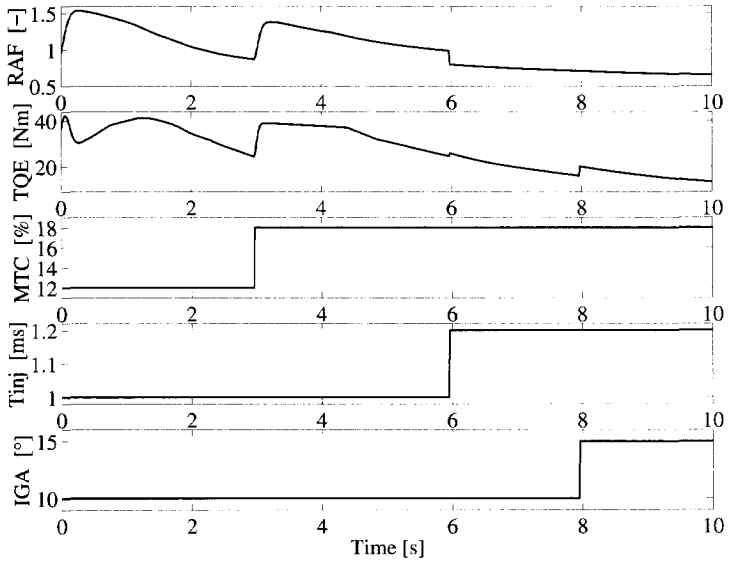


Figure 6.7. Relations between the RAF and TQE, and inputs MTC, T_{inj} and IGA in the HM.

6.3.1 Generation of experimental data

Many of the nonlinearities in the engine are related to the engine speed N, hence to obtain a good model one should cover the complete range of N. The initial experiments indicated that a random distribution of the inputs may drive N out of the operation range [900, 3700] rpm, therefore some type of control for N is needed.

The engine speed is an integral of the effective torque, so through TQE one can keep the N in the desired range. Figure 6.7 gives an impression of the influence of the inputs on TQE and RAF in the HM during the first ten seconds after ignition. An increase in the MTC results in a higher throttle opening ϕ_{MTC}. As a result the air flow increases, hence also RAF. The air flow influences the manifold pressure P_m. Together RAF and P_m influence the torque TQE. The injection time T_{inj} controls the fuel flow Q_{fuel}, and thus influences RAF inversely. The IGnition Advance angle IGA determines the spark flash moment, hence affects only TQE.

Proportional and PI controllers have been devised to provide the torque in the different combustion modes, so that the engine speed is well spread over the operational range [900, 3700] rpm. In the HM, this is accomplished by a cascaded manipulation of MTC, T_{inj} and IGA, keeping RAF approximately one. In the SM, only the T_{inj} and MTC inputs are used and RAF is maintained above one. In both modes MTC is proportional to the difference between the current and desired engine speed. Based on the MTC value, T_{inj} is adjusted proportionally to the difference between the current and desired engine speed. The ignition advance angle IGA is determined on the basis of T_{inj} and Q_{fuel}. Random variations are applied to each of these control variables

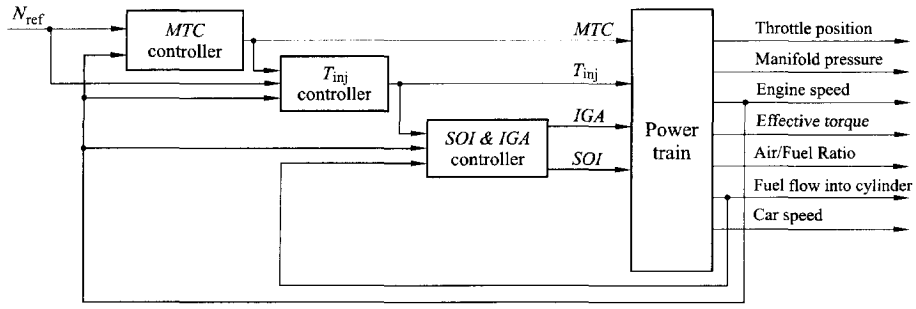


Figure 6.8. The closed-loop identification scheme.

in order to achieve proper excitation of the system. In the homogeneous mode, the start of injection SOI is a constant, and in the stratified mode, SOI and IGA are predescribed through a look-up table, thus they are not subject to optimization. Below the relations in the controllers used for data generation in the homogeneous mode are briefly presented. The output of the MTC controller is calculated as

$$\mathrm{MTC} = \min(\max(3 + 97 \cdot k_1 \cdot n_1^{k_2} + n_2, 3), 100),$$

where $k_1 = 0.1$, $k_2 = 6$ if $\mathrm{N} > \mathrm{N_{ref}}$ and $k_1 = 1$, $k_2 = 3$ otherwise. n_1 is a random signal, uniformly distributed in $[0, 1]$, and n_2 is a normally (Gaussian) distributed random signal with zero mean and a variance of 0.1.

The output of the $\mathrm{T_{inj}}$ controller is calculated as

$$\mathrm{T_{inj}} = \min(\max(\mathrm{T_{inj,base}} \cdot n_1 + \mathrm{minM_{air}} \cdot n_2, \mathrm{maxM_{air}}), \mathrm{minM_{air}}),$$

where

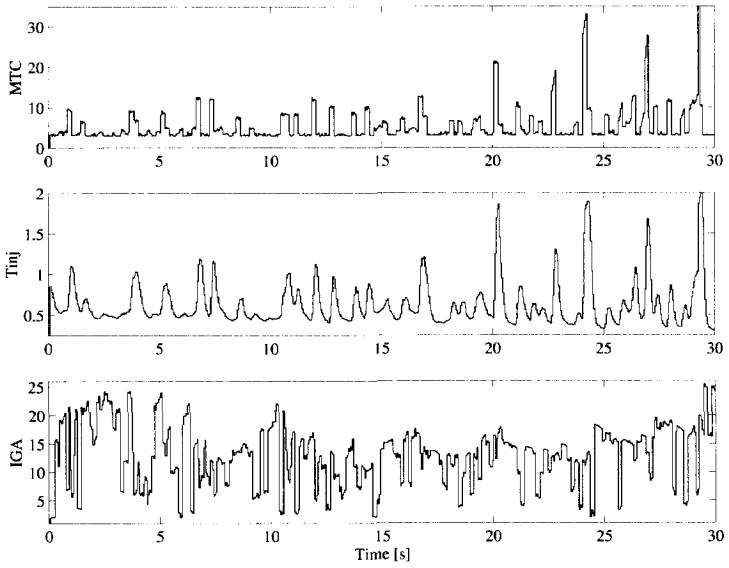
$$\mathrm{T_{inj,base}} = \min(50, \mathrm{maxM_{air}}) - \max(0, \mathrm{minM_{air}}),$$

n_1 is a random signal, uniformly distributed in $[0, 1]$, and n_2 is a normally (Gaussian) distributed random signal with zero mean and a variance of 0.00005. $\mathrm{M_{air}}$ is a function of the engine speed and throttle input, $\mathrm{M_{air}} = \mathrm{M_{air}}(\mathrm{N_{ref}}, \mathrm{N}, \mathrm{MTC})$, and $\mathrm{minM_{air}} = 1.8 \cdot 10^{-4}\mathrm{M_{air}} + 0.15$ and $\mathrm{maxM_{air}} = 0.1126 \cdot 10^{-4}\mathrm{M_{air}} + 0.15$.

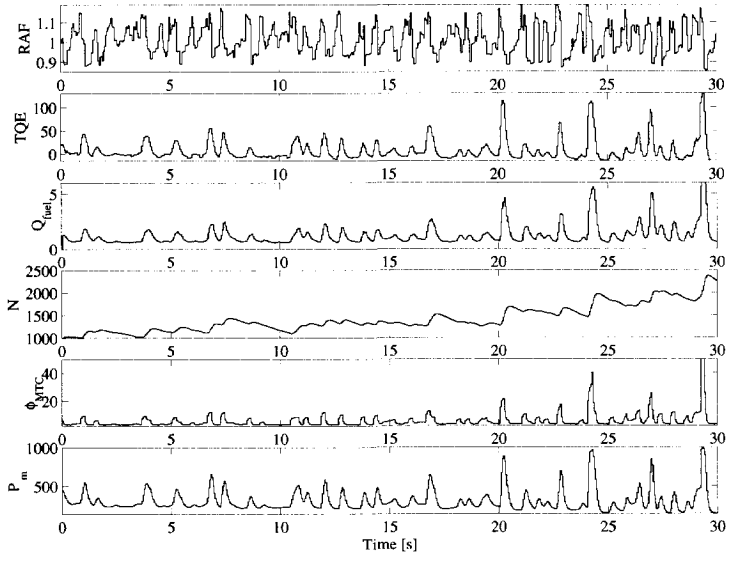
The signals IGA and SOI are determined through look-up tables, using the current engine speed, the flow of fuel in the cylinder and the computed $\mathrm{T_{inj}}$.

It should be noted that these controllers are designed only to generate data and cannot be used to control the engine, as they cannot handle the switches between the combustion modes. The controllers only maintain a "stable" mode (HM or SM) operation and thus it is not possible to bring the engine to a regime close to the one during a mode switch. As consequence, the identification data miss such regimes.

Figure 6.9 shows a part of the identification data set for the HM, sampled with sampling time of $T_s = 5$ ms. The apparent correlation between $\mathrm{Q_{fuel}}$ and $\mathrm{P_m}$ is not only caused by the engine characteristics but also a result of the control actions ($\mathrm{T_{inj}}$), as the fluctuations in it are due to RAF variations. The RAF is controlled to stay within 15% around one, a region where no control action is supplied. The engine speed increases, as intended.



(a) Engine Outputs



(b) Engine Inputs

Figure 6.9. Part of the identification data set for the HM.

6.3.2 TS Models

The characteristics of the engine depend on the combustion mode, therefore separate TS models (2.2) are developed for each mode. These are optimized with respect to optimal efficiency during a "stable" mode operation (i.e., when a switch is not expected and has not just been performed). However, at the moment of switching, the engine variables (states and outputs) have to allow operation in either mode. This results in a lower engine efficiency (sub-optimal operation), which cannot be predicted by the fuzzy models. More precisely, when the fuzzy model for the SM is used, it is not possible to bring the manifold pressure down during the transition to the HM. An additional model thus prove to be necessary for the transient SM→HM. For the transient mode HM→SM, the fuzzy model for the HM is still used. The transient mode SM→HM fuzzy model is derived based on a subset of the data generated in the HM, where RAF is in the interval $[1, 1.5]$.

The structure of the different TS models is selected using prior knowledge and some testing during the identification stage. The four engine inputs can be used as model inputs. In the HM, the start of injection is constant and MTC, T_{inj} and IGA are subject to optimization. Therefore only these are included in the model. In the SM, the start of injection and the ignition advance angle are pre-described according to a look-up table. The regressors for the outputs in the HM, SM and SM→HM models are given in Tab. 6.2 through Tab. 6.4, respectively.

The manifold pressure submodel for the HM is given as an example. The membership functions for the linguistic terms ('Low', 'Medium', etc.) were constructed by using fuzzy clustering. The functions used in the above manifold pressure submodel are shown in Fig. 6.10. The consequent parameters were estimated by local least squares (Appendix A).

$$
\begin{aligned}
&1.\ \textbf{If } \mathrm{N}(k) \text{ is } \textsc{Low} \textbf{ and } \phi_{\mathrm{MTC}}(k) \text{ is } \textsc{Zero} \textbf{ and } \mathrm{P_m}(k) \text{ is } \textsc{Very low} \textbf{ then} \\
&\quad \mathrm{P_m}(k+1) = -0.0178 \cdot \mathrm{N}(k) + 4.6 \cdot \phi_{\mathrm{MTC}}(k) + 0.92 \cdot \mathrm{P_m}(k) + 25.9 \\
&2.\ \textbf{If } \mathrm{N}(k) \text{ is } \textsc{Medium} \textbf{ and } \phi_{\mathrm{MTC}}(k) \text{ is } \textsc{Low} \textbf{ and } \mathrm{P_m}(k) \text{ is } \textsc{Low} \textbf{ then} \\
&\quad \mathrm{P_m}(k+1) = -0.0413 \cdot \mathrm{N}(k) + 6.12 \cdot \phi_{\mathrm{MTC}}(k) + 0.798 \cdot \mathrm{P_m}(k) + 111 \\
&3.\ \textbf{If } \mathrm{N}(k) \text{ is } \textsc{High} \textbf{ and } \phi_{\mathrm{MTC}}(k) \text{ is } \textsc{Medium} \textbf{ and } \mathrm{P_m}(k) \text{ is } \textsc{Medium} \textbf{ then} \\
&\quad \mathrm{P_m}(k+1) = -0.0148 \cdot \mathrm{N}(k) + 0.6 \cdot \phi_{\mathrm{MTC}}(k) + 0.765 \cdot \mathrm{P_m}(k) + 218 \\
&4.\ \textbf{If } \mathrm{N}(k) \text{ is } \textsc{Very high} \textbf{ and } \phi_{\mathrm{MTC}}(k) \text{ is } \textsc{High} \textbf{ and } \mathrm{P_m}(k) \text{ is } \textsc{High} \textbf{ then} \\
&\quad \mathrm{P_m}(k+1) = -0.005 \cdot \mathrm{N}(k) + 4.1 \cdot \phi_{\mathrm{MTC}}(k) + 0.835 \cdot \mathrm{P_m}(k) + 23.1
\end{aligned}
\tag{6.2}
$$

To achieve a satisfactory prediction accuracy of the fuzzy models, one must choose the number of rules properly. This number was optimized with respect to the obtained accuracy of prediction. It is important to consider the prediction error on a validation data set as well as the evaluation of the model within the controller. Therefore for each of the operation regimes, "a batch" identification was carried out that produced an extensive set of fuzzy models with different numbers of rules. Each of these fuzzy models was first validated on a data set different from the identification data set. Next, the models were utilized in the MPC to predict the engine outputs for a set of operating

Table 6.2. Regressors selected for the different outputs in the HM.

Output	Outputs					Inputs		
	TQE (k)	Q_{fuel} (k)	N (k)	ϕ_{MTC} (k)	P_m (k)	MTC (k)	T_{inj} (k)	IGA (k)
RAF (k+1)			×		×		×	
TQE (k+1)		×	×		×		×	×
Q_{fuel} (k+1)		×					×	
N (k+1)	×		×		×		×	×
ϕ_{MTC} (k+1)				×		×		
P_m (k+1)			×	×	×			

Table 6.3. Regressors selected for the different outputs in the SM.

Output	Outputs					Inputs	
	TQE (k)	Q_{fuel} (k)	N (k)	N (k-1)	P_m (k)	$\begin{bmatrix} MTC(k) \\ MTC(k-1) \\ MTC(k-2) \end{bmatrix}$	T_{inj} (k)
RAF (k+1)		×	×		×		
TQE (k+1)			×		×		×
Q_{fuel} (k+1)			×				×
N (k+1)	×		×				
N (k)				×			
P_m (k+1)			×	×	×	×	

Table 6.4. Regressors selected for the different outputs in the transient mode SM→HM.

Output	Outputs					Inputs	
	TQE (k)	Q_{fuel} (k)	N (k)	ϕ_{MTC} (k-1)	P_m (k)	MTC (k)	T_{inj} (k)
RAF (k+1)			×		×		×
TQE (k+1)			×		×		×
Q_{fuel} (k+1)			×				×
N (k+1)	×		×				×
ϕ_{MTC} (k+1)				×		×	
P_m (k+1)			×	×	×		

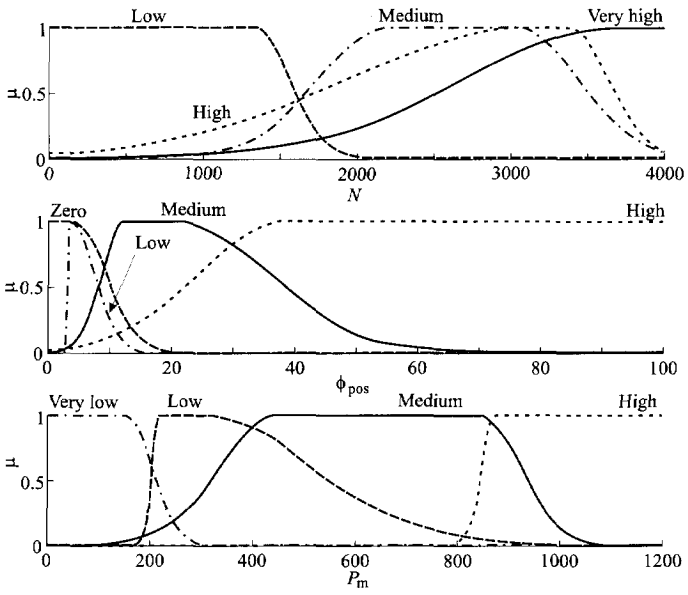


Figure 6.10. Membership functions.

conditions. Table 6.5 displays the numbers of rules for the different fuzzy models for which best validation and prediction performance was achieved.

Table 6.5. Number of fuzzy rules for different models.

Output	HM	SM	SM→HM
RAF	9	7	7
TQE	6	7	7
Q_{fuel}	4	6	6
N	4	1	3
ϕ_{MTC}	1	—	1
P_m	4	9	5

6.4 Control design

The EMS reacts to the driver's commands, given by the acceleration, brake and clutch pedals, and the gear engaged. Further, the EMS has to deal with different disturbances such as noisy sensor data and varying loads.

The controller operation is described by four modes: two corresponding to the combustion modes HM and SM, and two transient modes: HM $\rightarrow$ SM and SM $\rightarrow$ HM. The transient modes are necessary to prepare the engine for switching, as the combustion modes are based on different physical phenomena.

The implemented controller configuration consists of an *MPC optimizer*, assisted by *a mode switching logic* (Fig. 6.11). The MPC optimizer provides control signals based on the minimization of a cost function which guarantee optimal operation within the given combustion mode. To gain maximal efficiency from the engine, however, the selection of the combustion modes should be based on the demands on the torque and the engine speed. The mode switching could be based on the optimization of a cost function too, but this would result in an excessive computational load. Therefore the signal optimization is only performed within the MPC (containing all models of the engine), while the strategy for switching between the combustion modes is embedded in the switching logic. This logic is necessary to prepare the engine for the switch, as the two modes require different settings for the manifold pressure, the air/fiel ratio, etc. The switching logic facilitates this by suitably modifying the cost function, constraints and references.

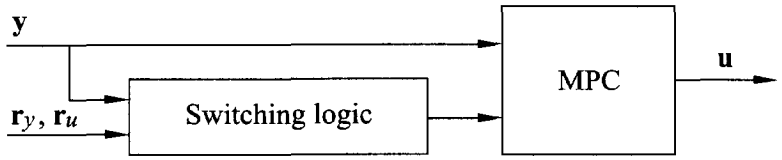


Figure 6.11. Control configuration.

6.4.1 MPC optimizer

The predictive controller described in Chapter 4 is utilized. It optimizes the following cost function

$$\min_{\mathbf{u}} J = \sum_{i=1}^{H_p} \left\| \mathbf{r}_y(k+i) - \hat{\mathbf{y}}(k+i) \right\|_{\mathbf{P}}^2 + \sum_{i=1}^{H_p} \left\| \Delta\hat{\mathbf{y}}(k+i-1) \right\|_{\Delta\mathbf{P}}^2 + \sum_{j=1}^{H_c} \left\| \mathbf{r}_u(k+j-1) - \mathbf{u}(k+j-1) \right\|_{\mathbf{Q}}^2 + \sum_{j=1}^{H_c} \left\| \Delta\mathbf{u}(k+j-1) \right\|_{\Delta\mathbf{Q}}^2 \quad (6.3)$$

where $\mathbf{r}_u$ and $\mathbf{r}_y$ are the input and output references, respectively, and $\mathbf{u} = [\mathrm{MTC}, \mathrm{T_{inj}}, \mathrm{IGA}]^T$ and $\mathbf{y} = [\mathrm{RAF}, \mathrm{TQE}, \mathrm{Q_{fuel}}, \mathrm{N}, \phi_{\mathrm{MTC}}, \mathrm{P_m}]^T$. The inputs and outputs are subject to level and rate constraints. The fuzzy models derived in Section 6.3 are used to predict the engine outputs during operation in the corresponding mode.

The second term in the cost function is used to restrict the output variations. During the switching periods, penalizing the variations is preferred to imposing explicit constraints, as the latter may result in infeasibility. The third term provides the necessary mechanism to manipulate the inputs during the transient modes: e.g., keeping

the T_{inj} (and thus the fuel flow) constant while closing the air throttle MTC reduces the air flow and hence the air/fuel ratio. Because of the closed-loop data generation (Section 6.3.1), the switching behaviour of the engine is not modeled precisely. Since the fuzzy models cannot accurately predict the engine outputs during a switch, the single-model method (Section 4.3) is preferred. Depending on the controller's operating mode (HM, HM→SM, SM or SM→HM), the corresponding fuzzy model is linearized at the current operation point and the derived linear model is used in the optimization routine.

The prediction and control horizons are selected using the guidelines proposed by Soeterboek (1992). The car is well damped, therefore the prediction horizon is $H_p = \text{int}(t_s/T_s)$ where the settling time $t_s \approx 50\text{ms}$ for a large part of the operation region (Bortolet et al., 1998), thus $H_p = 10$. Since the engine is of order two, according to Soeterboek (1992) a control horizon $H_c = 2$ was initially applied. However, experiments related to the switching between the combustion modes showed that the control horizon $H_c = 8$ results in the best performance. When $H_c = 8$ is used, the control signal has additional degrees of freedom that are necessary for a fast reaction during a switch in the combustion mode. The other parameters – the weighting matrices and the input and output constraints – vary in the different controller's modes. Reference signals are only used for RAF, TQE and Q_{fuel} (Section 6.2), so only these are weighted (Tab. 6.6). For N, ϕ_{MTC} and P_m only level and rate constraints are considered (Tab. 6.7). Note that in order to avoid infeasibility problems in the optimization routine, the output constraints are relaxed during the switching periods. In this way the stability of the control system is ensured, since feasibility rather than optimality suffices for guaranteeing stability (Michalska and Mayne, 1993; Scokaert et al., 1999).

Table 6.6. *Diagonal elements of the weighting matrices.*

Weight	HM	HM → SM	SM	SM → HM
P	$[2000, 4, 0]$	$[100, 2, 0]$	$[0, 160, 1]$	$[1000, 15, 0]$
$\Delta\mathbf{P}$	$[0, 50, 0]$	$[0, 400, 0]$	$[0, 0, 0]$	$[0, 0, 0]$
Q	$[0, 0, 0.01]$	$[0, 50, 0.01]$	$[0, 0]$	$[0, 0]$
$\Delta\mathbf{Q}$	$[150, 10, 1]$	$[0, 10^6, 0]$	$[100, 2 \cdot 10^3]$	$[0, 100]$

Some of the input constrains are imposed by the system itself, e.g., the level constraints on IGA in HM and HM→SM depend non-linearly on the throttle position, the manifold pressure, and the engine speed (Section 6.2)

$$\text{IGA}_{\text{min}}(\phi_{\text{MTC}}, \text{P}_\text{m}, \text{N}) \leq \text{IGA} \leq \text{IGA}_{\text{base}}(\phi_{\text{MTC}}, \text{P}_\text{m}, \text{N}),$$

where IGA_{min} and IGA_{base} are known nonlinear functions of ϕ_{MTC}, P_m and N, given analytically and through look-up tables (denoted by LUT in Tab. 6.7).

To reduce the influence of noise, Butterworth filters are used in the IMC scheme (Fig. 4.1). They are first-order filters, with cut-off frequencies 35 Hz for RAF and

Table 6.7. Level and rate constraints.

Mode		Output			Input		
		RAF	TQE	N	MTC	T_{inj}	IGA
HM	level			$[900,3700]$	$[0,100]$	$[0,3.5]$	LUT
	rate				$[-10,10]$	$[-0.05,0.05]$	
HM→SM	level	$[0.5,\infty)$	$[-50,100]$	$[900,3700]$	$[0,100]$	$[0,3.5]$	LUT
	rate				$[-1,1]$	$[-0.1,0.1]$	—
SM	level			$[900,3000]$	$[0,100]$	$[0,3.5]$	
	rate		$[-20,20]$	$[-100,100]$	$[-1,1]$	$[-0.1,0.1]$	—
SM→HM	level		$[-50,49.5+TQL]$	$[900,3000]$	$[0,100]$	$[0,3.5]$	
	rate			$[-100,100]$	$[-0.02,0.02]$	—	

20 Hz for TQE, respectively, taking into account the different dynamics of these outputs (Bortolet et al., 1998).

The control objectives are defined with respect to three of the engine's outputs: the air/fuel ratio, the fuel consumption and the produced torque. The references for the air/fuel ratio and the fuel consumption, imposed by the engine and the combustion mode, are available beforehand. The torque reference is based on the driver's torque demand expressed through the accelerator pedal, and as such is not known beforehand. Therefore it is kept constant over the prediction horizon. The following paragraphs discuss in more detail how the references are derived. The reference values are summarized in Table 6.8 on page 127.

Air/fuel ratio reference. The controller should keep the air/fuel ratio at a constant value (RAF $= 1$) in the HM operation. In SM, the engine must run in lean-burn condition, i.e., RAF > 1. Thus a reference value for RAF is only used in the HM: RAF_{ref} $= 1$. During the transient modes, the RAF_{ref} is modified to bring the RAF up or down, depending on the mode.

Fuel flow reference. The fuel consumption is an integral of the fuel flow Q_{fuel} that goes into the cylinder, hence for low consumption the fuel flow should be as low as possible. Therefore, the reference for the fuel flow is set to a value corresponding to consumption of six liters per 100 km.

Torque reference. The torque is the engine's output which directly influences the car's acceleration and thus much effort is spent on controlling it. The torque reference TQE_{ref} is designed to take into account other factors, such as the engine speed, the idle-speed control and the load disturbances:

1. Idle speed control. When no gear is engaged or the engine is declutched from the transmission, we have an idle speed operation. The engine speed in this phase should be close to $N_{idle} = 1000\,rpm$.

2. Safety constraints. During the decelerating phases, the torque is negative, i.e., the engine brakes. The maximal negative torque is limited from below by $-20\,Nm$, which corresponds to RAF = 100 (almost fuel-free). Despite being very attractive from an economic point of view, a big negative torque is far from desired as in the beginning of an acceleration phase the engine efficiency is low, due to the insufficient fuel content. The TQE cannot follow the reference during a period needed for getting the RAF down. To avoid this situation, the maximum allowed negative torque is restricted to $-6\,Nm$, i.e., when the reference TQE_{ref} goes below $-6\,Nm$, it is restored to that value.

3. Critical engine speed. The engine speed must always be higher than the critical engine speed $N_C = 900\,rpm$ which is the lowest level before the engine stalls. When the N drops below a prescribed value (925 rpm), an additional critical-speed controller is engaged to compensate for the difference between the current and the idle engine speed $N_{idle} = 1000\,rpm$.

4. Load disturbance rejection. Load disturbances such as lights and the air conditioner are also considered. The lights' load is represented by an increase in the torque demand of 3.5 Nm. This torque demand is relatively low and the controller does not consider it explicitly unless some of the safety constraints are violated. In such a case, the corresponding safety controller is engaged.

   The air conditioner's load is represented by an increase in the torque demand by 15 Nm. Contrary to the light load, here the request can be delayed for a period of at most five seconds, which allows the controller to increase the TQE_{ref} gradually. This prevents the engine from leaving the SM. The same strategy is used when the air conditioner is turned off: TQE_{ref} is linearly brought down for another five seconds.

6.4.2 Mode switching logic

The mode switching logic determines the current controller's mode of operation and adapts the relevant control parameters such that smooth alternation between HM and SM is provided (e.g., to keep the torque gradient within the allowed limits). The controller mode can have one of the following values (Fig. 6.12):

1. HM operation;

2. HM→SM operation (transient mode);

3. SM operation;

4. SM→HM operation (transient mode).

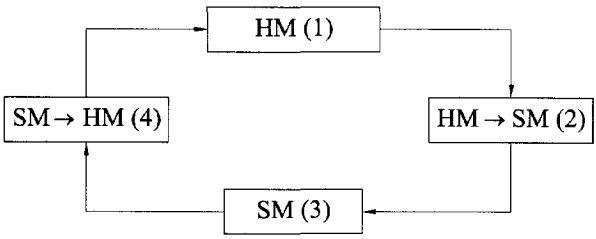


Figure 6.12. Controller modes.

To accomplish a switch, the controller goes through a sequence of operation modes: $(1) \rightarrow (2) \rightarrow (3)$ for a switch from the HM to the SM, or $(3) \rightarrow (4) \rightarrow (1)$ from the SM to the HM.

At the moment of switching, the engine variables (states and outputs) have to be such that they support operation in either mode. However, this is accompanied by a drop in the combustion efficiency. In the first couple of milliseconds after the switch, the efficiency in the new mode is lower than in the old one. As a result, since the aim is to gain maximal efficiency, a new switch would be initiated to the mode just left. To prevent such a back switch, a dead time is introduced at the moment of switching. During this dead time the engine variables are changed to values that ensure optimal operation in the new combustion mode. The dead time varies from three to six samples ($T_s = 5$ ms) for the different gears, and is used to adapt the controller parameters (weights and constraints) and the references.

The switching strategies are summarized below. Note that the signal variations are restricted by means of the weights in the cost function, rather than by using constraints. This is done on the one hand to avoid infeasibility problems in the optimization routine, and on the other hand it allows priority changes during operation. The weights are modified to obtain desired compromise between the requirements imposed on the different inputs and outputs. It appeared that the tuning of the weights and especially the timing is critical for the stable operation of the engine.

Switching strategy SM→HM. The stratified mode is preferred as it requires less fuel and reduces the pollution, but its operation range is limited with respect to the engine speed and the indicated torque, see (6.1) on p. 114. The controller's mode of operation is therefore based on the current and predicted values of these signals. For the current values, safety margins of one Nm and 50 rpm are included for N and TQI, respectively. Since the mode switch is accomplished within a single engine cycle and the driver's actions are not known in advance, N and TQI are predicted only one step ahead. Thus as soon as any of the constraints

$$\mathrm{TQI} < 49, \quad \mathrm{N} < 2950, \quad \mathrm{TQI}_{\mathrm{pred}} < 50, \quad \mathrm{N}_{\mathrm{pred}} < 3000 \tag{6.4}$$

is violated, a switch is initiated. During the switching period, RAF must decrease from about 4.5 to one, guaranteeing the most efficient combustion in the HM. The RAF can be reduced either by increasing the fuel flow, or by decreasing the air flow. Increasing the fuel flow will cause a rapid increase in the produced torque, which does not

only negatively influence the driving comfort, but may also damage the combustion chambers and the crankshaft. Therefore RAF is reduced through the air flow, by manipulating the air-throttle position. The switching strategy is summarized as follows.

1. Decrease the RAF reference in order to bring the RAF down (approximately to one), which is necessary for an efficient HM operation. The relative importance between the RAF and the TQE at this moment is expressed by increasing the weight for the RAF in the cost function (6.3): $\mathbf{P}_{\mathrm{RAF}} = 5000$, while $\mathbf{P}_{\mathrm{TQE}} = 1$. These temporal terms are used during the first 0.2 s in the HM instead of the nominal values $\mathbf{P}_{\mathrm{RAF}} = 2000$ and $\mathbf{P}_{\mathrm{TQE}} = 4$ (Tab. 6.6).

2. Allow large variations in MTC ($\Delta\mathbf{Q}_{\mathrm{MTC}} = 0$), while restricting $\mathrm{T_{inj}}$ variations ($\Delta\mathbf{Q}_{\mathrm{T_{inj}}} = 1000$). The nominal weights are $\Delta\mathbf{Q}_{\mathrm{MTC}} = 150$ and $\Delta\mathbf{Q}_{\mathrm{T_{inj}}} = 10$ (Tab. 6.6).

3. To achieve good driving comfort during the switch, use a rate limiter to restrict the variations in the TQE reference within 0.5 Nm per sample.

Switching strategy HM→SM. The switching is initiated when constraints (6.4) are satisfied. The switching strategy is summarized as follows.

1. The RAF reference increases to 1.6 to make the RAF sufficiently high to allow operation in the SM. The importance of the RAF with respect to the TQE is expressed by increasing the weight on the RAF in the cost function (6.3): $\mathbf{P}_{\mathrm{RAF}} = 1000$, while $\mathbf{P}_{\mathrm{TQE}} = 50$. These temporal terms are used during the first 0.2 s in the SM instead of the nominal $\mathbf{P}_{\mathrm{RAF}} = 0$ and $\mathbf{P}_{\mathrm{TQE}} = 160$ (Tab. 6.6).

2. In both HM→SM and SM, TQE is controlled by using $\mathrm{T_{inj}}$, rather than by MTC. In the HM, MTC is open relatively little (5 – 10%), while in the SM most of the time it is open about 80% and more. At the moment of switching large variations in the MTC are allowed while the variations in the $\mathrm{T_{inj}}$ are restricted. As a result, on the one hand RAF rapidly increases while on the other the fuel flow is kept low to prevent large gradients in the TQE. This is accomplished by using different weights in the cost function: $\Delta\mathbf{Q}_{\mathrm{MTC}} = 0$, while $\Delta\mathbf{Q}_{\mathrm{T_{inj}}} = 10000$. They are used during the first 0.2 s in the SM instead of the nominal $\Delta\mathbf{Q}_{\mathrm{MTC}} = 100$ and $\Delta\mathbf{Q}_{\mathrm{T_{inj}}} = 2000$ (Tab. 6.6).

Table 6.8. Reference values in the different modes. V_{ref} denotes the desired car speed.

Signal	HM	HM→SM	SM	SM→HM
RAF	1	$1 \rightarrow 1.6$	≥ 3, depending on TQE	Decrease to 1
$\mathrm{Q_{fuel}}$	$\frac{6 \cdot 0.75 V_{\mathrm{ref}}}{100}$	$\frac{6 \cdot 0.75 V_{\mathrm{ref}}}{100}$	$\frac{6 \cdot 0.75 V_{\mathrm{ref}}}{100}$	$\frac{6 \cdot 0.75 V_{\mathrm{ref}}}{100}$
TQE	See Section 6.4.2 and Section 6.4.1			

6.5 Results

In order to evaluate the fuel consumption and the pollutant gas emission of a car, the European Community has defined a normalized driving cycle, in which a combination of speed and gear reference is imposed on the driver (Commission, EU, 1993). A driving scenario with the duration of 595 s, extracted from the European driving cycle, is used to demonstrate the performance of the controller (Fig. 6.13). During the driving scenario, 26 combustion mode switches are performed. The engine spends about the same time in the HM and the SM. The mean fuel consumption of 7.12 l/100km is comparable to the consumption achieved by other controllers, developed within the FAMIMO project (Boverie, 2000).

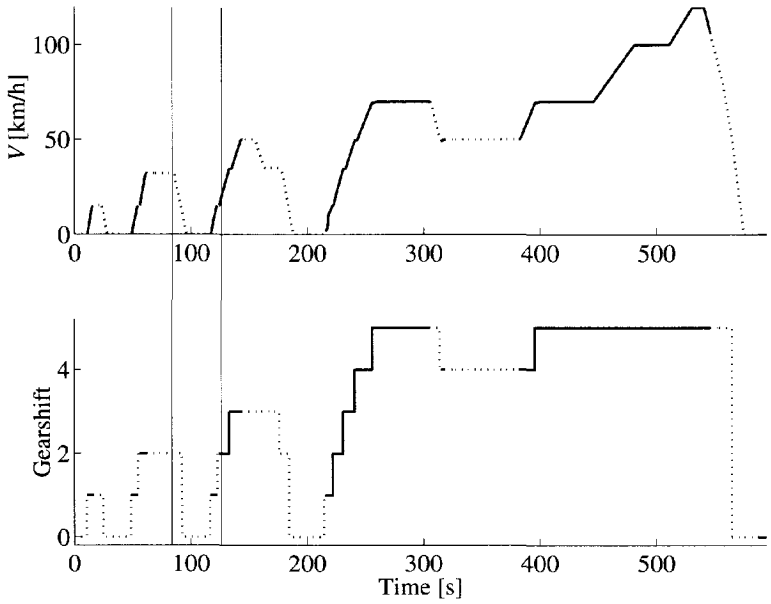


Figure 6.13. Car speed and gearshift engaged. SM (- -) and HM (–).

To give a more detailed picture of the behaviour of the car, we show the subinterval from the 85th to 125th second in Fig. 6.14 to Fig. 6.15. In this period all the considered events are present: gear changes, switches in the combustion mode, an idle speed phase and additional loads. Figure 6.14a shows the engine speed N, the clutch position, and the car speed V. During the first ten seconds, the car decelerates. At the moment of declutching, the idle speed controller is engaged (Fig. 6.14b), which brings the TQE to 0 Nm and N to $N_{idle} = 1000$ rpm. At $t = 106$ s the lights are turned on, which increases the torque demand. This causes the engine speed to drop approximately to 967 rpm, still well above the critical engine speed $N_C = 900$ rpm. At $t = 116$ s the first gear is engaged. The controller switches from the SM to the HM, as the former cannot provide the torque required (Fig. 6.14b). The necessary drop in

RAF (Fig. 6.15a) is achieved through a rapid decrease of MTC (Fig. 6.15b). However, this leads to a significant undershoot in RAF (RAF $\approx$ 0.4), thus reducing the engine's effectiveness (Fig. 6.14b). As soon as the constraints (6.4) are satisfied (at $t \approx 122$ s), a switch to the SM is initiated. Again, when second gear is engaged, the controller switches to the HM.

The control strategy has been applied to a reduced version of this benchmark that comprises only the engine model, see Fig. 6.5. Different engine configurations have been tested, based on combinations of aging phenomena and sensor noise as follows. The aging phenomena are expressed through variations in the torque losses, such as: (i) increased and (ii) reduced torque losses, and through sensor flaws given through white noise perturbations with amplitude 5% of the measured signal for N, P_m and RAF.

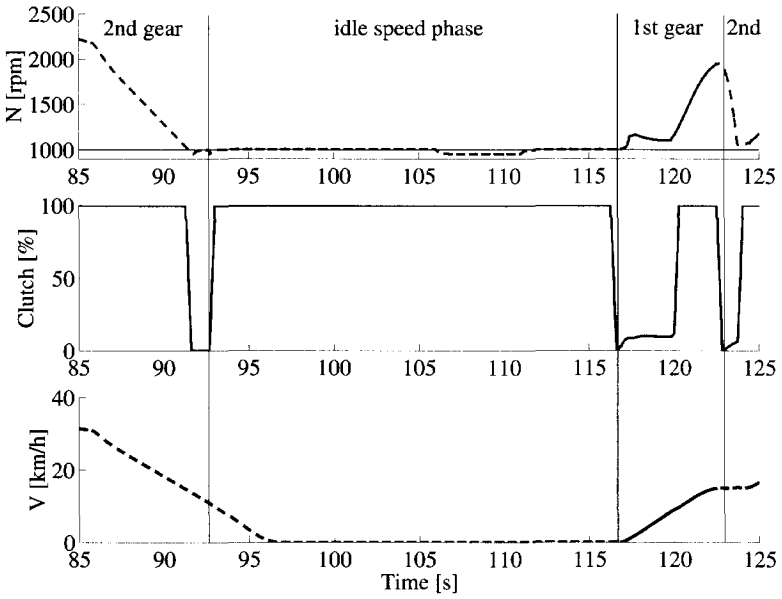
The simulations carried out showed that the controller is robust against torque variations, but is rather noise sensitive. To deal with the noisy sensors, we designed filters on the important signals (TQE, N and P_m) and introduced additional parameters dealing with the noise. For example, a switch in the combustion mode is initiated only if there is a constraint violation in three consequent samples. Note that because of the safety margins, the delay introduced in this way does not lead to violation of the technological constraints. With these modifications, reasonable performance has been achieved also for the noisy cases. Here, reasonable means that the stable operation is maintained, although at the expense of, e.g., increased fuel consumption and intensified throttle action.

6.6 Comparison with other control strategies

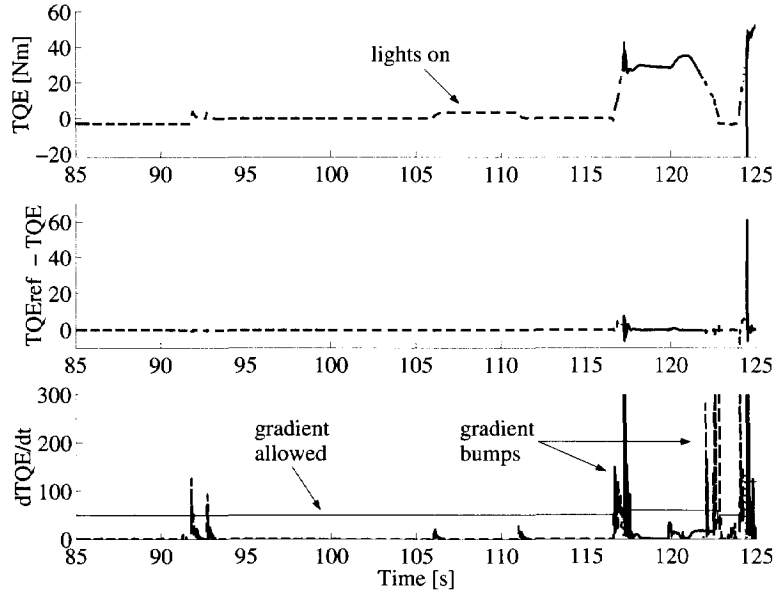
The presented control design is compared with two other approaches, developed within the FAMIMO project by the Department of Automatic Control of the Lund Institute (Sweden) and the Advanced Development Department of Siemens Automotive SA in Toulouse (France), respectively. Both of them are based on functional decomposition of the engine into subsystems (Fig. 6.16) and the complete control system is built by integrating local controllers, designed for the different sub-problems (Boverie, 2000).

The difference between these two approaches is in the way the control structure is realized: several combinations of feedforward and feedback controllers can be utilized to control the produced torque and the air/fuel ratio. In the Lund design, the linear feedback loops are predominant, hence the name *linear feedback design*. The feedforward actions are given by (almost) linear inversions of the mapping between the system inputs and outputs. The priority is put on the minimization of the fuel consumption. This is done through an extremum seeking controller for the optimization of the fresh air flow introduced into the cylinders.

The approach exploited by Siemens mainly relies on feedforward control, and the feedback actions are principally used to reject disturbances, hence the name *fuzzy feedforward design*. The multiple-model approach is used to derive a TS fuzzy model of the engine. The corresponding fuzzy controller is essentially an inverse controller.

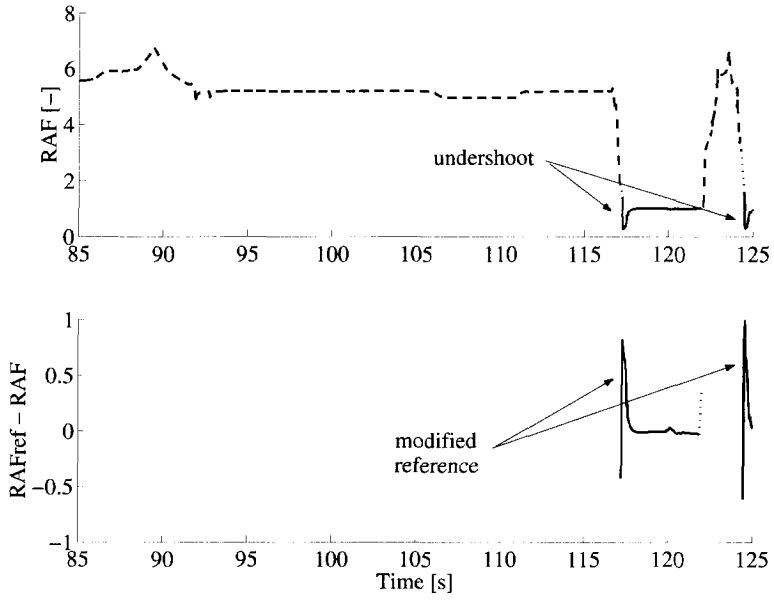


(a) Engine speed, clutch and car speed.

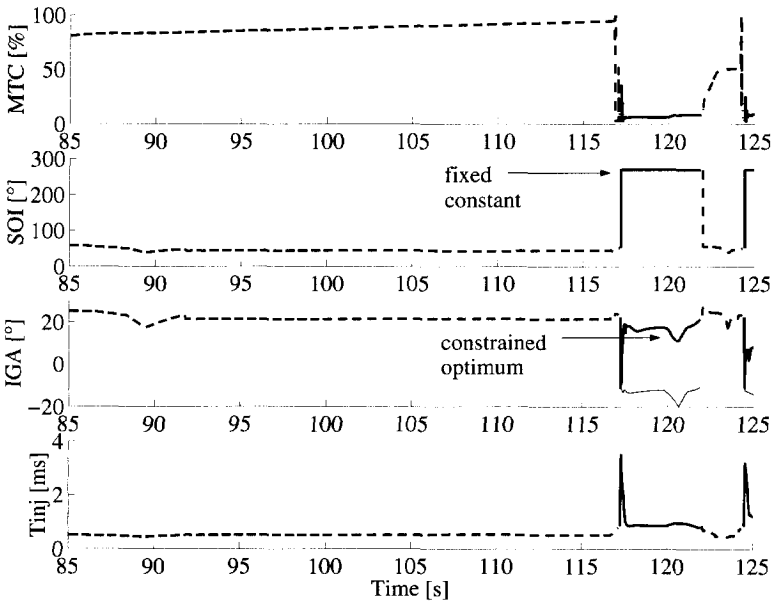


(b) Effective torque, torque variation form the reference and torque gradient.

Figure 6.14. Car behaviour: (a) engine speed, clutch and car speed, (b) effective torque, torque error and torque gradient.



(a) Air/fuel ratio and ratio error.



(b) Engine inputs.

Figure 6.15. Car behaviour: (a) Air/fuel ratio and ratio error, (b) engine inputs.

The priority is given to the air/fuel regulation in addition to the torque regulation, and the efficiency of the combustion process is controlled through the ignition advance.

The main difference between the fuzzy MPC design on the one hand and the linear feedback and fuzzy feedforward designs on the other is in the amount of "white-box" technological knowledge incorporated in the design. The linear feedback design follows the approach currently used in car manufacturing: the EMS is designed using engine maps (look-up tables) derived through extensive experiments with the engine prototype. The fuzzy MPC design, however, uses simple closed-loop experiments to derive prediction models. The fuzzy feedforward design is in between the fuzzy MPC and linear feedback designs, as it only uses fuzzy models to approximate unknown static relations.

A number of criteria have been defined to evaluate and compare the different designs.

Throttle criterion. The following criterion is used to evaluate the activity of the air throttle actuator

$$C_{\text{Throttle}} = \frac{\sum_k |\text{MTC}(k) - \text{MTC}(k-1)|}{N},$$

where $\text{MTC}(k)$ and $\text{MTC}(k-1)$ are the throttle actions at time k and $k-1$ respectively, and N is the total number of samples.

Consumption criterion. The consumption criterion is the mean consumption over the scenario:

$$C_{\text{Consumption}} = \frac{\int Q_{\text{fuel}} \cdot dt}{0.75 \cdot \int V \cdot dt} \cdot 100,$$

where V is the car speed, and 0.75 [kg/l] is the fuel density constant.

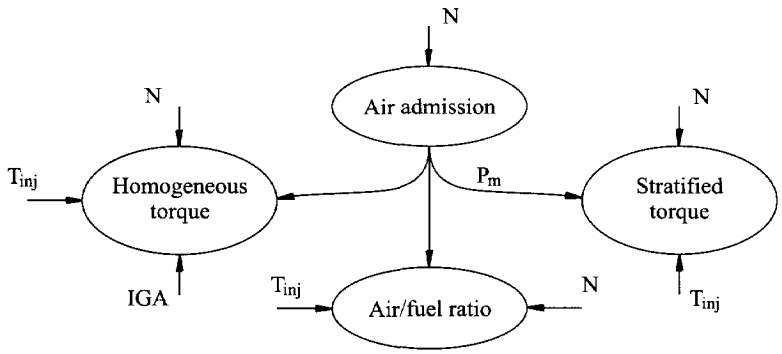


Figure 6.16. Engine functional decomposition. The *air admission* subsystem computes the manifold pressure P_m based on the engine speed N. The *air/fuel ratio* subsystem determines the RAF using the air and fuel flows. These flows are manipulated by the manifold pressure P_m and the injection time T_{inj}, respectively. Separate subsystems are used to administer the torque in the different combustion modes. In the HM, the torque depends on the engine speed, air and fuel flows, and the moment of ignition through IGA. In the SM, IGA is pre-described according to a look-up table.

Air/fuel ratio criterion. To evaluate the regulation of the RAF in the homogeneous mode, we used the following criterion

$$C_{RAF} = \frac{1}{T_{Homogeneous}} \int \left| \frac{RAF - RAF_{ref}}{\Delta RAF_{max}} \right| \cdot mode \cdot dt,$$

where $T_{Homogeneous}$ is the time spent in the HM, $RAF_{ref} = 1$ is reference for the air/fuel ratio, $\Delta RAF_{max} = 0.02$ is the maximum acceptable deviation from the reference, and *mode* is a boolean flag (1: homogeneous mode, 0: stratified mode).

Car speed criterion. The following criterion is defined to evaluate the car speed regulation,

$$C_v = \frac{1}{T} \int \left| \frac{V - V_{ref}}{\Delta V_{max}} \right| \cdot dt,$$

where T is the duration of the scenario, V_{ref} is the car speed reference defined by the driving scenario, and $\Delta V_{max} = 2\,\text{km/h}$ is the maximum acceptable deviation from the reference.

Idle speed criterion. The following criterion is defined to evaluate the idle speed regulation,

$$C_{N_{idle}} = \frac{1}{T_{idle}} \int \left| \frac{N - N_{idle}}{\Delta N_{idle\ max}} \right| \cdot idle \cdot dt,$$

where T_{idle} is the length of the idle phase period, $N_{idle} = 1000\,\text{rpm}$ is the engine idle speed, $\Delta N_{idle\ max} = 50\,\text{rpm}$ is the maximum acceptable deviation from N_{idle}, and *idle* is a boolean flag (0: non-idle phase, 1: idle phase).

Torque criteria. Several criteria are considered to evaluate the torque performance. To preserve *the car driveability* during acceleration and deceleration phases, it is necessary to limit the produced torque gradient. The corresponding criterion is

$$C_{Torque1} = \frac{1}{T} \int \frac{\max\left(\left| \frac{dTQE}{dt} \right| - \frac{dTQE}{dt}_{max}, 0 \right)}{\frac{dTQE}{dt}_{max} \cdot 0.2} \cdot dt,$$

where T is the duration of the scenario and $\frac{dTQE}{dt}_{max}$ is the maximal torque gradient authorized (Fig. 6.14b).

The main problem during the switching phases is to ensure that the torque change is within the allowed limit, so that the driver does not experience any hi-cough

$$|TQE_{before\ commutation} - TQE_{after\ commutation}| < 10\,\text{Nm}.$$

The respective criterion is defined as follows

$$C_{Torque2} = \frac{1}{Nb_{commutation}} \times \frac{\max\left(|TQE_{before\ commutation} - TQE_{after\ commutation}| - \frac{dTQE}{dt}_{max} \cdot T_s, 0 \right)}{\Delta TQE_{commutation\ max}},$$

where $Nb_{\text{commutation}}$ is the number of commutations between the two modes, $T_s = 5\,\text{ms}$ is the sample time of the controller, $\frac{d\text{TQE}}{dt}_{\max}$ is the maximal torque gradient allowed and $\Delta\text{TQE}_{\text{commutation max}} = 10\,\text{Nm}$ is the maximal torque deviation allowed during commutation (Fig. 6.14b).

Global criterion. A global criterion is defined that aggregates five of the seven criteria defined above:

$$C_{\text{Global}} = \frac{1}{5}\left(C_{\text{RAF}} + C_{\text{Torque1}} + C_{\text{Torque2}} + C_{\text{v}} + C_{\text{N}_{\text{idle}}}\right). \tag{6.5}$$

The objective of the controller is to minimize C_{Global}, C_{Throttle} and $C_{\text{Consumption}}$ as much as possible. Thus a lower value implies a more effective control strategy, resulting in more driving comfort, fewer control actions and less fuel consumption. The criterion C_{Global} is the most important one, as C_{Global} greater than *one* means that the EMS should not be used because it either does not comply with the specifications (high C_{v} and $C_{\text{N}_{\text{idle}}}$) or will damage the combustion chambers and the manifold (high C_{RAF}, C_{Torque1} and C_{Torque2}).

The criteria values for the discussed control strategies are summarized in Tab. 6.9. Comparing the fuel consumption, all designs use approximately equal amount of fuel. The lowest consumption is achieved with the linear feedback design, where an extremum seeking controller is applied to optimize the quantity of fresh air introduced into the cylinder during the SM phases. This is accomplished at the expense of very tight throttle control, as indicated by C_{Throttle}. In the fuzzy feedforward design (Siemens SA), ignition advance IGA is used instead of the throttle opening, which explains the low value achieved for this criterion. In the fuzzy MPC design, since the control actions are optimized with respect to a multivariable cost function, the effort is distributed between the different controls. As a result $C_{\text{Throttle}} = 0.04$ is higher than the value for the feedforward design, but much below the one for the linear design.

For the global criterion, the feedback and the feedforward designs achieved similar results, which were much better than the ones achieved by the fuzzy MPC. The main contributors to C_{Global} in the fuzzy MPC are C_{Torque1} and C_{Torque2} (6.5): the used fuzzy models are based on data collected for stable operation in a certain mode, and cannot accurately predict the torque during mode switches. This results in deviations from the reference and high torque gradients (Fig. 6.14b). The linear feedback design follows the torque reference much better (lower C_{Torque2}), however, at the expense of higher torque gradients (higher C_{Torque1}) during mode switches. On the contrary, the feedforward design tries to keep the torque gradients within the imposed limits, while

Table 6.9. Criteria results for the different control strategies.

Controller	C_{Global}	$C_{\text{Consumption}}$	C_{Throttle}
Fuzzy MPC design	0.59	7.12	0.04
Linear feedback design	0.0091	6.78	0.84
Fuzzy feedforward design	0.0102	6.87	0.016

allowing larger deviations from the reference (Boverie, 2000). The advantage that the linear feedback and fuzzy feedforward designs have over the fuzzy MPC design in terms of C_{Global} can be explained through the use of "white-box" technological knowledge. In the linear feedback design, the synthesis of the EMS controllers is based on the white-box engine model, and fuzzy models are used only to model the unknown nonlinearities. Local controllers are applied to the different subsystems also in the fuzzy feedforward design. The use of fuzzy models in this design is limited to the adaptation of the controller parameters depending on the operating conditions. Contrary, the fuzzy MPC design is centered on fuzzy models derived from experimental data. No white-box knowledge is used, except to provide the constraints on the input and output signals used in the MPC optimizer. Despite the relatively high value of C_{Global}, the fuzzy MPC design is still well below the allowed maximum, thus satisfying the imposed specifications. Using additional priori knowledge can further improve the control performance. For example, we could build a "gray-box" model of the engine, where TS models only represent the nonlinear input-state and state-output mappings.

Comparison concerning the complexity of designed EMSs is not straightforward. The linear feedback and the fuzzy feedforward designs are decentralized controllers reflecting the functional decomposition of the engine (Fig. 6.16). The separate controllers are based on output feedback or PIDs which can be individually tuned (an exception is the extremum seeking controller in the Lund design, which optimizes the air flow). The fuzzy MPC design is a centralized multivariable design, where the interactions between the different inputs and outputs are taken into account, and a compromise is made at each sampling instant. Here the complexity (measured in terms of time needed for producing the control actions) is considerable higher than reported for the two other approaches. So far, the fuzzy MPC has been implemented in MATLAB and has not been optimized for real-time application. While the current implementation needs 93ms on Pentium II 233MHz to compute one control action, we believe that it is possible to reduce the computation time to such an extend that it can be used in real time (recall that $T_s = 5$ms).

6.7 Summary and concluding remarks

This chapter presented a real-world application of fuzzy model predictive control. The process under consideration is a gasoline direct-injection engine. The core of the Engine Management System (EMS) controller consists of a model predictive control optimizer, with fuzzy models of the Takagi–Sugeno type derived through closed-loop identification. Because of the engine complexity, it was not possible to satisfactorily describe its behaviour in the whole domain of operation using a single fuzzy model. Different models were developed for each combustion mode and an additional model proved to be necessary for one of the transient modes. The fuzzy models were optimized with respect to the obtained accuracy of prediction, both on a validation data set and within the controller.

To avoid the drawbacks of the nonlinear optimization, we derived linear models from the fuzzy model at the current point and used in the MPC. Since the derived fuzzy models are not very accurate, at each sampling instant a single model was ex-

tracted and used during the whole prediction horizon. Switching logic was used to provide smooth switching between the combustion modes by means of modifying the references to be followed and the MPC parameters.

The most difficult part in the control design was the switching between the combustion modes. Not only different models were used, but when switching from one combustion mode to another the controller parameters were changed as well: the references were modified, temporary weights were introduced in the cost function (lasting less than half a second) and the constraints were relaxed.

The obtained results demonstrate the potential of fuzzy model predictive control for such a complicated system. Although slightly worse, for the most part they are comparable to the results achieved by using first-principle and technological knowledge. In the linear feedback and the fuzzy feedforward designs, the synthesis of the EMS controllers is based on the white-box engine model. The use of fuzzy models is limited to modeling the unknown nonlinearities, or to tuning the controller parameters in the different combustion modes. On the contrary, the fuzzy MPC design is based on fuzzy models derived from experimental data. Because the engine operation during switches in the combustion mode cannot be modeled accurately, the fuzzy MPC gives a poor performance expressed in terms of high torque gradients during such a switch. However, this design is the most general one. As it follows a relatively standard procedure and does not require detailed information about the engine, it can easily be applied to a different engine. In this sense, it can reduce the time and cost of the control design and the tuning phase.

7 FUZZY MODEL-BASED CONTROL OF A BINARY DISTILLATION COLUMN

This chapter presents the second application of the developed methods for fuzzy model-based control. The considered benchmark is a simulation model of a distillation column, used to separate a liquid mixture of two substances into its component fractions. Multivariable distillation columns pose a number of challenging problems for both system identification and control due to their nonlinear and ill-conditioned nature. The simulation model of the column proposed by S. Skogestad is introduced in Section 7.2. Section 7.3 describes the construction of the TS fuzzy models. The analysis of interactions is presented in Section 7.4. A decentralized control scheme with a decoupler design and a design of a multivariable controller are described in Section 7.5 and Section 7.6, respectively. The latter is based on the MPC algorithm that uses local linear models derived from the fuzzy model around the current operating point. To illustrate the influence of the model prediction on the achieved performance, a disturbance signal is included as an additional input of the fuzzy model. Finally, the robustifying effect of the stability constraints is shown. Comparison with MPC algorithms based on a Wiener model is given in Section 7.6.2.

7.1 Distillation unit

Distillation columns operate on the principle that the vapor of a boiling mixture is rich in the components that have lower boiling points. Therefore, when this vapor is cooled and condensed, the condensate will contain more volatile (light) components. The remaining mixture will contain more of the less volatile material, see Appendix F.

Distillation columns consist of several units such as a vertical shell, column stages (trays), reboiler, condenser, and reflux drum. These units are used to transfer heat energy or enhance material transfer. The vertical shell contains the column trays and together with the condenser and the reboiler constitutes the distillation column. A schematic diagram of a typical distillation column with a single feed (raw material) and two product streams is shown in Fig. 7.1.

The objective is to produce high-purity products at both ends of the distillation column. This means that we have to control the component compositions in the top and the bottom products, D and B, respectively. The top and bottom compositions are denoted by y_D and x_B, respectively.

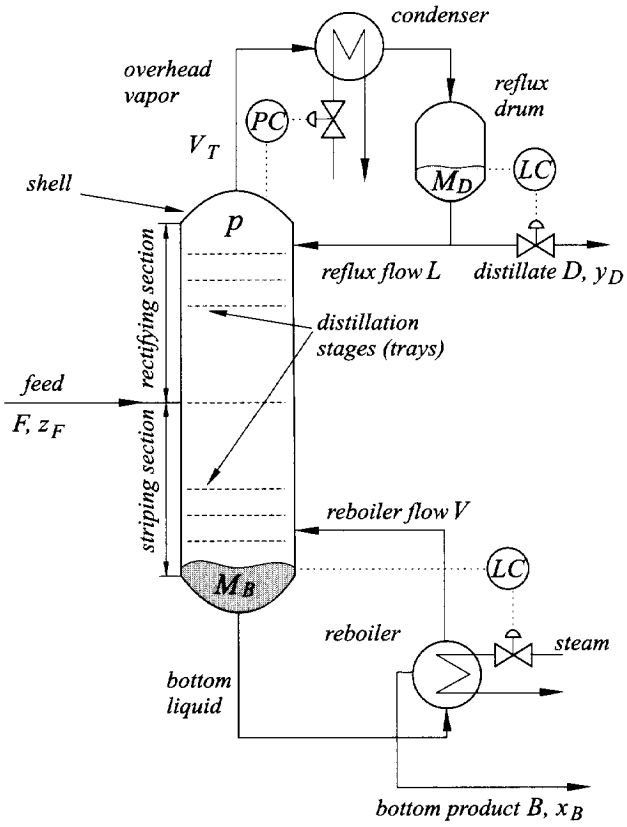


Figure 7.1. A schematic diagram of a typical distillation unit with a single feed and two product streams.

Among the signals in the distillation column, five manipulated variables (control inputs) and five controlled variables (outputs) can be distinguished. These are summarized in Tab. 7.1. Typically the distillation column is first stabilized by closing three decentralized (SISO) loops for level and pressure control, involving the condenser level, M_D, the reboiler level, M_B, and pressure in the shell, p,

$$\mathbf{y}_2 = [M_D \quad M_B \quad p]^T.$$

The three decentralized loops usually interact weakly and may be tuned independently of each other (Skogestad, 1997). The remaining two outputs are the product compositions

$$\mathbf{y}_1 = [y_D \quad x_B]^T.$$

There exist many possible choices for $\mathbf{u}_2$ that is used to control $\mathbf{y}_2$, and for $\mathbf{u}_1$ controlling $\mathbf{y}_1$. By convention, each choice ("configuration") is named by the inputs $\mathbf{u}_1$ left for composition control. Most commonly used is the "*LV*-configuration" with

$$\mathbf{u}_1 = [L \quad V]^T, \qquad \mathbf{u}_2 = [D \quad B \quad V_T]^T.$$

This means that the top product flow D, the bottom product flow B, and the overhead vapor V_T are used to stabilize the column, and the reflux and reboiler flows L and V, respectively, are used for composition control[1]. Hence, we end up with a 2×2 control problem as shown in Fig. 7.2.

Table 7.1. Control inputs, measured outputs and disturbances of the distillation column.

Control inputs	Symbol	Unit
Reflux flow	L	kmol/min
Reboiler flow	V	kmol/min
Distillate (top) product flow	D	kmol/min
Bottom product flow	B	kmol/min
Overhead vapor	V_T	kmol/min
Outputs	Symbol	Unit
Distillate (top) product composition	y_D	mole fraction, %
Bottom product composition	x_B	mole fraction, %
Condenser level	M_D	m
Reboiler level	M_B	m
Pressure	p	kPa
Disturbances	Symbol	Unit
Feed rate	F	kmol/min
Composition rate	z_F	mole fraction, %

[1] Another, less common configuration is the "*DV*-configuration" where $\mathbf{u}_1 = [D \quad V]^T$ and $\mathbf{u}_2 = [L \quad B \quad V_T]^T$.

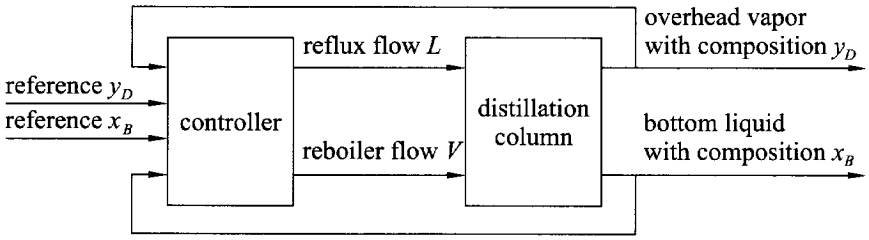


Figure 7.2. Distillation column LV control configuration.

When the column operates in a wide range, its characteristics are strongly nonlinear – especially towards high-purity, because of the physical saturation at 100% purity of the products (Fuentes and Luyben, 1983; Sriniwas et al., 1995). Apart from the nonlinearity, high-purity distillation columns exhibit the effect of *directionality* (or ill-conditioning), which means that the process gain depends on the direction of the input vector. (The direction of the input vector is determined as the ratio of the process inputs L and V.) The directions of the input vector which are least and most amplified are called the low-gain and high-gain input directions, respectively. The high-gain direction causes the top product to be purer, but, at the same time, the bottom product becomes less pure. The low-gain direction causes both products to become purer or less pure simultaneously. Since the objective is the dual composition control, the low-gain direction is the important direction for control (Skogestad and Morari, 1988).

7.2 The simulation model

The simulation model is implemented in MATLAB & Simulink[2] and has 39 stages, a reboiler and a condenser, see Fig. 7.1. The simulation model has been developed by Skogestad (1997)[3] under the assumption of a binary component, constant pressure, negligible vapor holdup, total condenser, equimolal flow, and vapor-liquid equilibrium at all stages. These assumptions cover the most important effects of the dynamics of a real distillation column. Throughout the identification and control experiments reported in this chapter, the product compositions are expressed as impurities, i.e., the mole fraction of the heavy component in the top product and of the light component in the bottom product.

The disturbances considered in the simulation model are the feed rate F and the feed composition z_F, modeled as uniformly distributed white-noise signals: $F \sim U([0.9, 1.0])$ and $z_F \sim U([0.45, 0.55])$. Such disturbance levels are common in an industrial environment (Skogestad, 1997). The feed-rate disturbance is assumed to be measurable, so its influence can be incorporated in the fuzzy model as an input that cannot be manipulated. The feed composition is typically a non-measurable quantity and its influence is not considered in the fuzzy model, leaving it as an unknown process disturbance.

[2]MATLAB and Simulink are registered trade marks of The Mathworks Inc., Natick, MA.
[3]It is available from the Internet at http://www.chembio.ntnu.no/users/skoge/distillation/.

7.3 Fuzzy modeling

Because of the phenomenon of directionality, open-loop identification only excites the process in the high-gain direction. As a consequence, it is hard to make both products purer at the same time, which corresponds to increasing the flows with L/V constant (low gain), see Section F.2. Hence, open-loop experiments do not give information about the low-gain direction, i.e., whether both products are simultaneously getting more pure or less pure (Fig. 7.3, top). To get both low-gain and high-gain directions well excited, we carried out a closed-loop experiment. Another reason for using closed-loop data generation is that the product compositions may not vary too much during the experiment, but sufficiently large variations are allowed on the reflux and reboiler flows. Based on the closed-loop experiments reported by Chou et al. (2000), identification data sets are generated in closed-loop experiments by using two separate (continuous-time) PI controllers. The controller parameters are given in Tab. 7.9. The reference signals of the controllers are designed to cover operating ranges of impurity $[0, 0.02]$ for both y_D and x_B (Fig. 7.3, bottom). Two experiments are carried out: (i) with constant nominal feed rate and feed composition, i.e., no disturbances are present, and (ii) with varying feed rate and feed composition, as described in Section 7.2. Part of the closed-loop generated data for the second case (with disturbances present) is shown in Fig. 7.4.

Two TS fuzzy models were constructed by using 8000 input-output data pairs sampled with the sampling time $T_s = 2\,\text{min}$. Since the process nonlinearity (gain directionality) is related to the composition concentrations, these variables are chosen as the antecedent variables. The domain of each of the compositions was manually partitioned into two fuzzy subsets, see Fig. 7.5. The respective rules describe the column operation in the four distinct cases: both products pure, both products impure, and one product pure and the other impure. For the disturbance-free case, the fuzzy model has the structure shown in (7.1) with consequent parameters given in Tab. 7.2 and Tab. 7.3, respectively. (When the disturbances are present, also the feed-rate disturbance is used as an input to the fuzzy model.) The cluster centers and membership functions are presented in Tab. 7.4 and Fig. 7.5, respectively. The standard deviations of the consequent parameters (see Appendix A for details) are given in Tab. 7.5 through Tab. 7.6. These deviations are used later in the MPC design to compute the bounds on the model-plant mismatch, which are necessary for the computation of the stability constraints on the control signals, recall Chapter 5.

$$
\begin{aligned}
&\mathcal{R}_{1i}: \textbf{If } y_{\mathrm{D}}(k-1) \text{ is } \mathcal{A}_{1i,1} \textbf{ and } x_{\mathrm{B}}(k-1) \text{ is } \mathcal{A}_{1i,2} \textbf{ then} \\
&\quad y_{\mathrm{D}}(k) = \zeta_{1i,1} y_{\mathrm{D}}(k-1) + \zeta_{1i,2} y_{\mathrm{D}}(k-2) + \zeta_{1i,3} x_{\mathrm{B}}(k-1) + \zeta_{1i,4} x_{\mathrm{B}}(k-2) + \\
&\qquad \eta_{1i,1} L(k-1) + \eta_{1i,2} L(k-2) + \eta_{1i,3} V(k-1) + \eta_{1i,4} V(k-2) + \theta_{1i} \\
&\mathcal{R}_{2i}: \textbf{If } y_{\mathrm{D}}(k-1) \text{ is } \mathcal{A}_{2i,1} \textbf{ and } x_{\mathrm{B}}(k-1) \text{ is } \mathcal{A}_{2i,2} \textbf{ then} \\
&\quad x_{\mathrm{B}}(k) = \zeta_{2i,1} y_{\mathrm{D}}(k-1) + \zeta_{2i,2} y_{\mathrm{D}}(k-2) + \zeta_{2i,3} x_{\mathrm{B}}(k-1) + \zeta_{2i,4} x_{\mathrm{B}}(k-2) + \\
&\qquad \eta_{2i,1} L(k-1) + \eta_{2i,2} L(k-2) + \eta_{2i,3} V(k-1) + \eta_{2i,4} V(k-2) + \theta_{2i} \\
&\quad i = 1, \ldots, 4
\end{aligned}
\tag{7.1}
$$

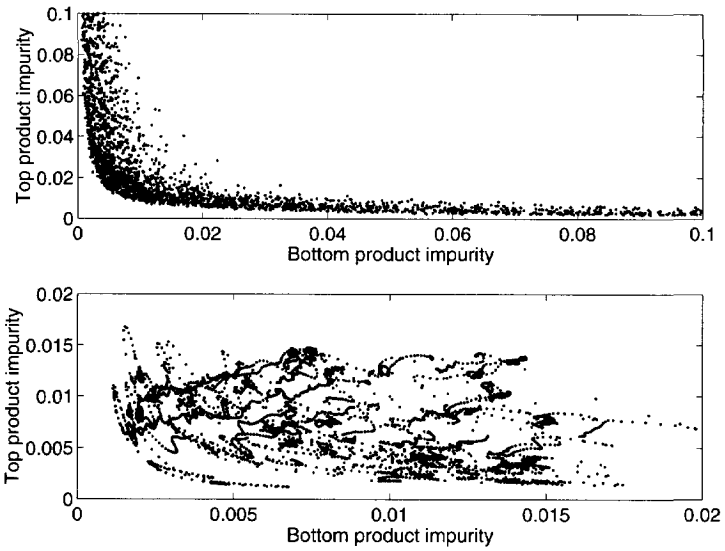


Figure 7.3. Distribution of open-loop (top) and closed-loop (bottom) identification data.

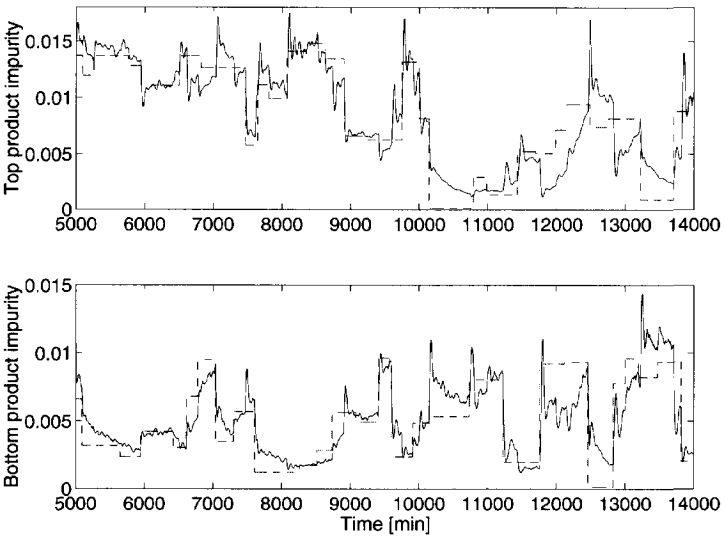


Figure 7.4. Closed-loop generated data. Solid line: process output, dashed line: reference.

Table 7.2. Consequent parameters for output *top composition impurity* (y_D).

$\mathcal{R}_i$	$\zeta_{1i,1}$	$\zeta_{1i,2}$	$\zeta_{1i,3}$	$\zeta_{1i,4}$	$\eta_{1i,1}$	$\eta_{1i,2}$	$\eta_{1i,3}$	$\eta_{1i,4}$	$\theta_{1,i}$
1	1.975	−0.976	0.0037	−0.0040	−0.0021	0.0017	0.0018	−0.0016	−0.0070
2	1.967	−0.969	−0.0543	0.0530	−0.0044	0.0049	0.0041	−0.0039	−0.001
3	1.937	−0.939	−0.0262	0.0258	−0.0045	0.0042	0.0041	−0.0039	−0.0030
4	1.980	−0.981	−0.0040	0.0035	−0.0018	0.0023	0.0017	−0.0019	−0.0002

Table 7.3. Consequent parameters for output *bottom composition impurity* (x_B).

$\mathcal{R}_i$	$\zeta_{2i,1}$	$\zeta_{2i,2}$	$\zeta_{2i,3}$	$\zeta_{2i,4}$	$\eta_{2i,1}$	$\eta_{2i,2}$	$\eta_{2i,3}$	$\eta_{2i,4}$	$\theta_{2,i}$
1	−0.2095	0.2134	1.9106	−0.9219	0.0034	−0.0032	−0.0066	0.0064	0.0118
2	−0.0133	0.0134	1.9067	−0.9071	0.0012	−0.0014	−0.0033	0.0032	0.0044
3	−0.0707	0.0725	1.8805	−0.8805	0.0029	−0.0025	−0.0075	0.0072	0.0057
4	−0.0244	0.0268	1.9437	−0.9420	0.0018	−0.0021	−0.0039	0.0037	0.0118

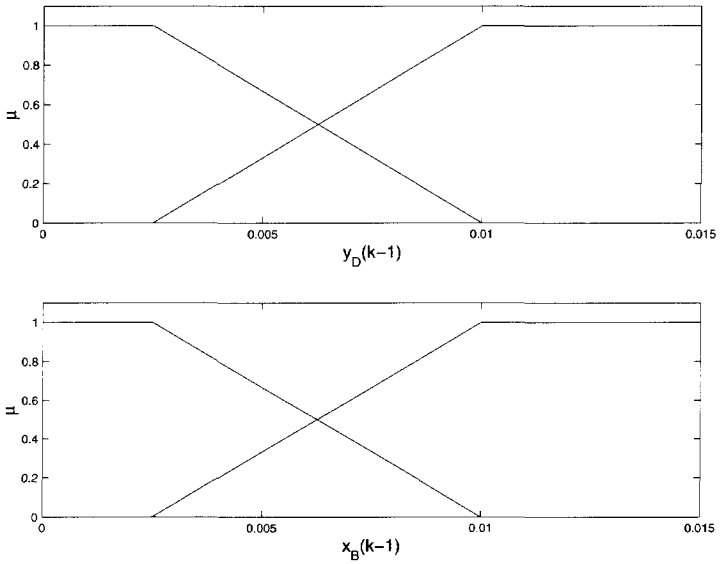


Figure 7.5. Membership functions for y_D and x_B.

The model performance on a validation data set is illustrated in Fig. 7.6. The performance is assessed by using the variance accounted for (VAF) and the root-mean-square error (RMS) indices. The results are RMS $= 0.0561$, VAF $= 96\%$ for top composition impurity (y_D) and RMS $= 0.0296$, VAF $= 99\%$ for bottom composition impurity (x_D), respectively.

Table 7.4. Cluster centers for rule $i = 1, \ldots, 4$, output $l = 1, 2$.

$\mathcal{R}_i$	$y_{\mathrm{D}}(k-1)$	$x_{\mathrm{B}}(k-1)$
1	0.0025	0.0025
2	0.0025	0.0125
3	0.0125	0.0025
4	0.0125	0.0125

Table 7.5. Standard deviations of the consequent parameters for output y_D, $\times 10^{-4}$.

$\mathcal{R}_i$	$\sigma_{\zeta_{1i,1}}$	$\sigma_{\zeta_{1i,2}}$	$\sigma_{\zeta_{1i,3}}$	$\sigma_{\zeta_{1i,4}}$	$\sigma_{\eta_{1i,1}}$	$\sigma_{\eta_{1i,2}}$	$\sigma_{\eta_{1i,3}}$	$\sigma_{\eta_{1i,4}}$	$\sigma_{\theta_{1,i}}$
1	20.22	20.10	14.82	14.66	0.17	0.24	0.15	0.19	0.08
2	23.56	23.43	9.91	9.81	0.17	0.23	0.15	0.20	0.07
3	22.06	21.96	23.17	22.85	0.22	0.30	0.17	0.23	0.10
4	20.70	20.58	10.09	10.01	0.17	0.22	0.15	0.19	0.07

Table 7.6. Standard deviations of the consequent parameters for output x_B, $\times 10^{-4}$.

$\mathcal{R}_i$	$\sigma_{\zeta_{1i,1}}$	$\sigma_{\zeta_{1i,2}}$	$\sigma_{\zeta_{1i,3}}$	$\sigma_{\zeta_{1i,4}}$	$\sigma_{\eta_{1i,1}}$	$\sigma_{\eta_{1i,2}}$	$\sigma_{\eta_{1i,3}}$	$\sigma_{\eta_{1i,4}}$	$\sigma_{\theta_{1,i}}$
1	47.98	47.63	32.52	32.15	0.36	0.52	0.33	0.42	0.18
2	80.48	79.93	31.04	30.70	0.47	0.77	0.42	0.65	0.30
3	75.79	75.32	30.96	30.66	0.52	0.70	0.48	0.61	0.23
4	32.15	32.00	26.98	26.67	0.29	0.40	0.23	0.31	0.13

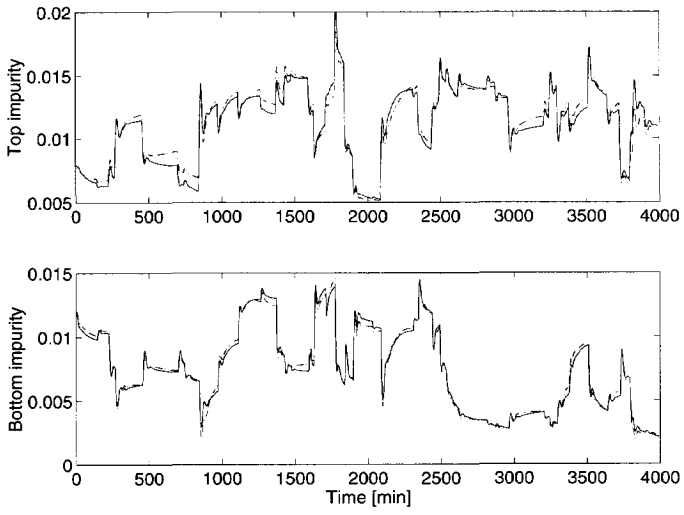


Figure 7.6. Fuzzy model validation. Solid line: validation data, dashed line: model prediction.

7.4 Analysis of interactions

Two methods have been recommended as suitable for quantifying the degree of directionality and the level of interactions in a distillation column: the condition number and the Relative Gain Array (RGA) (Skogestad and Postlethwaite, 1996). The reader interested in the condition number and its application to the distillation column problem should refer to the above reference, where the subject is treated in detail. In the remainder of this section, the fuzzy RGA method and the sensitivity analysis, proposed in Chapter 3, will be used to explore the degree of coupling (also called the input-output interactions) between the top and the bottom compositions (Mollov, Babuška and Verbruggen, 2001b).

7.4.1 RGA analysis

Since the antecedent structure of the fuzzy model (7.1) is identical for the two outputs, it is possible to start the analysis by computing the RGA for the individual rules. In this way a measure is obtained that shows how the interactions vary in the different operating regions.

Consider the linear model of the distillation column valid around $y_D = 0.0025$ and $x_B = 0.0025$, i.e., the first rule for each output of the model:

$$\begin{aligned} y_D(k) &= 1.975y_D(k-1) - 0.976y_D(k-2) + 0.0037x_B(k-1) - 0.0040x_B(k-2) \\ &\quad -0.0021L(k-1) + 0.0017L(k-2) + 0.0018V(k-1) - 0.0016V(k-2) - 0.007 \\ x_B(k) &= -0.2095y_D(k-1) + 0.213y_D(k-2) + 1.9106x_B(k-1) - 0.9219x_B(k-2) \\ &\quad +0.0034L(k-1) - 0.0032L(k-2) - 0.0066V(k-1) + 0.0064V(k-2) + 0.012 \end{aligned}$$

The offset term is omitted since it can be considered as an additional, constant input that does not influence the output change due to variations in the reflux and reboiler flows. This model is stable, with transfer matrix (recall (3.4))

$$\mathbf{G}_1 = \begin{pmatrix} -0.3077 & 0.1538 \\ 0.0769 & -0.0769 \end{pmatrix}$$

and the relative gain array is

$$\Lambda_1 = \mathbf{G}_1 .* \left(\mathbf{G}_1^{-1}\right)^T = \begin{pmatrix} 2.0000 & -1.0000 \\ -1.0000 & 2.0000 \end{pmatrix}.$$

If we compare Λ_1 with the Relative Gain Array obtained by using the white-box model of the column, linearized at this point

$$\Lambda_{lin} = \begin{pmatrix} 2.1500 & -1.1500 \\ -1.1500 & 2.1500 \end{pmatrix}, \tag{7.2}$$

we see that the corresponding elements of Λ_1 and Λ_{lin} are quite close. This can be seen as additional evidence of the steady-state model accuracy.

The RGAs for the four rules are shown in Tab. 7.7. Looking at the λs for the individual rules, one can see significant differences. This indicates varying degree of

coupling for the different compositions. Despite the variations for the separate rules, RGAs suggest pairing between reflux flow L and top composition y_D, and between reboiler flow V and bottom composition x_B. While this conclusion can also be derived by using only a single linear model, the fuzzy RGAs provide additional insight in the nature of interactions in the different operating regions. Negative λs in the RGAs for the first three rules indicate that the control loops counteract (i.e., try to compensate each other). For the last rule, the RGA values suggest that the interaction can lead to extra oscillations (recall Remark 3.2). As expected, the weakest interaction found is for the fourth rule, corresponding to the most impure compositions ($y_D = 0.0125$ and $x_B = 0.0125$). It is interesting, however, that the interaction indicated when the compositions are most pure ($y_D = 0.025$ and $x_B = 0.0025$) is not as strong as when the one is pure and the other impure (second and third fuzzy rules).

Table 7.7. RGA for rule $i = 1, \ldots, 4$.

rule	Λ_i	
1	2.0000	-1.0000
	-1.0000	2.0000
2	5.0000	-4.0000
	-4.0000	5.0000
3	9.0000	-8.0000
	-8.0000	9.0000
4	0.6250	0.3750
	0.3750	0.6250

To show why it is not possible to analyze the interactions in between the rules by using individual RGAs (3.5), let us analyze the evolution of the upper-left element of the RGA (λ_{11}) during a change in the active rule, e.g., from rule one to rule two. The transfer function matrix is a linear function of the degree of fulfillment β:

$$\begin{aligned}
\mathbf{G}_\beta &= (1-\beta)\cdot\mathbf{G}_1 + \beta\cdot\mathbf{G}_2 \\
&= (1-\beta)\begin{pmatrix} -0.3077 & 0.1538 \\ -0.0769 & 0.0769 \end{pmatrix} + \beta\begin{pmatrix} 0.1724 & 0.0690 \\ -0.6667 & -0.3333 \end{pmatrix} \\
&= \begin{pmatrix} -0.3077 + 0.1353\beta & 0.1538 - 0.0848\beta \\ -0.0769 - 0.5898\beta & -0.0769 - 0.2564\beta \end{pmatrix}. \qquad (7.3)
\end{aligned}$$

Applying formula (3.4) to $\mathbf{G}_\beta$, we obtain an RGA with an upper-left element

$$\lambda_{11,\beta} = -\frac{(0.4102\beta - 0.0769)(-0.3077 + 0.4801\beta)}{0.24695206\beta^2 - 0.24732835\beta - 0.0118}. \qquad (7.4)$$

The evolution of $\lambda_{11,\beta}$ when β changes from zero (rule 1 only active) to one (rule 2 only active) with a step of 0.001 is shown in Fig. 7.7 by a solid line. The indicated

values for $\lambda_{11,\beta}$ when a single rule is active coincide with the ones given in Tab. 7.7, however, in between $\lambda_{11,\beta}$ leaves the interval [2, 5] defined by the RGA elements for the individual rules.

For a comparison, consider the combined RGA (3.10) for $\beta = 0.9$. At this point the transfer matrix $\mathbf{G}_{\beta=0.9}$ is

$$\mathbf{G}_{\beta=0.9} = \begin{pmatrix} 0.1240 & 0.0775 \\ 0.4286 & 0.4286 \end{pmatrix} \tag{7.5}$$

which results in

$$\Lambda_{\beta=0.9} = \begin{pmatrix} 3.4427 & -2.4427 \\ -2.4427 & 3.4427 \end{pmatrix}. \tag{7.6}$$

This is what one would intuitively expect, as the elements in $\Lambda(\beta = 0.9)$ are in the intervals defined by the corresponding values of Λ_1 and Λ_2 (Tab. 7.7). The complete evolution of $\lambda_{11,\beta}$ is given in Fig. 7.7 by a dashed line.

From the performed analysis, a couple of guidelines for the control design can be given. First, the indicated degree of coupling varies for the different operating regions (fuzzy rules). The RGAs reveal that a decoupling design is required for the second and third region. In the first and the fourth regions, control with sufficient quality can be achieved using detached SISO control loops.

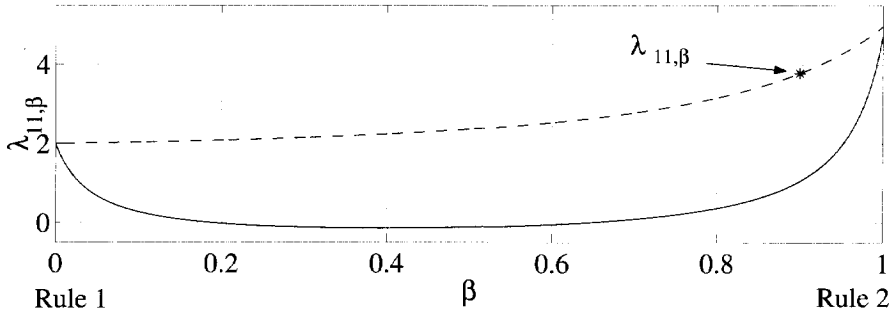


Figure 7.7. Evolution of $\lambda_{11,\beta}$ during a transition from the first to the second rule. Solid line: "individual RGA," Dashed line: "combined RGA".

7.4.2 Sensitivity analysis

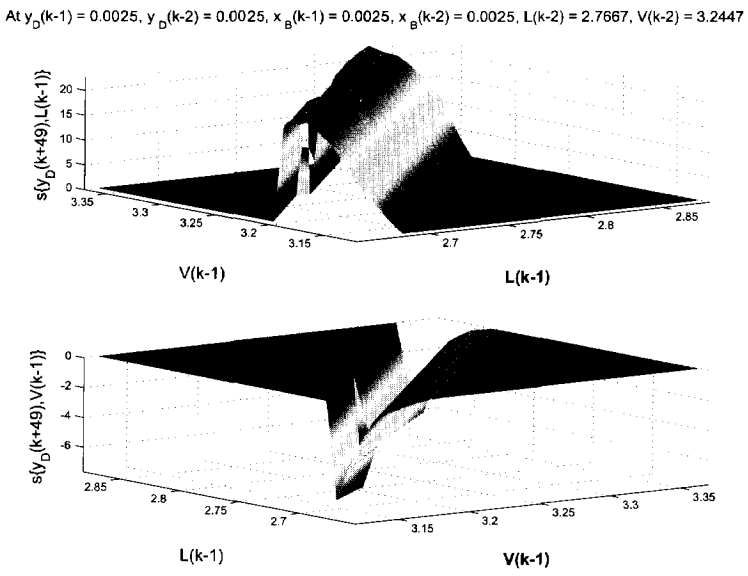
While the extended fuzzy RGA analysis provided an indication for the static interactions, the output sensitivity analysis can be used also to give insight in the asymmetry of the interactions and on the dynamic interactions. First the interactions for the linear models of the distillation column that correspond to the individual rules of the model (7.1) are analysed. At a steady state where a single rule is valid, a variation in either input is introduced and the sensitivity function (3.15) is computed based on the output response (Mollov, Babuška and Verbruggen, 2001b). The selected amplitude of 0.0345 corresponds to the input variations used to compensate for the disturbances

in the fuzzy MPC in a steady-state operation. The values of $s_{u_j}^{y_i}$ for the different rules are given in Tab. 7.8. Like in the previous section, for all rules the reflux flow L has a larger influence on the top composition impurity y_D than the reboiler flow V. Similarly, the bottom composition impurity x_B is affected more by V than by L. Comparing the RGAs (Tab. 7.7) with the sensitivity function (3.15) (Tab. 7.8) for the different rules, we can observe a close similarity. Both methods indicate that the interactions for compositions that are both pure (first rule) or impure (fourth rule) are weaker than those for a pure and an impure composition (second and third rules).

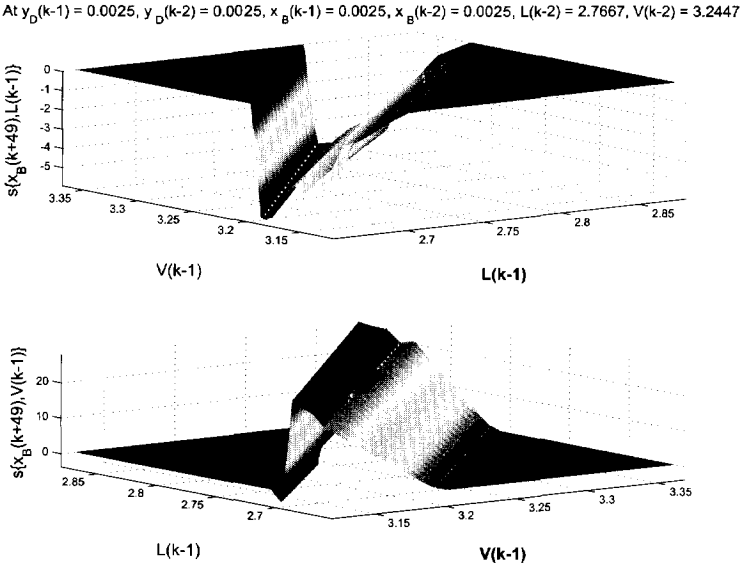
Table 7.8. Output Sensitivity Function for rule $i = 1, \ldots, 4$.

rule	$s_{u_j}^{y_i}$	L	V
1	y_D	19.92	-12.31
	x_B	-7.53	13.42
2	y_D	28.47	-22.47
	x_B	-16.77	25.84
3	y_D	39.65	-28.65
	x_B	-30.33	39.63
4	y_D	12.2	-5.43
	x_B	-5.35	10.3

We demonstrate how the amplitude of an input change influences the interactions with Fig. 7.8, which shows the static input-output interactions after a change in either input with the amplitude varying in $[0.0045, 0.0445]$. Note that since the sensitivity analysis is based on the complete fuzzy model rather than on the individual fuzzy rules, as the fuzzy RGA (recall Section 3.2.2), the fuzzy model has to be explicitly simulated to compute the output sensitivity with respect to a change in a certain input. The fuzzy model is simulated long enough (50 steps ahead) to reach a steady state. (The sensitivity functions are only depicted for a relatively narrow band of feasible combinations of L and V.) At the moment of the input variation, the process is in a steady-state operation. From the figure the following can be observed: (i) the influence of the reflux flow L on y_D is greater than the influence of the reboiler flow V (up to 25 and up to -7, respectively), see panels 1 through 2, and the influence of L on x_B is significantly lower than that of V (up to -6 and up to 22, respectively) see panels 3 through 4; (ii) there are differences between the influence of the L on y_D and that of V on x_B, as a result of the non-symmetric coupling (see also Tab. 7.8). It can be observed that in all four cases, the largest influence is achieved for a certain ratio of input variations. While this ratio is not constant for the different input-output combinations, in each of them it can be considered approximately linear (Fig. 7.8). The conclusion is that for the fastest transition between two operating regions, one should keep the input ratio constant while modifying the amplitudes, i.e., use the low-gain direction for control purposes.



(a) Top composition.



(b) Bottom composition.

Figure 7.8. Static coupling for 50-step ahead simulation of the model: (a) Top composition y_D, (b) Bottom composition x_B.

To give an impression of the dynamic coupling, Fig. 7.9 shows the series $s_{u_j(k)}^{y_i(k+\tau)}$, $\tau = 0, \ldots, 50$, at level $(y_D, x_B) = [0.0025, 0.0025]^T$ and $(L, V) = [2.7667, 3.2447]^T$ for the input change of 0.0345. A non-symmetric steady-state coupling is clearly visible, as the influence of V on y_D ($s_L^{x_B}$) is higher than that of L on x_B ($s_V^{y_D}$).

The sensitivity analysis thus provides additional insight for the control design. The two control loops have a non-symmetric influence: the top composition is more sensitive to the reflux flow than the bottom composition to the reboiler flow. Dynamically, the top composition reacts faster to changes in the reflux flow than to changes in the reboiler flow. Similarly, the bottom composition reacts faster to changes in the reflux flow than to the ones in the reflux flow. This can be explained by taking into account the positions where the two flows enter the column: the reflux flow is introduced at the top of the shell, while the reboiler flow is inserted at the bottom.

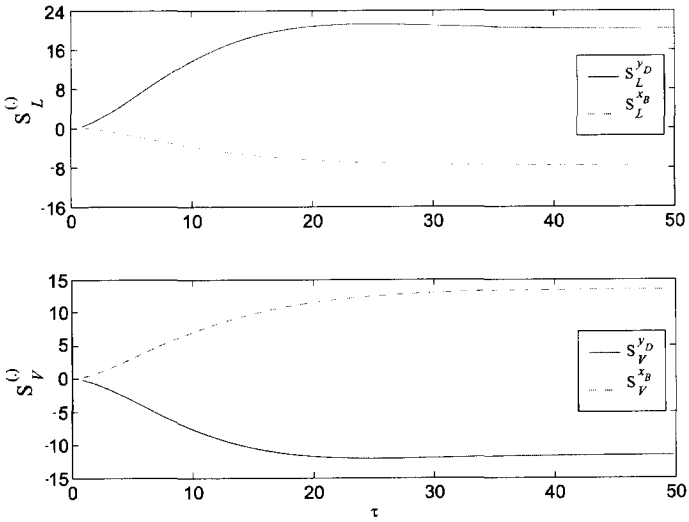


Figure 7.9. Dynamic coupling.

7.5 Decoupling control design

As mentioned in the introduction, the *LV*-configuration is preferable since the effect of the reflux and reboiler flows ($\mathbf{u}_1$) on the output compositions ($\mathbf{y}_1$) is nearly independent of the tuning of the level and pressure controllers (involving $\mathbf{u}_2$ and $\mathbf{y}_2$), as shown in (Skogestad and Postlethwaite, 1996). However, in this configuration $\mathbf{y}_1$ and $\mathbf{u}_1$ interact significantly (Section 7.4.1).

When using the *LV*-configuration for composition control, one should compensate for the interactions between the product compositions. Figure 7.10, left shows the control performance achieved by two detached PI controllers, proposed by Skogestad and Postlethwaite (1996). The PI parameters are given in Tab. 7.9. These controllers are tuned for fast reference tracking. As a result, however, there is a significant coupling after a change in either of the composition references. Different controller settings

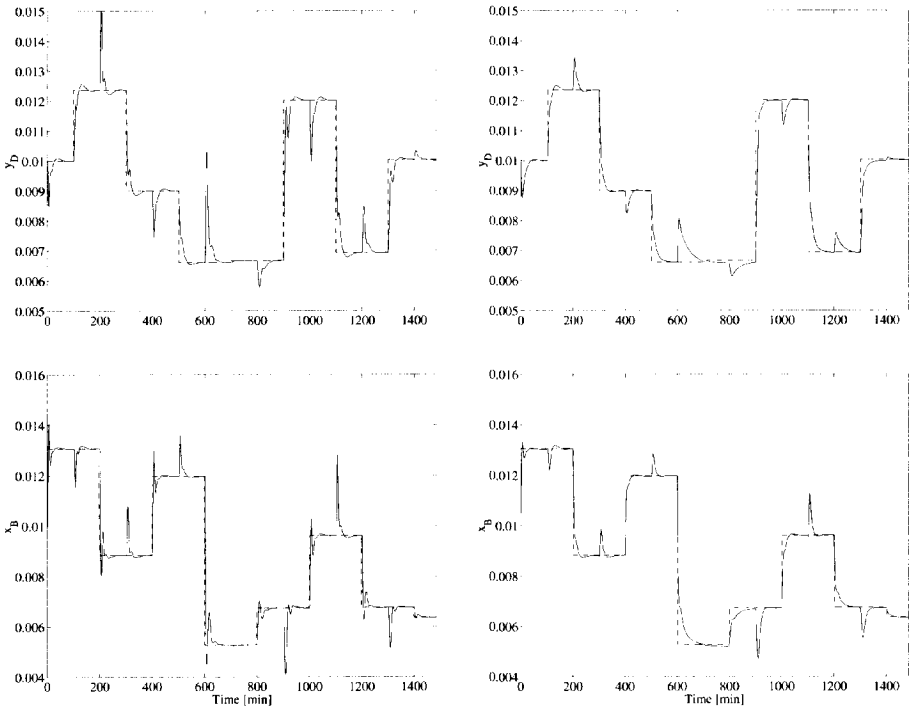


Figure 7.10. Distillation column: simulated control system performance. Left: Skogestad PI, right: PI & fuzzy decoupler. Top: top composition impurity y_D, bottom: bottom composition impurity x_B.

were proposed by Chou et al. (2000), see Tab. 7.9, to achieve a smooth rather than fast response (not shown in Fig. 7.10). Nevertheless, they cannot deal with the coupling.

To reduce the coupling, we apply the control scheme depicted in Fig. 3.5. The outputs of the detached PI controllers are fed to a time-varying decoupler based on the fuzzy model (7.1). Using the model, at each sampling instant the input-output gains are calculated and the corresponding transfer matrix is inverted and placed between the PI controllers and the system (Mollov and Babuška, 2002). Note that here, because of the discrete-time fuzzy model, discrete-time PI controllers are used (Åström and Wittenmark, 1997b). Since the decoupler changes the dynamics of the process, the discrete-time PI controllers are experimentally tuned with respect to the aggregated "fuzzy decoupler & process" system (Tab. 7.9).

The performance achieved with the decoupling control scheme is shown in Fig. 7.10 on the right. Comparing the results with the ones achieved without a decoupler shows that the improvement is significant. Although not completely removed, the coupling effects are significantly reduced. This also leads to an improvement in the overall performance, e.g., less overshoot during transitions. A quantitative measure of the

control performance is given in Tab. 7.10. The sum squared error (SSE) is used as a performance criterion, as lower values represent better control performance.

7.6 Fuzzy model predictive control

A model predictive controller has been designed as a typical example of a truly multivariable control technique (Mollov, Babuška, Abonyi and Verbruggen, 2002; Mollov and Babuška, 2002). The performance is compared with the PI controllers discussed above. Two different PI control configurations are shown (Tab. 7.9) because they are tuned with respect to different objectives. As a consequence, the difference in their performance is significant.

The predictive controller use the single model method in the optimization problem (Section 4.3.1). The MPC parameters are selected according to the tuning rules given in Section 5.1. Since there is a delay of one sample, the minimum cost horizon is set to $H_{\text{min}} = 1$. The process is well damped for the larger part of the operating range, therefore a prediction horizon $H_{\text{p}} = 6$ is used. Larger values of H_{p} would slow the response and reduce the disturbance rejection properties of the controller (the current disturbance is used throughout the prediction horizon). Although the process is of order four, a control horizon $H_{\text{c}} = 2$ rather than $H_{\text{c}} = 4$ is chosen as it makes the control less aggressive. The weighting matrices $\mathbf{P} = \mathbf{I}$ and $\Delta\mathbf{Q} = 0.025 \cdot \mathbf{I}$ are experimentally fine-tuned to achieve a reasonably fast response without an excessive overshoot. The input constraints are respectively

Table 7.9. PI controllers parameters. The last controller is a discrete-time one, with a sampling time of $T_{\text{s}} = 2\text{min}$.

Controller	$K_{L,P}$	$K_{L,I}$	$K_{V,P}$	$K_{V,I}$
PI1 (Skogestad and Postlethwaite, 1996)	26.1	6.94	−37.5	−11.329
PI2 (Chou et al., 2000)	8.1	0.69	−9.5	−0.772
PI & Fuzzy decoupler	20.25	1.24	−23.75	−1.39

Table 7.10. Relative performance of the different controllers, in %. Skogestad's PI controller performance is taken as 100%. Lower values represent better control performance.

Controller	SSE_{y_D}	SSE_{x_B}
PI1 (Skogestad and Postlethwaite, 1996)	100.0	100.0
PI2 (Chou et al., 2000)	98.2	95.4
PI & Fuzzy decoupler	62.2	75.4

$$L \in [1.7371,\ 3.7371] \qquad V \in [2.1536,\ 4.1536]$$
$$\Delta L \in [-0.2,\ 0.2] \qquad \Delta V \in [-0.2,\ 0.2].$$

The performance of the control system for the different controllers is compared for several set-point changes within the operating region (Fig. 7.11). The proportional & integral controllers PI1 (not shown in the figure) and PI2 have similar performance. Both predictive controllers perform superior to the PI controllers (Tab. 7.11). Note that to obtain a meaningful comparison, we use the same output references the PIs and the fuzzy MPC. The reference signal for the PI controllers is correspondingly shifted in order to anticipate a reference change.

Next, in order to show the influence of the model prediction on the control performance, we compare it with a fuzzy MPC that uses a fuzzy model that has as an input also the feed-rate disturbance F. The incorporation of the feed rate improves the model's prediction and thus the control system performance, see (Fig. 7.11, bottom). A quantitative performance measure is presented in Tab. 7.11. One can see that the presented fuzzy predictive controllers significantly improves the performance criteria.

Table 7.11. Relative performance of the different controllers, in %. Skogestad's PI controller performance is taken as 100%. The abbreviation FMPC+F indicates the MPC algorithm using a fuzzy model which has as an input the feed-rate disturbance.

Controller	SSE_{y_D}	SSE_{x_B}
PI1 (Skogestad and Postlethwaite, 1996)	100.0	100.0
PI2 (Chou et al., 2000)	102.3	97.9
FMPC	43.1	45.4
FMPC+F	39.4	41.2

7.6.1 Robust stability constraints

To illustrate the effect of the robust stability constraints on the control performance, we introduce the **C1**-constraints (Chapter 5). The gain of the feedforward filter R (Fig. 5.2) is equal to the inverse of the minimal steady-state gain (2nd option in Section 5.4.2). The parameter p in (5.30) and (5.31) is set to $p = 0.5$, which gives a first-order filter $\frac{0.5}{z-0.5}$.

The calculation of the uncertainty bounds at time $t = 33$ s is shown below. At this moment the column is at a steady-state operation, which is influenced by the disturbances on the feed rate and feed composition. The bounds ϵ_Ω and ϵ_{Ω_Δ} on the model uncertainty $\bar{\Omega}$ and $\bar{\Omega}_\Delta$ are computed according to (5.9) and (5.22), respectively. For the sake of illustration, the considered convolution operators in (5.10) and (5.23) are of third order.

At $t = 33$ s, i.e., at $k = 16$ the previous two values for the product compositions are $\big(y_D(k-1), y_D(k-2)\big) = [0.0101, 0.0101]$ mole fraction and $\big(x_B(k-1), x_B(k-$

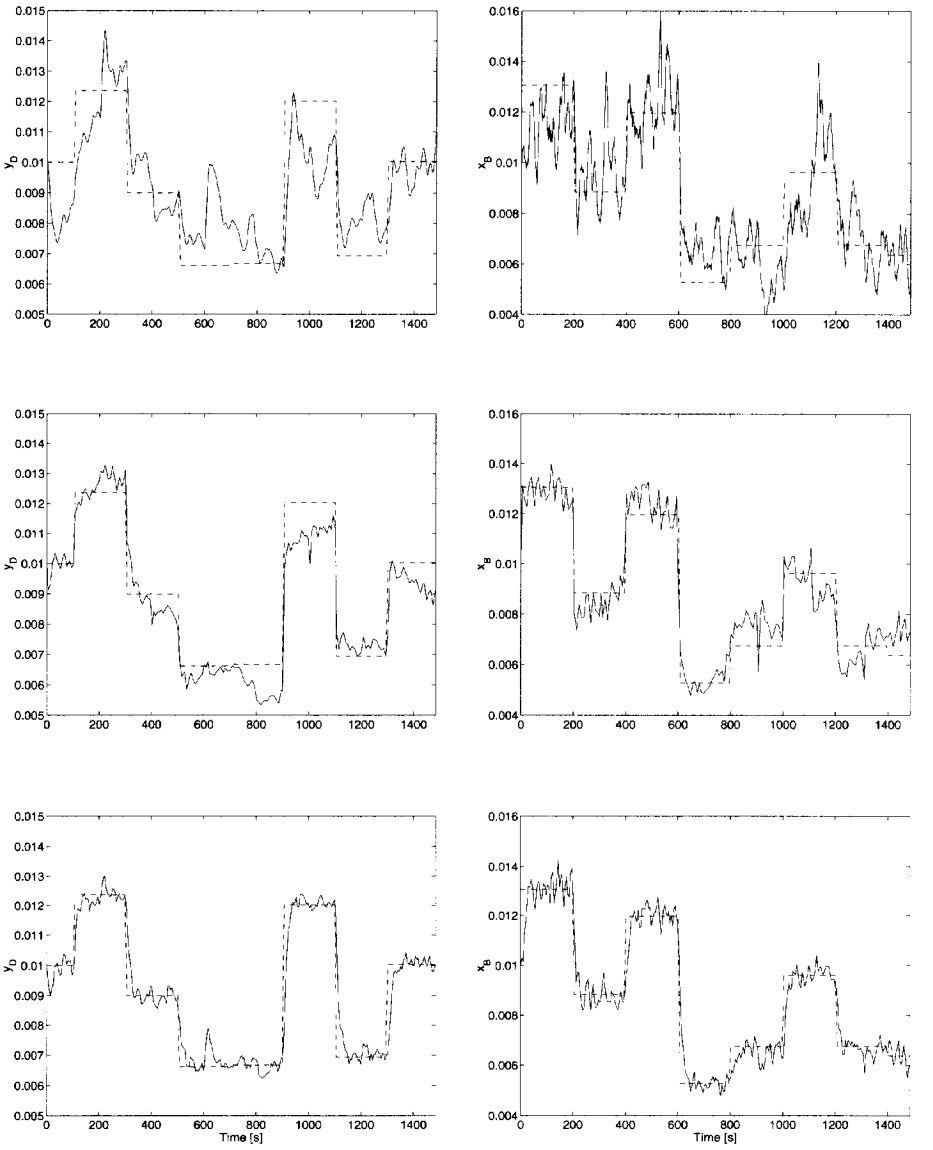


Figure 7.11. Simulated control system performance for varying feed rate and feed composition. Top: PI2 controller, middle: fuzzy predictive controller, bottom: fuzzy predictive controller with fuzzy model including the feed rate disturbance ($F \sim U([0.9, 1.0])$). Left: top composition impurity y_D, right: bottom composition impurity x_B.

$2)) = [0.0131, 0.0130]$ mole fraction; the control inputs are $(L(k-1), L(k-2)) = [2.5374, 2.5398]$ kmol/min and $(V(k-1), V(k-2)) = [3.0183, 3.0279]$ kmol/min. Note that although the output references do not change, the previous values for all the signals are different as a result of the feed-rate disturbance. The degrees of fulfillment for the individual rules of the first and the second output are $\beta_1 = [0.0052, 0.4389, 0.7616, 0.0024]$ and $\beta_2 = [0.05824, 0.14866, 0.7240, 0.8355]$, respectively. Based on these values, the matrices $\mathbf{A}_c(k)$ and $\mathbf{B}_c(k)$ extracted from the fuzzy model (5.16) are

$$\mathbf{A}_c(k) = \begin{pmatrix}
1.8 & -0.8 & 0.07 & -0.07 & 0.005 & -0.004 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.01 & 0.01 & 1.53 & -0.53 & 0.002 & 0.009 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.81 & -0.8 & 0.08 & -0.07 & 0.005 & -0.004 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.01 & 0.01 & 1.53 & -0.53 & 0.002 & 0.01 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

The corresponding matrices obtained in the previous two sampling instants $\mathbf{A}_c(k-1)$, $\mathbf{A}_c(k-2)$ and $\mathbf{B}_c(k)$, $\mathbf{B}_c(k-1)$ and $\mathbf{B}_c(k-2)$ are

$$\mathbf{A}_c(k-1) = \begin{pmatrix}
1.8 & -0.8 & 0.07 & -0.07 & 0.005 & -0.004 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.01 & 0.01 & 1.56 & -0.56 & 0.002 & 0.008 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.82 & -0.82 & 0.07 & -0.07 & 0.005 & -0.004 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.01 & 0.02 & 1.56 & -0.56 & 0.002 & 0.01 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$$\mathbf{A}_c(k-2) = \begin{pmatrix}
1.8 & -0.8 & 0.07 & -0.06 & 0.005 & -0.004 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.01 & 0.02 & 1.55 & -0.56 & 0.002 & 0.008 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.82 & -0.81 & 0.07 & -0.07 & 0.005 & -0.004 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.01 & 0.02 & 1.56 & -0.55 & 0.001 & 0.01 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

and

$$\mathbf{B}_c(k)=\begin{pmatrix} -0.007 & 0.006 \\ 0 & 0 \\ 0.003 & -0.014 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \\ -0.007 & 0.006 \\ 0 & 0 \\ 0.003 & -0.014 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \end{pmatrix} \mathbf{B}_c(k-1)=\begin{pmatrix} -0.007 & 0.006 \\ 0 & 0 \\ 0.004 & -0.013 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \\ -0.007 & 0.006 \\ 0 & 0 \\ 0.004 & -0.013 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \end{pmatrix} \mathbf{B}_c(k-2)=\begin{pmatrix} -0.006 & 0.006 \\ 0 & 0 \\ 0.004 & -0.014 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \\ -0.006 & 0.006 \\ 0 & 0 \\ 0.004 & -0.014 \\ 0 & 0 \\ 1.000 & 0 \\ 0 & 1.000 \end{pmatrix}$$

The common matrix $\mathbf{C}_c$ is

$$\mathbf{C}_c=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}.$$

From (5.19), for the kernels $\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$, $i=0,\ldots,3$ we have

$$|\mathbf{H}_{\delta,\mathrm{tv}}(k,i)|=\begin{cases} i=0 & : \quad |\mathbf{H}_{\delta,\mathrm{tv}}(k,0)|=0 \\ i=1 & : \quad |\mathbf{H}_{\delta,\mathrm{tv}}(k,1)|\le 0.0085 \\ i=2 & : \quad |\mathbf{H}_{\delta,\mathrm{tv}}(k,2)|\le 0.0236 \\ i=3 & : \quad |\mathbf{H}_{\delta,\mathrm{tv}}(k,3)|\le 0.0288 \end{cases} \tag{7.7}$$

and by using (5.9) through (5.11), we obtain $\epsilon_\Omega(k)=0.0609$. The resulting stability constraints (max) on the non-incremental signals are $L^{\max}=3.224$ and for $V^{\max}=3.66$.

Analogously, for the incremental kernels $\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,i)$, $i=0,\ldots,3$ from (5.28) we have

$$|\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,i)|=\begin{cases} i=0 & : \quad |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,0)|=0 \\ i=1 & : \quad |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,1)|\le 0.0085 \\ i=2 & : \quad |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,2)|\le 0.0088 \\ i=3 & : \quad |\Delta\mathbf{H}_{\delta,\mathrm{tv}}(k,3)|\le 0.0131 \end{cases} \tag{7.8}$$

and by using (5.9) through (5.11), we obtain the upper bound of the uncertainty $\epsilon_{\Omega_\Delta}(k)=0.0304$. The stability constraints (max) on the incremental signals are $\Delta L^{\max}=0.1393$ and $\Delta V^{\max}=0.1174$, respectively.

The robust stability constraints on L and V, non-incremental and incremental, for the first 300 minutes are given in Fig. 7.12, middle and bottom panels, respectively. Since $L_{\min}$ and $L^{\max}$ (and $V_{\min}$ and $V^{\max}$) are symmetric around zero (Theorem 5.4 and Corollary 5.1), only $L^{\max}$ and $V^{\max}$ are shown in Fig. 7.12, middle.

In the beginning, the stability constraints on the incremental signals are rather loose as a consequence of the large model-plant mismatch. Between the 100th and 200th second, the model-plant mismatch is limited and thus the constraints are much tighter. The non-incremental constraints are smoother in this period as well. During the next 100 seconds the model-plant mismatch becomes larger, which is again reflected in the stability constraints: the ones imposed on the incremental signals become less tight and the ones imposed on the non-incremental signals fluctuate more.

The application of the stability constraints, however, has a significant drawback: its complexity and, as a consequence, the computation time. As shown in Chapter 5, the computation of these constraints requires an additional control loop, with the corresponding memory and processing time requirements.

Table 7.12. Relative performance of the different controllers, in %. Skogestad's PI controller performance is taken as 100%, see Tab. 7.9.

Controller	SSE_{y_D}	SSE_{x_B}
FMPC	43.1	45.4
FMPC& Robust Stability constraints	40.3	41.1

7.6.2 Comparison with an MPC algorithm based on Wiener models

The results presented in this chapter are comparable to the results reported by Bloemen et al. (2001), who use a nonlinear Wiener-type prediction model in two MPC algorithms.

Wiener models consist of a linear dynamic part followed in series by a static nonlinear element. To be useful for control purposes, however, the nonlinearity must be invertible. Although any form of nonlinear function can be used, usually a polynomial model is employed.

The authors identify a Wiener model by using again the closed-loop data generation experiment proposed by Chou et al. (2000). The output nonlinearity is described by two third-order univariate polynomials

$$\mathbf{y}_m = \mathbf{h}(\mathbf{w}) = \mathbf{H}(\mathbf{w})\mathbf{w} = \begin{bmatrix} H_{11}(w_1) & 0 \\ 0 & H_{22}(w_2) \end{bmatrix} \mathbf{w},$$

where H_{11} and H_{22} are second-order polynomials and $\mathbf{w}$ is the output of the linear dynamics. By restricting the operation region for $\mathbf{w}$ (since the model is only valid within the region of the identification data), it is possible to compute the lower and upper bounds on H_{11} and H_{22}. All possible values of H_{11} and H_{22} are within the convex hull defined by these lower and upper bounds.

The cost function used in the MPC algorithm is similar to the one presented in Chapter 4

$$J(k) = \sum_{i=1}^{\infty} \mathbf{y}_m(k+i+1|k)^T \mathbf{P} \mathbf{y}_m(k+i+1|k) + \Delta\mathbf{u}(k+i|k)^T \Delta\mathbf{Q}\mathbf{u}(k+i|k) \quad (7.9)$$

under the assumption of $(\mathbf{y}_{m,\text{ref}}, \Delta\mathbf{u}_{\text{ref}}) = (0, 0)$. When this assumption does not hold, the origin of the system can be shifted respectively.

To avoid the infinite number of degrees of freedom arising from the infinite prediction and control horizons, the above cost function is split into two parts, in a way

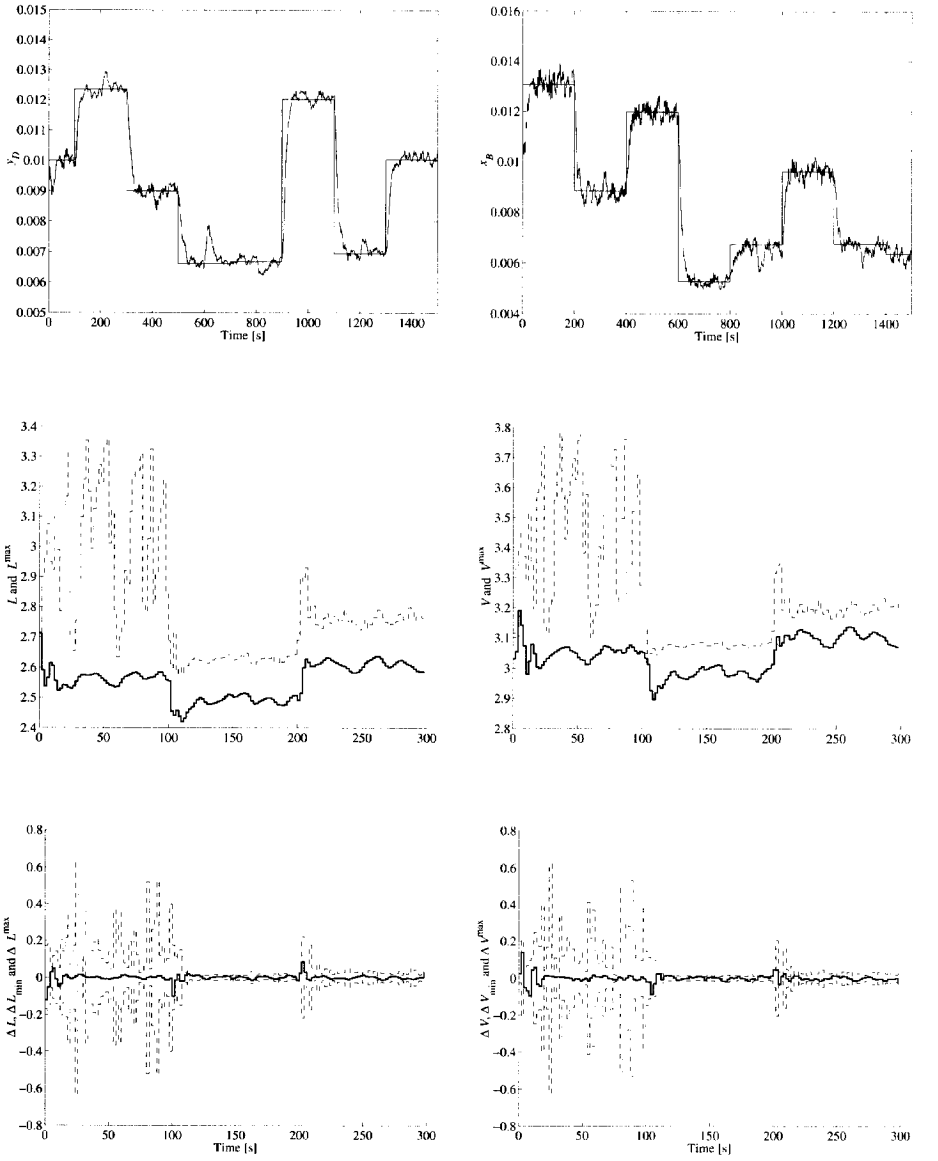


Figure 7.12. Simulated control system performance with robust stability constraints imposed. Top: control system performance. Left: top composition impurity y_D, right: bottom composition impurity x_B. Middle: constraints on the non-incremental input signals, bottom: constraints on the incremental input signals. Left: reflux flow L, right: reboiler flow V.

that transforms the minimization of $J(k)$ into a linear cost function, subject to matrix inequalities which are affine in the nonlinearity $\mathbf{H}(\mathbf{w})$.

In the first MPC algorithm, the nonlinearity of the Wiener model is transformed into a polytopic description. Through the use of the polytopic description in the cost function (7.9), these nonlinear matrix inequalities are satisfied when they hold for every vertex of the polytopic description. Thus the minimization problem then leads to a convex, LMI-based, optimization problem.

In the second MPC algorithm, the inverted nonlinearity is used in the control scheme to compensate for the nonlinearity (Norquay et al., 1998) and thus to remove it from the optimization problem. The resulting convex optimization problem can be stated as a quadratic programme plus a terminal cost term.

In Bloemen et al. (2001), the performance achieved through Wiener MPC is compared to the performance achieved by using (i) a finite impulse response (FIR) model, based on open-loop data generation, and (ii) a linear state-space model based on closed-loop data generation. The MPC algorithm using the FIR model gives poor performance, because the low-gain direction of the process is not captured in the model. The linear state-space model captures the low-gain direction much better, and improves the system performance. However, the linear model is not able to capture the nonlinearity of the process, which leads to a model-plant mismatch at certain regions. The Wiener model can straightforwardly handle this nonlinearity, which improves the model prediction and thus the control system performance.

In the following, we compare the MPC algorithms based on a Wiener model with the fuzzy MPC algorithms used in this chapter in terms of generality of the method, stability properties and computational complexity.

The MPC algorithms using the Wiener model are developed with respect to the specific structure of these models – linear dynamics followed by a static *invertible* nonlinearity. Moreover, the algorithm based on polytopic description of the nonlinearity relies on a certain assumption about the nonlinearity. For general nonlinearities, the number of LMIs rapidly increases. The algorithm based on the inversion of nonlinearity can be applied only when the model nonlinearity is invertible.

From this point of view, the fuzzy MPC algorithms presented in this thesis are more general, as they do not rely on a certain structure of the nonlinearity. Moreover, the robust stability constraints are not even restricted to TS models and can be applied to an arbitrary nonlinear model within a certain class. Additionally, when these constraints are used, stability is guaranteed also during a transient between two operating regions, which is not the case with the LMI-based MPC algorithm. However, the application of the stability constraints has a significant drawback: its on-line complexity and, as a consequence, the computational time.

7.7 Summary and concluding remarks

In this chapter, the techniques for analysis and control design based on TS fuzzy models have been applied to a simulation model of a distillation column. This benchmark is well-known for its nonlinear behaviour and directionality character, typical for most distillation columns used in the industry.

The need of closed-loop identification that can identify the low-gain direction was motivated and the corresponding experiment was presented. Based on the collected data, fuzzy models of the TS type were identified and used (i) in the analysis of input-output interactions and input-output decoupling, and (ii) in the design of fuzzy MPC algorithms.

The input-output analysis of the fuzzy model, based both on RGA and output sensitivity functions, indicated that the input-output coupling varies for the different operating regions. To reduce this coupling, we constructed a decoupler that uses the fuzzy model. It compensates for the coupling existing in the process by inverting the model on-line, at each sampling instant. It was demonstrated that in this way the coupling can be significantly reduced.

Thereafter the fuzzy model is utilized in the fuzzy MPC algorithm proposed in Chapter 4. We have showed that the performance of the resulting control system is significantly better than that of detached PI controllers. As a next step, it was demonstrated how the MPC performance can be improved by improving the model prediction. This is done by including disturbances as inputs in the fuzzy model. Finally the effect of the robust stability constraints on the control performance was discussed. The stability constraints were computed and linked to the model-plant mismatch. It was demonstrated that the stability constraints provide additional smoothness of the control signal, which results in a better control performance. However, the application of the stability constraints has a significant drawback: its complexity and consequently the computational time.

The results presented in this chapter are comparable to those of MPC algorithms using a nonlinear Wiener-type prediction model. Both of them are superior to the ones obtained by MPC using a FIR model based on open-loop data generation, and to those obtained by MPC using a linear state-space model based on closed-loop data generation. In terms of the method's generality and stability properties, the fuzzy MPC algorithm appears to be more general, as its computational complexity does not depend on the type of model nonlinearity.

8 CONCLUSIONS AND SUGGESTIONS

This thesis addresses various aspects of multivariable process control using fuzzy models and fuzzy model-based control. Several algorithms are proposed that have a sound theoretical background and a potential for industrial-scale applications. The purpose of this final chapter is to summarize the main ideas and results, to present the general conclusions of the research described in this thesis, and finally, to give recommendations for future research directions in this field.

8.1 Motivation and preliminaries

During the past decades, fuzzy control has emerged as one of the most active areas of application of fuzzy set theory and fuzzy logic. Fuzzy control systems have been recognized as an appealing alternative to classical control schemes when partially known, nonlinear or multivariable processes are addressed. Their structure facilitates the combination of qualitative knowledge and input-output process data. Still, the design of fuzzy control systems is a time-consuming task involving knowledge acquisition and parameter tuning, especially for MIMO control applications. There have been no generic solutions for the design of stable fuzzy logic controllers that take into account the analysis of the interactions (static and dynamic) among the inputs and outputs, and that allow for parameter tuning for nominal or robust performance.

Chapter 1 gives a short overview of the MIMO aspects in the process control and introduces the fuzzy systems in the context of multivariable process control. It also states the primary motivation for the research carried out, namely that the fuzzy approach should offer a *user-friendly* way of designing *nonlinear* controllers for multivariable processes.

Chapter 2 contains background material about the local linear approach used in the developed methods for fuzzy analysis and control design. The chapter introduces the Takagi–Sugeno (TS) type fuzzy models used throughout the thesis. If local models are available for different operating regions, controllers corresponding to them can be found in advance. This local approach appears to be well suited for the design of fuzzy state feedback and output feedback controllers and fuzzy observers. Still, there are issues of different character that limit the applications of this approach both in conceptual and implementation terms, as shown in Chapter 2. In the consequent chapters, different paradigms are used for the development of methods for fuzzy control design. As a result, some of those shortcomings can be avoided or solved.

8.2 Extensions and novel results

The methods for fuzzy analysis and control design presented in the thesis are ordered, according to the author's opinion, from least to most important with respect to their level of novelty and contribution to already existing techniques.

8.2.1 RGA for TS fuzzy models

In Chapter 3, an extension of the Relative Gain Array (RGA), which was originally introduced for linear systems, was presented to facilitate the analysis of interactions in MIMO TS fuzzy models. The TS structure is analyzed in order to obtain an RGA per rule that indicates the interactions present in the region where the rule is valid. When all MISO submodels have the same antecedent structure, RGAs can be computed for the individual rules. These RGAs, however, cannot provide information about the interaction when two or more rules are active, or when the antecedent structure differs per output. To obtain that information for a specific point when two or more rules are active, we derive first the model valid around that point. This method can only be applied to analyse static input-output interactions. Therefore, to have a meaningful

interpretation of the RGA values, it is necessary that the fuzzy model under consideration is a stable one. Another drawback of the method is that for 2×2 models, the interactions are always indicated as being symmetrical.

8.2.2 Output sensitivity analysis

Another tool to analyse interactions is the output sensitivity function introduced in Chapter 3. It measures whether and to what extend a process output depends on variations in one or more of its inputs. The sensitivity function is computed as partial derivative of the output with respect to a given input, while the remaining inputs are kept constant. The sensitivity analysis gives additional information for the control design. Unlike the RGA, the sensitivity functions can indicate a non-symmetrical coupling between the control loops even for 2×2 models. Further, not only static but also dynamic coupling can be obtained to characterize the intermediate effects that an input change has on the considered output.

8.2.3 Input-output decoupling

The insight obtained by the RGA and sensitivity analysis can be used in the design of a MIMO control system. While within an operation region with weak interactions, SISO controllers may suffice, for regions where strong coupling is present, a decoupler should be designed. In Chapter 3, three methods were discussed that achieve such decoupling, by taking advantage of the structure of the TS fuzzy models. All of them essentially invert the model; the differences are in the means used to achieve this inversion. If the fuzzy model is an affine one with at least as many inputs as outputs, an analytical decoupling law can be obtained. For non-affine fuzzy models, the inversion can be accomplished either by inverting the input-output transfer function matrix or by solving a nonlinear optimization problem. The computational complexity varies for the different decoupling methods. The decoupling of affine TS models can computed very efficiently, allowing applications when short sampling times are necessary. The same holds for the analytic decoupling method for the non-affine TS model. The numerical decoupling is computationally demanding, which limits its application to fast processes. However, it is the most general method that can be applied regardless of the model structure.

8.2.4 Fuzzy model predictive control

TS fuzzy models proved to be suitable for use in nonlinear MPCs because of their ability to accurately approximate complex nonlinear systems. With a nonlinear prediction model, however, the optimization problem is a non-convex one, requiring time-consuming iterative optimization methods with no guarantee of reaching the global minimum. The approach to fuzzy MPC presented in Chapter 4 is based on linear time-varying prediction models derived by the fuzzy model at a given operating point or along a pre-computed trajectory. The optimization problem obtained in this way is convex and can be effectively solved. To account for errors introduced by the linearization, we propose an iterative optimization scheme where the QP solution pro-

vides a search direction toward the minimum of the optimization problem. Essentially, the algorithm is a specific implementation of the Sequential Quadratic Programming method, based on the structure used in linear MPC to predict the process behavior. Its advantage is its generic form that does not depend on the particular structure of the fuzzy model used, as opposed to that of, e.g., the feedback linearization techniques. Moreover, because of the fast convergence rate, which is due to the straightforward manner in which the optimization problem is formulated, the algorithm can be applied to nonlinear multivariable processes with a short sampling time. For such processes, linear MPC does not give satisfactory results, while the optimization which directly applies the nonlinear TS model is time consuming.

Compared to already reported applications of MPC using fuzzy prediction models, the algorithms presented in this thesis have several advantages. First, they can be directly applied to MIMO processes. Second, the input and output constraints are explicitly considered during optimization. Last but not least, the achieved performance is near to the one that is achieved by using the nonlinear fuzzy model directly and thus nonlinear optimization techniques, while at the same time the computational complexity is comparable to that of the linear MPC.

8.2.5 Robust stability constraints for fuzzy model predictive control

In Chapter 5, a general scheme for computing robust stability constraints for nonlinear MPC is presented. Constraints on the control signals and their increments are calculated that guarantee stability for any model-plant mismatch within given uncertainty bounds for general nonlinear plants. Next, a systematic procedure for the computation of the bounds on the model uncertainty is proposed when the process model is a TS fuzzy model. To compute the uncertainty bounds, the model-plant mismatch is represented as a linear time-varying convolution operator.

It is shown that the stability constraints robustify the system performance in the presence of a model-plant mismatch without deteriorating the nominal performance. The latter is achieved by an additional feedforward filter that accounts for the process gain. The stability constraints, rather than the weights in the cost function, smooth out the control signal. The application of the stability constraints, however, has a significant drawback – it leads to complexity and, as a consequence, to a considerable computational effort.

8.2.6 Fuzzy control of a High-Purity Distillation Column

Chapter 7 demonstrates the advantages of using fuzzy modeling and control design techniques on a simulation model of a distillation column. This benchmark is well-known by its nonlinear behaviour and directionality character, typical for most distillation columns used in the industry. Because of the directionality phenomena, a closed-loop identification experiment was necessary. Based on the collected data, TS fuzzy models have been identified and used in the analysis of input-output interactions and in the control design. A decentralized control structure with a decoupler and a multivariable controller based on fuzzy MPC were presented.

The input-output analysis carried out on the fuzzy model indicated that the input-output coupling varies for the different operating regions. To reduce this coupling, we redesigned a decoupler that inverts the model on-line, at each sampling instant, which reduces the coupling significantly. Next, the fuzzy model was utilized in the fuzzy MPC. The resulting performance is significantly better than the one achieved by means of decentralized controllers. The performance of the MPC controller is also ameliorated by including a disturbance as an input in the fuzzy model, improving the model prediction. Additional improvement is gained by using the robust stability constraints.

The presented results are comparable to the results achieved through MPC algorithms using a nonlinear Wiener-type prediction model. These results are superior to the ones obtained by linear MPC using a FIR model based on open-loop data generation, and by MPC using a linear state-space model based on closed-loop data generation. In terms of the generality and stability properties, the fuzzy MPC algorithm appears to be more general, as its computational complexity does not depend on the type of model nonlinearity. Moreover, the stability of the fuzzy MPC-based control system is guaranteed in the presence of model-plant mismatch, while with the Wiener MPC algorithms only the nominal stability is guaranteed. The application of the stability constraints, however, considerably increases the computational effort.

8.2.7 Fuzzy model predictive control of a GDI engine

A real-world application of fuzzy model predictive control to a gasoline direct injection (GDI) engine was presented in Chapter 6. GDI engines can operate in two different combustion modes: homogeneous and stratified. The GDI engine is a highly nonlinear, multi-variable system with four inputs, three states and six outputs. The nonlinearity results from the combustion efficiency and generated torque in the different combustion modes as well as from the flow relations and saturation effects in each mode. An additional complication is the switching between the combustion modes. Then the engine parameters have to be properly adapted to allow efficient operation in either mode.

The Engine Management System controllers are normally designed by using engine maps, i.e., look-up tables derived through extensive experiments with an engine prototype. The application of MPC to engine control is a new approach, which can potentially overcome the shortcomings of the traditional control strategies applied to increasingly sophisticated engines. The obtained results demonstrate the potential of fuzzy model predictive control for such a complicated system. Generally, they are comparable to the results achieved by using first-principle and technological knowledge, as the remaining problem is the control of the torque gradients during a switch in the combustion mode. As the control design follows a relatively standard procedure and does not require detailed information about the engine, it can reduce the time and cost of the control design and tuning phase of new engines to be developed.

8.3 Suggestions for future research

The set of tools and methods for fuzzy control design presented in this thesis is not complete or fixed. Based on the obtained results and the possible applications, we recommend a couple of "starting points" for future research, which are outlined below.

8.3.1 Analysis of dynamic interactions using the Relative Gain Array

The RGA was proposed as a technique that allows for the analysis of the input-output interactions, both static and dynamic. The majority of the results published so far have been directed to continuous-time domain systems. In Chapter 3, a modification has been proposed that addresses these interactions in discrete-time fuzzy models. While the extension to the static case is relatively straightforward, a question arises when dynamic interactions are to be analysed. A problem still to be investigated is the relation between the sampling frequency of the model and the interactions of the system at a given frequency, i.e., how the indicated dynamic interactions are influenced as a function of the frequency range of the model.

8.3.2 Robust stability analysis through sensitivity analysis

Recently much effort has been focused on the robustness and stability analysis of TS fuzzy control systems. However, few attempts have been made to incorporate the sensitivity analysis (Chapter 3) into the robustness analysis, as most effort has been concentrated on stability margins without providing bounds on the input-output interactions. These should be addressed when one discusses stability and robustness issues of fuzzy systems. With the output sensitivity function, it might be possible to formulate an energy-like function that describes the evolution of the system outputs in terms of a quadratic Lyapunov function. While in the continuous-time case such a quadratic Lyapunov function can be derived relatively easily as shown by Farinwata (2000), for discrete-time systems no such a formulation is available yet.

8.3.3 Speed-up of fuzzy model predictive control

The optimization problem in fuzzy MPC (Chapter 4) uses a QP subproblem based on the gradient and Hessian of the corresponding Lagrange function. As a result of the specific structure of the optimization problem, in many situations the Hessian is sparse. Then significant speed can be gained if first an SVD decomposition is performed and thus the dimension of the optimization problem is correspondingly reduced.

8.3.4 Design of max-min-plus and max-plus-linear predictive controllers using fuzzy models

Although in MPC usually linear discrete-time models are applied, it is possible to extend MPC to a class of discrete-event systems, as shown in (De Schutter and van den Boom, 2000a) where an MPC framework for max-plus-linear systems is presented. The term *max-plus-linear* means that the basic operations in these systems are maximization and addition. In general, the resulting optimization problem is nonlin-

ear and non-convex. However, if the control objective and the constraints depend monotonously on the outputs of the process, the MPC problem can be recast as a problem with a convex feasible set. In addition, if the objective function is convex, then the resulting optimization problem is also convex. This, in turn, would provide an optimal solution to the exact, not the approximated problem.

The *max-min-plus* systems are extensions of the max-plus-linear systems that can be described by equations in which the operations maximization, minimization and addition appear (Olsder, 1994; Gunawardena, 1994). As for the max-plus-linear case, here also the resulting optimization problem is, in general, nonlinear and non-convex (De Schutter and van den Boom, 2000b). However, if the state equations are decoupled and if the control objective and the constraints depend monotonously on the states and outputs of system, the max-min-plus MPC problem can be recast as a problem with a convex feasible set. As before, if the objective function is convex, then the optimization problem is also convex.

The TS fuzzy models seem to be very suitable for use in min-max and max-plus-linear predictive controllers, as the inference mechanism can easily be modified to match both types of systems. By replacing the product operator in the fuzzy inference mechanism by max-min or max-plus-linear operators, we obtain fuzzy min-max or max-plus-linear models, respectively, that can be applied in the corresponding min-max or max-plus-linear predictive controller.

8.3.5 Design of analytic constrained predictive controllers using fuzzy models

As stated in Chapter 4, one of the biggest advantages of MPC is that it can effectively deal with constraints; disadvantage is that for every time step a computationally expensive optimization problem has to be solved. The time required for the optimization makes MPC not always suitable for fast systems or complex problems. A viable possibility is to avoid the on-line optimization via off-line approximating the input-output mapping of the MPC controller by means of fuzzy models.

Recently, it was shown that the solution to the predictive control problem is a continuous function of the output, the reference signal, the noise and the disturbances (Hoekstra et al., 2001). Therefore, instead of performing an optimization at every time-instant, one can use a fuzzy model to provide the control signal. The resulting controller would allow a faster implementation compared to the traditional optimization procedure and the approach is applicable to complex problems with many constraints.

There have been attempts to utilize a feed-forward neural network to compute on-line the control move, see (Parsini and Zopolli, 1995; Pottmann and Seborg, 1997; Hoekstra et al., 2001). A disadvantage is that in this setting, the controller performs an input-output mapping that is generated by using certain control parameters, hence it cannot handle varying control parameters as the prediction and control horizons, weights and constraints. Additionally if the neural network cannot learn the mapping perfectly, a steady-state error or violation of the constraints may result. If a fuzzy model is used instead, it could be possible to design separate rules that correspond

to certain control configurations, e.g., a set of control parameters. Moreover, in this way desired control actions (safety rules, for example) can be imposed "on top" of the controller mapping.

8.4 Summary of the main results of the research described in this thesis

The main contributions of this thesis to the field of fuzzy MIMO control design can be summarized as follows. Chapter 3 deals with the analysis of input-output coupling and interactions, and the consequent decoupling design. Chapter 4 focuses on effective optimization in fuzzy model predictive control. Chapter 5 addresses the derivation of constraints on the control input in MPC that guarantee closed-loop robust asymptotic stability for open-loop BIBO stable processes with an additive l_1-norm bounded model uncertainty.

Appendix A
Constructing TS fuzzy models from data

Although the development of a TS fuzzy model was not one of the aims of this research, we briefly discuss below a class of fuzzy clustering algorithms based on input-output process data, used to derive fuzzy models throughout the thesis. An extensive survey of these data-driven methods can be found in (Babuška, 1998).

Because a single MISO submodel is derived at a time, the submodel index is skipped unless it is necessary to preserve the consistency. Further, it is assumed that the structure of the model, i.e., the input and output variables, has been determined. In the case of dynamic systems, it also means that the representation of the dynamics within the fuzzy model has been chosen. To construct a TS fuzzy model it is then necessary to determine the number of fuzzy rules, and to derive the antecedent fuzzy sets and the consequent parameters for each rule (recall (2.4) on page 16). Prior knowledge, input-output data, or a combination of the two can be used to obtain these parameters.

A.1 Knowledge-based approach

The two main sources of process knowledge are the precise mechanistic description and the unstructured, (semi-) qualitative expert knowledge. Complex processes can also be described through a combination of simpler and well-understood linear subsystems only valid in certain operating regions. Such knowledge can be expressed in the form of TS fuzzy rules. The variables that characterize the change of the operating regions become antecedent variables, and membership functions are defined to specify the relevancy of each model for the given region (Murray-Smith and Johansen, 1997). The local subsystems are stored in the consequents of the rules, and the global output is obtained as the weighted mean of the individual contributions. The consequent parameters can be derived through linearization of a known nonlinear white-box model, or estimated from input-output data.

The process complexity is typically not uniform, which means, for instance, that a single model can well approximate the process dynamics in certain regions, while other regions require rather fine partitioning. In order to accurately describe the pro-

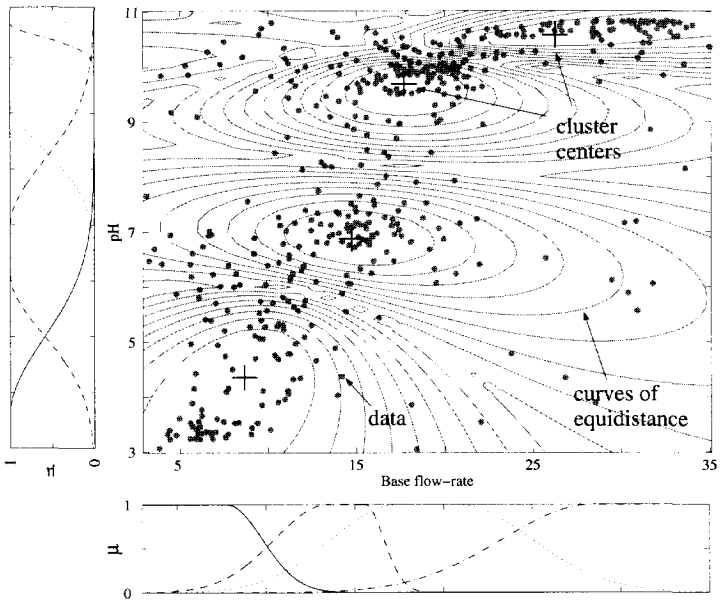


Figure A.1. Hyperellipsoidal fuzzy clusters in the product space of the regressors for a pH neutralization process, see Section 4.5 on page 70. The antecedent membership functions are obtained by projecting the clusters onto the antecedent variables. The points represent the data, '+' are the cluster centers. Also shown are equidistance level curves of the clusters.

cess with as few rules as possible, the membership functions must be placed such that they capture the non-uniform behavior of the process.

Identification methods based on fuzzy clustering use the concept of graded membership to represent the degree to which a given object, represented as a vector of features, is similar to some prototype. The degree of similarity is calculated through a suitably chosen distance measure. Based on the similarity, objects can be arranged such that the objects within a cluster are as similar as possible, and objects from different clusters are as dissimilar as possible. Cluster prototypes can be defined as linear subspaces (lines, planes, hyperplanes) and the distance measure quantifies the distance of data points from that linear subspace (Bezdek, 1981). Each obtained cluster corresponds to one rule in the TS fuzzy model. The antecedent membership functions are obtained by projecting the clusters onto the antecedent variables. This approach is illustrated in Fig. A.1 and described below. Then the consequent parameters are estimated by least-squares methods.

A.2 Data clustering

Product-space fuzzy clustering is based on the data in the product space of the regressor and the regressand $X \times Y$, recall (2.2). Let N denote the number of data samples

selected from the input and output data sequences, which has to be much larger of the number of the regressors, $N \gg \vartheta$. Let $n_{\max} = \max_{i,j}(n_{y,i}, n_{u,j})$ be the largest delay in the regressors. Then the number of points actually used in the identification is $N_a = N - n_{\max} - 1$.

Let $\mathbf{\Phi} \in \mathbb{R}^{N_a \times \vartheta}$ denote the regressor matrix having the regression vector $\boldsymbol{\chi}(k)^T$ in its rows, and $\Upsilon \in \mathbb{R}^{N_a}$ contain the regressand $y(k+1)$, $k = n_{\max}, \ldots, N-1$:

$$\mathbf{\Phi} = \begin{bmatrix} \chi^T(n_{\max}) \\ \vdots \\ \chi^T(N-1) \end{bmatrix} \qquad \Upsilon = \begin{bmatrix} y(n_{\max}+1) \\ \vdots \\ y(N) \end{bmatrix}. \tag{A.1}$$

The set to clustered $\mathbf{Z}$ is formed by appending Υ to $\mathbf{\Phi}$

$$\mathbf{Z} = [\mathbf{\Phi}, \Upsilon].$$

The rows of $\mathbf{Z}$ are denoted by $\{\mathbf{z}_k, k = 1, \ldots, N_a\}$. Let $\mathbf{U} = [\mu_{ik}] \in [0, 1]^{K_l \times N_a}$ denote a fuzzy partitioning matrix of $\mathbf{Z}$, let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{K_l}]$ be a vector of cluster prototypes (centers) to be determined, and let $\mathcal{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{K_l}]$ be a set of cluster covariance matrices $\mathbf{F}_i \in \mathbb{R}^{\vartheta+1 \times \vartheta+1}$

$$\mathbf{F}_i = \frac{\sum_{k=1}^{N_a} (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)(\mathbf{z}_k - \mathbf{v}_i)^T}{\sum_{k=1}^{N_a} (\mu_{ik})^m},$$

where $m \geq 1$ is a weighting exponent that determines the fuzziness (or the overlapping) of the resulting clusters.

Given $\mathbf{Z}$ and the desired number of clusters K_l, the Gustafson and Kessel (1979) algorithm (GK) searches for an optimal fuzzy partition $\mathbf{U}$ and vector of cluster prototypes $\mathbf{V}$ by minimizing the cost function

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{E}) = \sum_{i=1}^{K_l} \sum_{j=1}^{N_a} (\mu_{ij})^m d^2_{ik\mathbf{E}_i}, \tag{A.2}$$

where the function $d_{ik\mathbf{E}_i}$ is the distance of a data point to the cluster prototype yielded by a norm-inducing matrix $\mathbf{E}_i$

$$d^2_{ik\mathbf{E}_i} = (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{E}_i (\mathbf{z}_k - \mathbf{v}_i).$$

The measure of dissimilarity in (A.2) is the squared distance between each data point $\mathbf{z}_j$ and the cluster prototype $\mathbf{v}_i$. This distance is weighted by the power of the membership degree of that point $(\mu_{ij})^m$. The shape of the clusters is determined through the matrix $\mathbf{E}_i$: $\mathbf{E}_i = \mathbf{I}$ induces the standard Euclidean norm, while

$$\mathbf{E}_i = \left(\frac{1}{N_a} \sum_{j=1}^{a} (\mathbf{z}_j - \bar{\mathbf{z}})(\mathbf{z}_j - \bar{\mathbf{z}})^T \right)^{-1}$$

induces the Mahalanobis norm on $\mathbb{R}^{\vartheta}$ (Bezdek, 1981). Here $\bar{\mathbf{z}}$ denotes the sample mean of the data.

The GK algorithm can be found in (Babuška, 1998), among others.

A.3 Estimating antecedent membership functions

The antecedent membership functions can be derived by projecting the fuzzy partition matrix $\mathbf{U}$ onto the antecedent variables, or by computing the membership degrees directly in the product space of the antecedent variables.

When antecedent membership functions are generated by projection, the multidimensional fuzzy sets stored in the rows of the partitioning matrix $\mathbf{U}$ are projected onto the individual antecedent variables. These variables can be the original regression variables or antecedent variables transformed by means of eigenvalue projection, using the ρ largest eigenvectors of the cluster covariance matrices $\mathbf{F}_i$. The eigenvector projection should be applied when the clusters are opaque to the regression space, and cannot be represented by an axis-orthogonal projection with sufficient accuracy (Babuška and Verbruggen, 1995).

To derive a point-wise representation of the antecedent fuzzy set $\mathcal{A}_{i,h}$, we project the ith row $\boldsymbol{\mu}_i$ of the partitioning matrix $\mathbf{U}$ onto the antecedent variable χ_h, $h = 1, \ldots, \vartheta$. This point-wise membership function is approximated by some suitable analytic function, e.g., a piece-wise exponential membership function. In this way the membership degrees can be computed for control or prediction purposes, and also for input data not contained in the data set $\mathbf{Z}$.

When multidimensional antecedent membership functions are utilized, the membership degree is computed directly for the entire input vector (without decomposition), i.e., $\beta_i(\boldsymbol{\chi}) = \mu_{\mathcal{A}_i}(\boldsymbol{\chi})$. Here the membership functions $\mathcal{A}_i$ are derived through the distance of $\boldsymbol{\chi}$ from the projection of the cluster prototype (center) $\mathbf{v}_i$ onto X. The membership degree is computed in inverse proportion to this distance. Denote with

$$\mathbf{F}_i^{\boldsymbol{\chi}} = [f_{jh}], \quad 1 \leq j, h \leq \vartheta$$

the partition of the cluster covariance matrix $\mathbf{F}_i$ which includes all but the last column. This matrix describes the norm of the cluster in the antecedent space X. The corresponding norm-inducing matrix is

$$\mathbf{E}_i^{\boldsymbol{\chi}} = \left[\kappa_i \det\left(\mathbf{F}_i^{\boldsymbol{\chi}}\right)\right]^{1/\vartheta} \left(\mathbf{F}_i^{\boldsymbol{\chi}}\right)^{-1},$$

where $\kappa_i = |\mathbf{E}_i|$. Let $\mathbf{v}_i^{\boldsymbol{\chi}} = [v_{1i}, \ldots, v_{\vartheta i}]$ denote the projection of the cluster center $\mathbf{v}_i$ onto the regressor space. The inner-product norm

$$d_{\mathbf{E}_i^{\boldsymbol{\chi}}}(\boldsymbol{\chi}, \mathbf{v}_i^{\boldsymbol{\chi}}) = (\boldsymbol{\chi} - \mathbf{v}_i^{\boldsymbol{\chi}})^T \mathbf{E}_i^{\boldsymbol{\chi}} (\boldsymbol{\chi} - \mathbf{v}_i^{\boldsymbol{\chi}})$$

is converted into membership degree by

$$\mu_{\mathcal{A}_i}(\boldsymbol{\chi}) = \frac{1}{\sum_{j=1}^{K_l} \left(d_{\mathbf{E}_i^{\boldsymbol{\chi}}}(\boldsymbol{\chi}, \mathbf{v}_i^{\boldsymbol{\chi}}) / d_{\mathbf{E}_i^{\boldsymbol{\chi}}}(\boldsymbol{\chi}, \mathbf{v}_j^{\boldsymbol{\chi}})\right)^{2/(m-1)}}.$$

The last expression is known as a "probabilistic" method, since the degree of fulfillment of one rule is computed relatively to the other rules, and the sum of the membership degrees of all rules equals one.

A.4 Estimating consequent parameters

Once the antecedent membership functions have been determined, the consequent parameters ζ_i, η_i and θ_i can be obtained by using two different approaches based on the least-squares estimate. One is to solve K_l independent, local or weighted, least-squares problems, one for each rule. This approach is independent of the aggregation/defuzzification method used; the parameters for each model are estimated separately. The other is to solve a global least-squares problem resulting from the weighted mean defuzzification formula (2.6). For the purpose of local interpretation and analysis of the TS model, the weighted least-squares approach is preferable as it gives more reliable local models. The global least-squares method gives a minimal prediction error, and thus it is more suitable for deriving prediction models. At the same time, however, it biases the estimate of the local model parameters.

The identification data are given in the matrices Φ and Υ, see (A.1), and the membership degrees of the fuzzy partition are arranged in the following matrix:

$$\mathbf{W}_i = \begin{bmatrix} \beta_{i1} & 0 & \cdots & 0 \\ 0 & \beta_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_{iN_a} \end{bmatrix}. \tag{A.3}$$

The consequent parameters of the rule belonging to the ith cluster, ζ_i, η_i and θ_i are concatenated into a single parameter vector Θ_i:

$$\Theta_i = \left[\zeta_i^T, \eta_i^T, \theta_i\right]^T. \tag{A.4}$$

Appending a unitary column to Φ gives the extended regressor matrix Φ_e:

$$\Phi_e = [\Phi, \mathbf{I}], \tag{A.5}$$

and correspondingly $\mathcal{X}_e = [\mathcal{X}, 1]$.

A.4.1 Local Least-squares Method

Assuming that each cluster represents a local linear model of the system, we can obtain the consequent parameters for each rule Θ_i, $i = 1, 2, \ldots, K_l$, as a weighted least-squares estimate. The association of the data pair $(\mathcal{X}_k, y_k)$ with the ith local model is given through the corresponding membership degree β_{ik}.

If the columns of Φ_e are linearly independent and $\beta_{ik} > 0$ for $1 \leq k \leq N_a$, then

$$\Theta_i = \left[\Phi_e^T \mathbf{W}_i \Phi_e\right]^{-1} \Phi_e^T \mathbf{W}_i \mathbf{y} \tag{A.6}$$

is the least-squares solution of $\mathbf{y} = \Phi_e \Theta + \epsilon$, where the kth data pair $(\mathcal{X}_k, y_k)$ is weighted by β_{ik}. The parameters ζ_i, η_i and θ_i are given by:

$$\zeta_i = \left[\Theta_1', \Theta_2', \ldots, \Theta_\rho'\right], \quad \eta_i = \left[\Theta_{\rho+1}', \Theta_{\rho+2}', \ldots, \Theta_{\rho+m}'\right], \quad \theta_i = \left[\Theta_{\rho+m+1}'\right]. \tag{A.7}$$

Note that for a computer implementation, it is more efficient to first multiply each row of Φ_e and $\mathbf{y}$ by $\sqrt{\beta_{ik}}$, rather than to use (A.6):

$$\tilde{\Phi}_i = \begin{bmatrix} \sqrt{\beta_{i1}}\boldsymbol{\chi}_{e1}^T \\ \sqrt{\beta_{i2}}\boldsymbol{\chi}_{e2}^T \\ \vdots \\ \sqrt{\beta_{iN_a}}\boldsymbol{\chi}_{eN_a}^T \end{bmatrix}, \quad \tilde{\mathbf{y}}_i = \begin{bmatrix} \sqrt{\beta_{i1}}y_1 \\ \sqrt{\beta_{i2}}y_2 \\ \vdots \\ \sqrt{\beta_{iN_a}}y_{N_a} \end{bmatrix}, \tag{A.8}$$

and to compute Θ_i by:

$$\Theta_i = \left[\tilde{\Phi}_i^T \tilde{\Phi}_i\right]^{-1} \tilde{\Phi}_i^T \tilde{\mathbf{y}}_i . \tag{A.9}$$

Having determined a best-fit set of parameter values, it is useful to have some indication of the significance of the estimated parameters. Such an indication can be, for example, the standard deviation of each parameter, see Chapter 5. One approach for estimating the standard deviations $\{\sigma(\Theta_{i,j})\}_{j=1}^{\vartheta}$ of the parameter values Θ_i is given in (Bard, 1974). An estimate of the standard deviation of the parameter $\Theta_{i,j}$ is given by $\sigma(\Theta_{i,j}) = \sqrt{\varsigma_{jj}}$, where

$$[\varsigma_{j,\iota}]_{j,\iota=1}^{\vartheta} = \frac{2}{N_a - \vartheta}\sqrt{\left(\mathbf{U}_j .* \tilde{\mathbf{y}}_j - \tilde{\Phi}_i\Theta_i\right)^T \left(\mathbf{U}_j .* \tilde{\mathbf{y}}_j - \tilde{\Phi}_i\Theta_i\right)} \left(\tilde{\Phi}_i^T \tilde{\Phi}_i\right)^{-1} . \tag{A.10}$$

The symbol $.*$ here denotes element-by-element multiplication (Hadamard or Schur product).

A.4.2 Global Least-squares Method

The weighted least-squares method gives an optimal estimate of the parameters of the local models, but it does not provide an optimal TS model in terms of a minimal prediction error. In order to obtain an optimal global predictor, we consider the contribution of all rules simultaneously. The fuzzy-mean defuzzification formula (2.6) is linear in the consequent parameters, so parameter estimates can be obtained by solving a least-squares problem.

The degree of fulfillment of the ith rule β_{ik} can be obtained by projecting the ith row $\boldsymbol{\mu}_i$ of the fuzzy partitioning matrix $\mathbf{U}$ onto the antecedent space (recall that each row of $\mathbf{U}$ contains a point-wise definition of the membership function for the data in the product space $X \times Y$):

$$\beta_{ik} = \text{proj}(\mu_{ik}), \quad k = 1, \ldots, N_a, \tag{A.11}$$

where "proj" denotes the projection operator (Yager and Filev, 1994). Through the projection, a set of repeated identical regression vectors $\boldsymbol{\chi}_k$ in the data is assigned the maximum membership degree from this set (which may depend on y_k as well). Another possibility is to compute the degrees of fulfillment after generating the antecedent membership functions.

In order to write (2.6) in a matrix form for all the data $(\boldsymbol{\chi}_k, y_k)$, $1 \leq k \leq N_a$, denote Γ_i a diagonal matrix in $\mathbb{R}^{N_a \times N_a}$ having the normalized membership degree

$w_{ik} = \beta_{ik} / \sum_{j=1}^{K_l} \beta_{jk}$ as its kth diagonal element. Denote Φ' the matrix in $\mathbb{R}^{N_a \times K_l(\vartheta+1)}$ composed from matrices Γ_i and Φ_e as follows

$$\Phi' = [(\Gamma_1 \Phi_e), (\Gamma_2 \Phi_e), \ldots, (\Gamma_{K_l} \Phi_e)]$$

Denote the parameter vector $\Theta' \in \mathbb{R}^{K_l(\vartheta+1)}$ by

$$\Theta' = \left[\Theta_1^T, \Theta_2^T, \ldots, \Theta_{K_l}^T\right] \tag{A.12}$$

The resulting least-squares problem $\Upsilon = \Phi' \Theta' + \epsilon$ has the solution

$$\Theta' = \left[(\Phi')^T \Phi'\right]^{-1} (\Phi')^T \Upsilon.$$

From (A.12) the optimal parameters ζ_i, η_i and θ_i are

$$\begin{aligned} \zeta_i &= \left[\Theta'_{s+1}, \Theta'_{s+2}, \ldots, \Theta'_{s+\rho}\right], \\ \eta_i &= \left[\Theta'_{s+\rho+1}, \Theta'_{s+\rho+2}, \ldots, \Theta'_{s+\rho+m}\right], \\ \theta_i &= \left[\Theta'_{s+\rho+m+1}\right], \qquad \text{where } s = (i-1)(\vartheta+1). \end{aligned} \tag{A.13}$$

Now the estimate of the standard deviation $\{\sigma(\Theta_{i,j})\}_{j=1}^{\vartheta}$ of the parameters Θ_i is given by $\sigma(\Theta_{i,j}) = \sqrt{\varsigma_{jj}}$, where

$$[\varsigma_{j,\iota}]_{j,\iota=1}^{\vartheta} = \frac{2}{N_a - \vartheta} \sqrt{(\Upsilon - \Gamma_i \Phi_e \Theta_i)^T (\Upsilon - \Gamma_i \Phi_e \Theta_i)} \left(\Phi_e^T \Phi_e\right)^{-1}. \tag{A.14}$$

A.4.3 Interpretation of the consequent parameters

The output of the TS model is an interpolation of the local linear submodels, hence the steady-state dynamic gain of the model can be locally interpreted as the gain of the interpolated local submodels (Abonyi and Babuška, 2000). This locally interpreted gain is not equal to the gain obtained by linearization of the fuzzy model at the considered equilibrium. Fuzzy models with consequent parameters obtained through the weighted least-squares method typically yield a poor steady-state representation and the model can only be interpreted locally. In contrast, when the global least-squares method is used to derive the consequent parameters, a qualitatively bad local interpretation of the gain can result, even though the model approximates the process well.

The difference between the globally and locally interpreted gain can be reduced by incorporating prior knowledge in global identification (Abonyi et al., 1999), while the steady-state representation of locally identified fuzzy models can be improved through inference based on the smoothed maximum operator rather than on the weighted mean (Babuška et al., 1996).

Appendix B
The concept of right invertibility

A right inverse is a second discrete-time nonlinear system that is such that when the original system is applied in series with this right inverse, then its outputs are equal to the inputs of the right inverse system (Fig. B.1). More general is the concept of forward time-shift right inverse in which the input to the right inverse system is obtained by using the α-step forward time-shift operator q^{α} on the reference signal: $q^{\alpha} y_{\text{ref}}(k) \stackrel{\text{def}}{=} y_{\text{ref}}(k+\alpha)$ (Kotta, 1995).

Consider the discrete-time nonlinear system

$$S: \left\{ \begin{array}{rcl} \mathbf{x}(k+1) & = & \mathbf{F}\big(\mathbf{x}(k), \mathbf{u}(k)\big) \\ \mathbf{y}(k) & = & \mathbf{H}\big(\mathbf{x}(k)\big), \end{array} \right. \tag{B.1}$$

$\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n$, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$, $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^p$, for which we state the following

Definition B.1 *The system (B.1) is called locally right invertible in a neighborhood of its equilibrium point $(\mathbf{x}^0, \mathbf{u}^0)$, if there exist sets $\mathcal{U}^0$ and $\mathcal{Y}^0$ such that given initial $\mathbf{x}_0 \in \mathbf{X}_0$, we are able to find for any sequence $\{\mathbf{y}_{\text{ref}}(k); 0 \leq k \leq k_F\} \in \mathcal{Y}^0$ a control sequence $\{\mathbf{u}_{\text{ref}}(k);\ 0 \leq k \leq k_F\} \in \mathcal{U}^0$ (not necessarily unique) such that*

$$\mathbf{y}\,(k, \mathbf{x}_0, \mathbf{u}_{\text{ref}}(0), \ldots, \mathbf{u}_{\text{ref}}(k)) = \mathbf{y}_{\text{ref}}(k),\ \ 0 \leq k \leq k_F,$$

for k_F some finite time instant.

Definition B.2 *The system (B.1) is called locally forward time-shift right invertible in a neighborhood of its equilibrium point $(\mathbf{x}^0, \mathbf{u}^0)$, if there exist integers $0 \leq \alpha_1 \leq \alpha_2 \leq$ $, \cdots, \leq \alpha_p$, a reordering of output components $\mathbf{y}_i$, $i = 1, \ldots, p$, sets $\mathcal{U}^0$ and $\mathcal{Y}^0$ such that given $\mathbf{x}_0 \in \mathbf{X}_0$ we are able to find for any sequence $\{\mathbf{y}_{\text{ref}}(k); 0 \leq k \leq k_F\} \in \mathcal{Y}^0$ a control sequence $\{\mathbf{u}_{\text{ref}}(k); 0 \leq k \leq k_F\} \in \mathcal{U}^0$ yielding*

$$\begin{aligned} &\mathbf{y}_i\,(k, \mathbf{x}_0, \mathbf{u}_{\text{ref}}(0), \ldots, \mathbf{u}_{\text{ref}}(k)) = \mathbf{y}_{\text{ref},i}(k), \\ &\alpha_i \leq k \leq k_F,\ \ i = 1, \ldots, p. \end{aligned}$$

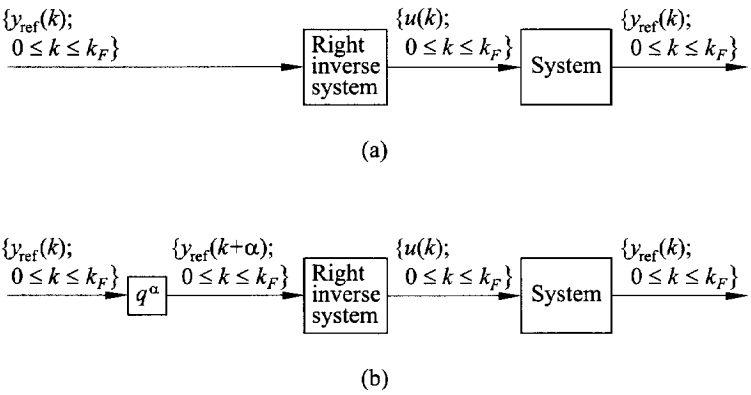


Figure B.1. Right invertibility: (a) Right inverse system; (b) FTS Right inverse system

Definition of $(d_1, \ldots, d_p)$-forward time-shift right invertibility. With each component of the output y_i we can associate a delay order d_i (also referred to as characteristic number or relative degree) with respect to u in the following manner: d_i – if it exists – determines the inherent time delays between the inputs and the ith output; the input $\mathbf{u}(k)$ affects the ith output only after d_i steps, that is, at the time instant $k + d_i$. The interpretation of $d_i = \infty$ is that the ith output evolves over time independently of the input sequence applied to the (B.1).

Definition B.3 *The system (B.1) is locally $(d_1, \ldots, d_p)$-forward time-shift $((d_1, \ldots, d_p)$ - $FTS)$ right invertible in a neighborhood of its equilibrium point $(\mathbf{x}^0, \mathbf{u}^0)$ if there exist sets $\mathcal{U}^0$ and $\mathcal{Y}^0$ such that given $\mathbf{x}_0 \in \mathbf{X}_0$ we are able to find for any sequence $\{\mathbf{y}_{\text{ref}}(k); 0 \le k \le k_F\} \in \mathcal{Y}^0$ a control sequence $\{\mathbf{u}_{\text{ref}}(k); 0 \le k \le k_F\} \in \mathcal{U}^0$ yielding*

$$\mathbf{y}_i\,(k, \mathbf{x}_0, \mathbf{u}_{\text{ref}}(0), \ldots, \mathbf{u}_{\text{ref}}(k)) = \mathbf{y}_{\text{ref},i}(k),$$
$$d_i \le k \le k_F, \;\; i = 1, \ldots, p.$$

This definition says that for ith output it is possible to reproduce all sequences from $\mathcal{Y}_i^0$ beginning at time instant d_i. But $(d_1, \ldots, d_p)$ -FTS right invertibility does not allow us to reproduce the first d_i terms in the arbitrary sequence $\{\mathbf{y}_{\text{ref},i}(k); 0 \le k \le k_F\}$ from $\mathcal{Y}_i^0$.

Necessary and sufficient conditions for $(d_1, \ldots, d_p)$ -forward time-shift right invertibility. To be useful in the state-space approach, the conditions for invertibility should be phrased directly in terms of the function $\mathbf{F}$ and $\mathbf{H}$.

The idea is to apply the one-step FTS operator q to those output components which do not explicitly depend on the input. Assuming that each delay order d_i is finite, we modify the output equation of the system by repeatedly operating on each of the scalar output equations the FTS operators, so as to obtain a system of equations each of which depends explicitly on the control $\mathbf{u}(k)$. From the definition of delay orders

d_i, $i = 1,\ldots,p$, we obtain

$$\begin{aligned} q^{d_1} y_1(k) &= y_1(k+d_1) = h_1^{d_1}\left(\mathbf{F}(\mathbf{x}(k),\mathbf{u}(k))\right) \\ &\cdots \\ q^{d_p} y_p(k) &= y_p(k+d_p) = h_p^{d_p}\left(\mathbf{F}(\mathbf{x}(k),\mathbf{u}(k))\right), \end{aligned} \tag{B.2}$$

or in the vector form

$$\begin{bmatrix} y_1(k+d_1) \\ \vdots \\ y_p(k+d_p) \end{bmatrix} = \mathbf{A}\left(\mathbf{x}(k),\mathbf{u}(k)\right), \tag{B.3}$$

where $h_i^0(\mathbf{F}(\mathbf{x},\mathbf{u}))$ and $h_i^j(\mathbf{F}(\mathbf{x},\mathbf{u}))$ are defined as $h_i(\mathbf{x},\mathbf{u}) \stackrel{\text{def}}{=} h_i^0(\mathbf{F}(\mathbf{x},\mathbf{u}))$ and $h_i^{j+1}(\mathbf{x},\mathbf{u}) \stackrel{\text{def}}{=} h_i^j\left(\mathbf{F}(\mathbf{x},\mathbf{u})\right)$, for $j = 0,\ldots,\alpha_i$.

Let the decoupling matrix $\mathbf{K}(\mathbf{x},\mathbf{u})$ be

$$\mathbf{K}(\mathbf{x},\mathbf{u}) = \frac{\partial}{\partial \mathbf{u}}\mathbf{A}(\mathbf{x},\mathbf{u}) = \frac{\partial}{\partial \mathbf{u}}\begin{bmatrix} h_1^{d_1}\left(\mathbf{F}(\mathbf{x},\mathbf{u})\right) \\ \vdots \\ h_p^{d_p}\left(\mathbf{F}(\mathbf{x},\mathbf{u})\right) \end{bmatrix}. \tag{B.4}$$

The rank of $\mathbf{K}(\mathbf{x},\mathbf{u})$ is, in general, state and control dependent. To ensure smoothness of the solution of (B.3) with respect to $\mathbf{u}(k)$ we have to assume that $\mathbf{K}(\mathbf{x},\mathbf{u})$ has a constant rank.

Definition B.4 *We call the equilibrium point* $(\mathbf{x}^0,\mathbf{u}^0)$ *of the system (B.1) regular with respect to* $(d_1,\ldots,d_p)$ *-FTS right invertibility, if the rank of the decoupling matrix* $\mathbf{K}(\mathbf{x},\mathbf{u})$ *of the system (B.1) is constant around* $(\mathbf{x}^0,\mathbf{u}^0)$.

Theorem B.1 *Assume that for the system (B.1)* $d_i \leq n$, $i = 1,\ldots,p$. *Then the system (B.1) is locally* $(d_1,\ldots,d_p)$ *-FTS right invertible around a regular equilibrium point* $(\mathbf{x}^0,\mathbf{u}^0)$, *if and only if the* $rank\left(\mathbf{K}(\mathbf{x}^0,\mathbf{u}^0)\right) = p$.

Remark. Clearly, rank $K(\mathbf{x}^0,\mathbf{u}^0) = p$ requires $m \geq p$. So $p \leq m$ is always a necessary condition for system to have a $(d_1,\ldots,d_p)$ -FTS right inverse, that is, the system must have at least as many inputs as outputs.

The right inverse of a discrete-time nonlinear system can be obtained by solving a system of nonlinear (transcendental) algebraic equations with respect to the control vector. Let $\Sigma_{\mathbf{x}_0}^S$ be the input/output map of S with initial condition $\mathbf{x}_0$. Then

$$\Sigma_{\mathbf{x}_0}^S\left(\mathbf{u}(k)\right) = \mathbf{y}_{\text{ref}}(k+1). \tag{B.5}$$

The solution of (B.5) with respect to $\mathbf{u}(k)$ (provided $\Sigma_{\mathbf{x}_0}^S\mathbf{u}(k)$ is invertible) is

$$\mathbf{u}_{\text{ref}}(k) = \left(\Sigma_{\mathbf{x}_0}^S\right)^{-1}\left(\mathbf{y}_{\text{ref}}(k+1)\right). \tag{B.6}$$

Finding an analytical solution for (B.6) is not a trivial problem. It could be solved numerically at each sample by transforming it into constrained nonlinear optimization

problem. The latter, however, is a time-consuming task with no guarantees for reaching the global optimum. For linear-in-control (or affine) systems, the conditions for the existence of an analytical solution of (B.6) are much looser. Consider a discrete-time affine nonlinear input-output system

$$\mathbf{y}(k+1) = \mathcal{F}\big(\mathbf{x}(k)\big) + \mathcal{G}\big(\mathbf{x}(k)\big)\mathbf{u}(k),$$

where the matrix function $\mathcal{G}(\mathbf{x})$ is invertible in a neighborhood of an equilibrium $\mathbf{x}^0$. Then

$$\mathbf{u}_{\text{ref}}(k) = \mathcal{G}^{-1}\big(\mathbf{x}(k)\big)\big(\mathbf{y}_{\text{ref}}(k) - \mathcal{F}(\mathbf{x}(k))\big).$$

Appendix C
Sequential quadratic programme

The constrained nonlinear optimization problem to be solved at time instant k is stated as

$$\min_{\mathbf{u}(k)\in\mathbb{R}^n} J(\mathbf{u}(k)) \tag{C.1}$$

subject to:

$$G_j\big(\mathbf{u}(k)\big) = 0 \quad j = 1,\ldots,c_e \tag{C.2a}$$

$$G_j\big(\mathbf{u}(k)\big) \leq 0 \quad j = c_e+1,\ldots,c \tag{C.2b}$$

where $\mathbf{u}(k) \in \mathbb{R}^n$, $n = mH_c$ is the vector of manipulated variables (control actions), $J\big(\mathbf{u}(k)\big) : \mathbb{R}^n \to \mathbb{R}$ is the nonlinear cost function, and the nonlinear vector function $\mathbf{G}\big(\mathbf{u}(k)\big) : \mathbb{R}^n \to \mathbb{R}^c$ returns the values of equality and inequality constraints evaluated at $\mathbf{u}(k)$.

Given the optimization problem (C.1)-(C.2), the idea is to iteratively formulate a QP sub-problem based on linearized constraints and a quadratic approximation of the Lagrangian function

$$L\big(\mathbf{u}(k),\boldsymbol{\lambda}\big) = J\big(\mathbf{u}(k)\big) + \sum_{j=1}^{c} \lambda_j G_j\big(\mathbf{u}(k)\big)\,.$$

At each iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. These are used to generate a QP sub-problem whose solution forms a search direction for a line search procedure. Below the three main stages are discussed briefly.

To avoid ambiguity, throughout this appendix the index j is used to denote an element of a vector and the indices i and s refer to an SQP and a QP sub-problem iteration, respectively.

C.1 Updating the Hessian matrix

At each SQP iteration i, a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function is calculated using the formula of Broyden (1970), Fletcher

(1970), Goldfarb (1970) and Shanno (1970) (BFGS)

$$\mathbf{Hs}_{i+1} = \mathbf{Hs}_i + \frac{\mathbf{q}_i\mathbf{q}_i^T}{\mathbf{q}_i^T\mathbf{s}_i} - \frac{\mathbf{Hs}_i^T\mathbf{Hs}_i}{\mathbf{s}_i^T\mathbf{Hs}_i\mathbf{s}_i}$$

where

$$\mathbf{s} = \mathbf{u}_{i+1}(k) - \mathbf{u}_i(k)$$

$$\mathbf{q}_i = \nabla J\big(\mathbf{u}_{i+1}(k)\big) + \sum_{j=1}^{n} \lambda_j \nabla G_i\big(\mathbf{u}_{i+1}(k)\big) - \left(\nabla J\big(\mathbf{u}_i(k)\big) + \sum_{j=1}^{n} \lambda_j \nabla G_i\big(\mathbf{u}_i(k)\big)\right).$$

A positive definite Hessian is maintained providing that $\mathbf{q}_i^T\mathbf{s}_i$ is positive at each update and that **Hs** is initialized with a positive definite matrix. When $\mathbf{q}_i^T\mathbf{s}_i$ is not positive, $\mathbf{q}_i$ is modified on an element by element basis to ensure $\mathbf{q}_i^T\mathbf{s}_i > 0$.

C.2 Quadratic programme solution

At each SQP iteration a QP problem

$$\begin{aligned} \min_{\mathbf{u}(k)\in\mathbb{R}^n} q(\mathbf{u}(k)) &= \frac{1}{2}\mathbf{u}(k)^T\mathbf{Hs}\mathbf{u}(k) + \mathbf{f}^T\mathbf{u}(k) \\ A_j\mathbf{u}(k) &= b_j \quad j = 1,\ldots,c_e \\ A_j\mathbf{u}(k) &\le b_j \quad j = c_e+1,\ldots,c \end{aligned} \tag{C.3}$$

is solved, where A_j refers to the jth row of the $c \times n$ matrix **A** of linearized constraints. The solution of the QP involves a two-stages procedure. In the first stage, a feasible point is calculated (if one exists). During the second stage, a sequence of feasible points is generated that converge to the solution. In this way an active set $\bar{\mathbf{A}}_s$ is maintained, which is an estimate of the active constraints at the solution point.

The active set $\bar{\mathbf{A}}_s$ is updated at each iteration s and is used to form a basis for a search direction $\hat{\mathbf{d}}_s$ (here $\hat{\mathbf{d}}_s$ is used to distinguish from $\mathbf{u}_i(k)$ in the high-level iterations). Equality constraints always remain in the active set $\bar{\mathbf{A}}_s$. The search direction $\hat{\mathbf{d}}_s$ is calculated such that to minimize the objective function while remaining on the boundaries of any active constraints. The feasible subspace for $\hat{\mathbf{d}}_s$ is formed from a basis $\mathbf{Z}_s$ whose columns are orthogonal to the active set $\bar{\mathbf{A}}_s$ ($\bar{\mathbf{A}}_s\mathbf{Z}_s = 0$). Thus a search direction, based on columns of $\mathbf{Z}_s$ is guaranteed to remain on the boundaries of the active constraints.

The matrix $\mathbf{Z}_s$ contains the last $c - l$ columns ($l < c$) of the QR decomposition the matrix $\bar{\mathbf{A}}_s^T$, where l is the number of active constraints

$$\mathbf{Z}_s = \mathbf{Q}[:, l+1 : c], \quad \text{where } \mathbf{Q}^T\bar{\mathbf{A}}_s^T = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}.$$

Having found $\mathbf{Z}_s$, a search direction $\hat{\mathbf{d}}_s$ is sought that minimizes $q(\mathbf{d})$, where $\hat{\mathbf{d}}_s$ is a linear combination of the columns of $\mathbf{Z}_s$: $\hat{\mathbf{d}}_s = \mathbf{Z}_s\mathbf{p}$ for some vector $\mathbf{p}$.

Then substituting $\mathbf{p}$ for $\hat{\mathbf{d}}_s$ in the cost function (C.3)

$$q(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{Z}_s^T\mathbf{H}\mathbf{s}\mathbf{Z}_s\mathbf{p} + \mathbf{f}^T\mathbf{Z}_s\mathbf{p}$$

which, when differentiated with respect to $\mathbf{p}$, yields

$$\nabla q(\mathbf{p}) = \mathbf{Z}_s^T\mathbf{H}\mathbf{s}\mathbf{Z}_s\mathbf{p} + \mathbf{Z}_s^T\mathbf{f}.$$

The terms $\nabla q(\mathbf{p})$ and $\mathbf{Z}_s^T\mathbf{H}\mathbf{s}\mathbf{Z}_s$ are respectively the projected gradient and Hessian of the cost function onto the subspace defined by $\mathbf{Z}_s$. Assuming the Hessian is positive definite, then the minimum of the cost function in the subspace defined by $\mathbf{Z}_s$ is at the point where $\nabla q(\mathbf{p}) = 0$, which is the solution of the system of linear equations

$$\mathbf{Z}_s^T\mathbf{H}\mathbf{s}\mathbf{Z}_s\mathbf{p} = -\mathbf{Z}_s^T\mathbf{f}.$$

Then $\mathbf{d}$ is updated

$$\mathbf{d}_{s+1} = \mathbf{d}_s + \hat{\alpha}\hat{\mathbf{d}}_s, \quad \text{where } \hat{\mathbf{d}}_s = \mathbf{Z}_i^T\mathbf{p}.$$

Because of the quadratic nature of the cost function, there are only two possible values for the step length $\hat{\alpha}$ at each iteration. A unit step along $\hat{\mathbf{d}}_s$ is the exact step to the minimum of the cost function restricted to the null space of $\bar{\mathbf{A}}_s$. If such a step can be taken without violating the constraints, then this is the solution to the QP (C.3). Otherwise, the step along $\hat{\mathbf{d}}_s$ to the nearest constraint is less that unity and a new constraint is included in the active set in the next iteration. The distance to the constraint boundaries in any direction $\hat{\mathbf{d}}_s$ is

$$\hat{\alpha} = \min_j \left\{ \frac{-(A_j\mathbf{d}_s - b_j)}{A_j\hat{\mathbf{d}}_s} \right\}, \quad j = 1, \ldots, c$$

and is defined for constraints not in the active set, and where the direction $\hat{\mathbf{d}}_s$ is toward the constraint boundary, i.e., $A_j\hat{\mathbf{d}}_s > 0$.

When n independent constraints are included in the active set without locating the minimum, Lagrange multipliers $\boldsymbol{\lambda}_s$ are calculated that satisfy the nonsingular set of linear equations

$$\bar{\mathbf{A}}_s^T\boldsymbol{\lambda}_s = \mathbf{f}.$$

If all elements of $\boldsymbol{\lambda}_s$ are positive, then $\mathbf{d}_s$ is the solution to the QP (C.3). If an element of $\boldsymbol{\lambda}_s$ is negative and does not correspond to an equality constraint, it is deleted from the active set and new iteration is initiated.

Initialization. A feasible point is needed to start the algorithm. If initial point is not feasible, then the linear programme (LP)

$$\min_{\gamma\in\mathbb{R},\mathbf{d}\in\mathbb{R}^n} \gamma \tag{C.4}$$

$$A_j\mathbf{d} = b_j \quad j = 1, \ldots, c_e$$

$$A_j\mathbf{d} - \gamma \leq b_j \quad j = c_e + 1, \ldots, c$$

can be applied to find a feasible point. Such a feasible point to (C.4) can be found by setting $\mathbf{d}$ to a value that satisfies the equality constraints.

The QP algorithm is modified for LP by setting the search direction to the steepest descent direction at each iteration

$$\hat{\mathbf{d}}_s = -\mathbf{Z}_s \mathbf{Z}_s^T \mathbf{g}_s$$

where $\mathbf{g}_s$ is the gradient of the linear cost function (equal to the coefficients).

If a feasible point is found using the LP method, the second QP stage is entered. The search direction $\hat{\mathbf{d}}_s$ is initialized with a search direction $\hat{\mathbf{d}}_1$ obtained by solving

$$\mathbf{H}\mathbf{s}\hat{\mathbf{d}}_1 = \mathbf{g}_s,$$

where $\mathbf{g}_s$ is the gradient of the cost function at the current iteration, i.e., $\mathbf{g}_s = \mathbf{H}\mathbf{s}\mathbf{d}_s + \mathbf{f}$.

If a feasible solution is not found for the QP problem, the search direction of search for the main SQP routine $\mathbf{d}_i$ is taken as the one that minimizes γ.

C.3 Convergence through line search

In an iterative optimization scheme, the QP solution in ith SQP iteration at time instant k, $\mathbf{d}_i(k)$, provides a search direction toward the minimum of the optimization problem. To guarantee convergence, a line search mechanism is used that takes into account reduction both in the cost function and in the constraints

$$\mathbf{u}_i(k) = \begin{cases} \mathbf{u}(k-1) + \alpha_i \mathbf{d}_i(k) & \text{If } i = 1 \\ \mathbf{u}_{i-1}(k) + \alpha_i \mathbf{d}_i(k) & \text{If } i > 1 \end{cases} \tag{C.5}$$

where $0 < \alpha_i \leq 1$ is a step length chosen by a line search routine to give a reduction in the merit function proposed by Powell (1983)

$$\Psi\big(\mathbf{u}_i(k), \mathbf{r}_i\big) = J\big(\mathbf{u}_i(k)\big) + \sum_{j=1}^{c_e} r_{i,j} G_j\big(\mathbf{u}_i(k)\big) + \sum_{j=c_e+1}^{c} r_{i,j} \max\Big[0, G_j\big(\mathbf{u}_i(k)\big)\Big] \tag{C.6}$$

which considers reduction both in the cost function (C.1) and in the constraints (C.2). The positive parameters $\mathbf{r} \in \mathbb{R}^c$ are updated according to

$$r_{i+1,j} = \max_j \left\{ \lambda_j, \frac{1}{2}\big(r_{i,j} + \lambda_j\big) \right\}, \quad j = 1, \ldots, c \tag{C.7}$$

where λ_j are the Lagrange multipliers for the current QP solution. This allows positive contributions from constraints that are inactive in the QP solution, but were recently active. The penalty parameter $\mathbf{r}$ is initially set to

$$r_{1,j} = \frac{\big\|\nabla J\big(\mathbf{u}_1(k)\big)\big\|}{\big\|\nabla G_j\big(\mathbf{u}_1(k)\big)\big\|}, \quad j = 1, \ldots, c \tag{C.8}$$

where $\|\cdot\|$ represents the Euclidean norm. This results in larger contributions to the penalty parameter from constraints with smaller gradients, which would be the case for active constraints at the solution point.

Let α_1 be the first element in a monotonously decreasing sequence $\{\alpha_1, \alpha_2, \alpha_3, \ldots\}$. Then it can be proved (Powell, 1983) that the line search generates a sequence $\{\mathbf{u}_i(k), i = 1,2,3,\ldots\}$ whose limit is the Kuhn-Tucker point of the optimization problem (4.9) to (4.25). The line search routine is summarized in Algorithm C.1.

The introduction of the line search routine, however, as reported by Powell (1983) may decrease the convergence rate. The situation $\alpha = 1$ which gives superlinear convergence in (C.5) is not always allowed by the merit function (C.6).

Algorithm C.1 (Line search algorithm)

Step 1: **Initialize** $\mathbf{r}_i$ according to (C.8) if $i = 1$ or according to (C.7) otherwise.

Step 2: **Calculate** the cost function $J(\mathbf{u}_i(k))$ and constraints $G_j(\mathbf{u}_i(k))$, $j = 1,\ldots,c$. The merit function $\Psi(\mathbf{u}_i(k), \mathbf{r}_i)$ is split into two parts looking for improvements in the cost function and the constraints, respectively

$$\Psi_{J,init} = -1/(J(\mathbf{u}_i(k)) + 1)$$

$$\Psi_{G,init} = J(\mathbf{u}_i(k)) + \sum_{j=1}^{c_e} \mathbf{r}_{i,j} G_j(\mathbf{u}_i(k)) + \sum_{j=c_e+1}^{c} \mathbf{r}_{i,j} \max\left(0, G_j(\mathbf{u}_i(k))\right)$$

Step 3: **Set** $\alpha_i = 2$, $\Psi_J = \Psi_{J,init} + 1$ and $\Psi_G = \Psi_{G,init} + 1$.

Repeat:

Step 4.1: **Reduce** the step length $\alpha_i = \alpha_i/2$.

Step 4.2: **Update** the solution $\mathbf{u}_i(k) = \mathbf{u}_{i-1}(k) + \alpha_i \cdot \mathbf{d}_i(k)$.

Step 4.3: **Update** the cost function $J = J(\mathbf{u}_i(k))$.

Step 4.4: **Update** the constraints and take max

$$G_j = G_j(\mathbf{u}_i(k)),\ j = 1,\ldots,c$$
$$G_{\max} = \max_j(G_j)$$

Step 4.5: **Update** Ψ_J and Ψ_G

if $G_{\max} > 0$, $\Psi_J = G_{\max}$
elseif $J \geq 0$, $\Psi_J = -1/(J+1)$
else $\Psi_J = 0$

$$\Psi_G = J + \textstyle\sum_{j=1}^{c_e} r_{i,j} G_j(\mathbf{u}_i(k)) + \sum_{j=c_e+1}^{c} r_{i,j} \max\left(0, G_j(\mathbf{u}_i(k))\right)$$

Until: $\Psi_J < \Psi_{J,init}$ or $\Psi_G < \Psi_{G,init}$ or the maximum number of iterations is reached.

Appendix D
Fuzzy model linearization for predictive control

The local linear state-space model to be used in the MPC algorithms in Chapter 4 is extracted by freezing the parameters of the fuzzy model at a certain operating point. This point can be the current one or along the trajectory predicted with the fuzzy (nonlinear) model. Given the operating point $\mathbf{y}(k)$ and $\mathbf{u}(k-1)$ in the first iteration or $\mathbf{u}(k)$ afterwards, the local model is calculated as follows.

Comparing the time-varying representation (2.6) of the TS fuzzy model given on page 17 with the LTV state-space model (4.7) on page on page 66, the state $\mathbf{x}_{\text{lin}}(k)$ contains the regression vectors $\mathbf{x}_l(k)$ for the separate MISO submodels. The elements in $\mathbf{x}_{\text{lin}}(k)$ are reordered such that delayed outputs from all $\mathbf{x}_l(k)$, $l = 1,\ldots,p$ come first and then delayed inputs. The matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$ are in the standard controllable canonical form with the parameters $\boldsymbol{\zeta}_l\big(\boldsymbol{\chi}_l(k)\big)$, $\boldsymbol{\eta}_l\big(\boldsymbol{\chi}_l(k)\big)$ and $\theta_l\big(\boldsymbol{\chi}_l(k)\big)$ positioned such that they reflect the structure of the state vector. Note that for notational simplicity $\boldsymbol{\zeta}_l^* = \boldsymbol{\zeta}_l\big(\boldsymbol{\chi}_l(k)\big)$, $\boldsymbol{\eta}_l^* = \boldsymbol{\eta}_l\big(\boldsymbol{\chi}_l(k)\big)$ and $\theta_l^* = \theta_l\big(\boldsymbol{\chi}_l(k)\big)$ in $\mathbf{A}(k)$, $\mathbf{B}(k)$ and $\mathbf{C}(k)$.

$$\mathbf{A}(k) = \begin{bmatrix} \zeta_{1,1}^* & \zeta_{1,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{1,\rho}^* & \theta_1^* \\ 1 & 0 & \cdots & & \cdots & & 0 & 0 \\ 0 & 1 & \vdots & & \ddots & & 0 & 0 \\ \vdots & \vdots & \ddots & & \ddots & & \vdots & \vdots \\ \zeta_{2,1}^* & \zeta_{2,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{2,\rho}^* & \theta_2^* \\ \vdots & \vdots & \ddots & & \ddots & & \vdots & \vdots \\ \zeta_{p,1}^* & \zeta_{p,2}^* & \cdots & \cdots & \cdots & \cdots & \zeta_{p,\rho}^* & \theta_p^* \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{D.1}$$

$$\mathbf{B}(k) = \begin{bmatrix} \eta^*_{1,1} & \eta^*_{1,2} & \cdots & \eta^*_{1,m} \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ \eta^*_{2,1} & \eta^*_{2,2} & \cdots & \eta^*_{2,m} \\ 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \eta^*_{p,1} & \eta^*_{p,2} & \cdots & \eta^*_{p,m} \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{D.2}$$

$$\mathbf{C}(k) = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{D.3}$$

As an example, a local state-space model is extracted from a 2×2 fuzzy model:

$\mathcal{R}_{1,i}$: **If** $y_1(k)$ is $\mathcal{A}_{1i,1}$ **and** $y_1(k-2)$ is $\mathcal{A}_{1i,2}$ **and** $u_1(k)$ is $\mathcal{A}_{1i,3}$ **and** $u_1(k-1)$ is $\mathcal{A}_{1i,4}$ **and** $u_2(k)$ is $\mathcal{A}_{1i,5}$ **and** $u_2(k-1)$ is $\mathcal{A}_{1i,6}$ **then**

$$y_1(k+1) = \underbrace{\zeta_{1i,1}y_1(k) + \zeta_{1i,2}y_1(k-2) + \zeta_{1i,3}u_1(k-1) + \zeta_{1i,4}u_2(k-1)}_{\mathbf{x}_1(k)} + \underbrace{\eta_{1i,1}u_1(k) + \eta_{1i,2}u_2(k)}_{\mathbf{u}(k)} + \theta_{1i} \tag{D.4a}$$

$\mathcal{R}_{2,i}$: **If** $y_2(k)$ is $\mathcal{A}_{2i,1}$ **and** $y_2(k-2)$ is $\mathcal{A}_{2i,2}$ **and** $u_1(k)$ is $\mathcal{A}_{2i,3}$ **and** $u_1(k-1)$ is $\mathcal{A}_{2i,4}$ **and** $u_2(k)$ is $\mathcal{A}_{2i,5}$ **and** $u_2(k-1)$ is $\mathcal{A}_{2i,6}$ **then**

$$y_2(k+1) = \underbrace{\zeta_{2i,1}y_2(k) + \zeta_{2i,2}y_2(k-2) + \zeta_{2i,3}u_1(k-1) + \zeta_{1i,4}u_2(k-1)}_{\mathbf{x}_2(k)} + \underbrace{\eta_{1i,1}u_1(k) + \eta_{1i,2}u_2(k)}_{\mathbf{u}(k)} + \theta_{1i} \tag{D.4b}$$

$i = 1, \ldots, K_l, \quad l = 1, 2.$

The state, input and output vectors of the linear state-space model are

$$\begin{aligned} \mathbf{x}_{\text{lin}}(k) &= [y_1(k), y_1(k-1), y_1(k-2), y_2(k), y_2(k-1), y_2(k-2), \ldots \\ &\quad u_1(k-1), u_2(k-1), 1]^T \\ \mathbf{u}(k) &= [u_1(k), u_2(k)]^T \\ \mathbf{y}_{\text{lin}}(k) &= [x_{\text{lin},1}(k), x_{\text{lin},4}(k)]^T . \end{aligned} \tag{D.5}$$

The $\mathbf{C}(k)$ matrix is thus

$$\mathbf{C}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The $\mathbf{B}(k)$ matrix contains η_l^*'s corresponding to $u_1(k)$ and $u_2(k)$:

$$\mathbf{B}(k) = \begin{bmatrix} \eta_{1,1}^* & \eta_{1,2}^* \\ 0 & 0 \\ 0 & 0 \\ \eta_{2,1}^* & \eta_{2,2}^* \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The ones in the 7th and 8th rows store the current inputs in the state vector in the positions for the delayed inputs, making them available at time $k+1$ as $\mathbf{u}(k-1)$.

The rows in $\mathbf{A}(k)$ correspond to the state vector in (D.5) – outputs, inputs and offset, while the columns reflect the structure of fuzzy rule consequents. Thus the first row correspond to $y_1(k)$, the second to $y_1(k-1)$, third to $y_1(k-2)$, fourth to $y_2(k)$, fifth to $y_2(k-1)$ and the sixth to $y_2(k-2)$. The elements in the first and the fourth row are positioned accordingly to (D.4): in the first row $\zeta_{1,1}^*$ multiplies $y_1(k)$, $\zeta_{1,2}^*$ multiplies $y_1(k-2)$, $\zeta_{1,3}^*$ multiplies $u_1(k-1)$ and $\zeta_{1,4}^*$ multiplies $u_2(k-1)$, in the fourth $\zeta_{2,1}^*$ multiplies $y_2(k)$, $\zeta_{2,2}^*$ multiplies $y_2(k-2)$, $\zeta_{2,3}^*$ multiplies $u_1(k-1)$ and $\zeta_{2,4}^*$ multiplies $u_2(k-1)$ correspondingly. The ones in the second, third, fifth and sixth rows arrange the states for the next sample.

$$\mathbf{A}(k) = \begin{bmatrix} \zeta_{1,1}^* & 0 & \zeta_{1,2}^* & 0 & 0 & 0 & \zeta_{1,3}^* & \zeta_{1,4}^* & \theta_1^* \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_{2,1}^* & 0 & \zeta_{2,2}^* & \zeta_{2,3}^* & \zeta_{2,4}^* & \theta_2^* \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that the above linearization method differs from the standard Jacobian linearization (Johansen et al., 2000). Recall the input-output representation of the TS

fuzzy model (2.6)

$$\begin{aligned}\mathbf{y}_l(k+1) &= \frac{\sum_{i=1}^{K_l}\beta_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k))\cdot\left(\boldsymbol{\zeta}_{li}\boldsymbol{\xi}(k)+\boldsymbol{\eta}_{li}\mathbf{u}(k)+\theta_{li}\right)}{\sum_{i=1}^{K_l}\beta_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k))} \\ &= \sum_{i=1}^{K_l}\omega_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k))\cdot f_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big), \qquad \text{(D.6)}\end{aligned}$$

where

$$\omega_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k)) = \frac{\beta_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k))}{\sum_{i=1}^{K_l}\beta_{li}(\boldsymbol{\xi}(k),\mathbf{u}(k))}$$

is the normalized degree of fulfillment and

$$f_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big) = \boldsymbol{\zeta}_{li}\boldsymbol{\xi}(k)+\boldsymbol{\eta}_{li}\mathbf{u}(k)+\theta_{li}$$

is the linear model of the ith fuzzy rule.

Applying the Jacobian linearization to (D.6), we have

$$\begin{aligned}\frac{\partial\mathbf{y}_l(k+1)}{\partial\boldsymbol{\xi}(k)} &= \sum_{i=1}^{K_l}\left(\frac{\partial\omega_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big)}{\partial\boldsymbol{\xi}(k)}\cdot f_{li}\big(\boldsymbol{\xi}(k)\big)+\omega_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big)\cdot\frac{\partial f_{li}\big(\boldsymbol{\xi}(k)\big)}{\partial\boldsymbol{\xi}(k)}\right) \\ \frac{\partial\mathbf{y}_l(k+1)}{\partial\mathbf{u}(k)} &= \sum_{i=1}^{K_l}\left(\frac{\partial\omega_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big)}{\partial\mathbf{u}(k)}\cdot f_{li}\big(\mathbf{u}(k)\big)+\omega_{li}\big(\boldsymbol{\xi}(k),\mathbf{u}(k)\big)\cdot\frac{\partial f_{li}\big(\mathbf{u}(k)\big)}{\partial\mathbf{u}(k)}\right)\end{aligned}$$

Comparing this result with (2.9) and (2.10), one can see that the terms containing the derivatives of the membership functions (degrees of fulfillment) are missing in these expressions. We can thus conclude that rather than a variant of the standard linearization, our approach is parameter scheduling that relies on the assumption that the individual rules of the TS model are good local linear models of the system. The linear parameter-varying model obtained at each operating condition then 'tracks' the nonlinear process dynamics in a way similar to using a linear adaptive model to track nonlinear dynamics. This is in contrast with the 'global approach', where the fuzzy rules typically do not have any local interpretation and the emphasis is on the global approximation accuracy. For the global approach, the standard Jacobian linearization would be more appropriate. For more details see (Johansen et al., 2000; Abonyi and Babuška, 2000). A clear advantage of our approach is that the time-consuming computation of the membership function derivatives is avoided.

Appendix E
$\mathcal{L}_1$-control theory

E.1 Discrete-time linear systems

A Linear Time-Invariant (LTI) discrete-time system is defined by its state-space representation

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{E.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the input vector and $\mathbf{y} \in \mathbb{R}^p$ is the output vector. The system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, of compatible size, are fixed in time and describe the behaviour of the system. The notation $\mathbf{G}(z) = \mathbf{G}\big(z\mathbf{I} - \mathbf{A}\big)^{-1}\mathbf{B} + \mathbf{D}$ is also used for representing an LTI system. The symbol z is the argument of the Z-transform of the system $\mathbf{G}$.

A discrete-time LTI system is stable if $\mathbf{A}$ has all its eigenvalues strictly within the unit disc.

A transfer function is *proper* if $\mathbf{G}(\infty)$ is bounded. A system is *inversely stable*, if $\mathbf{G}^{-1}(z)$ is proper and stable.

A Linear Time-Varying (LTV) system has time-varying system matrices and is defined by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k). \end{aligned} \tag{E.2}$$

E.2 Lebesgue spaces

Consider a discrete-time signal $\mathbf{u}(k) \in \mathbb{Z}$ defined in the interval $[0, \infty)$. Restrict $\mathbf{u}(k)$ to be square-Lebesgue integrable

$$\sum_{k=0}^{\infty} \mathbf{u}(k)^T \mathbf{u}(k) < \infty.$$

The set of all such signals is the Lebesgue space denoted by $l_2^n[0, \infty)$ or just by l_2.

E.3 Norms

In the scalar case the gain of the system $\mathbf{G}(z)$ at a specific frequency ω is given by the absolute value of $\mathbf{G}(e^{j\omega T_s})$. For multiple-input multiple-output systems this gain is not a single value but should be rather viewed as a range of gains. Therefore matrix norms are used induced by the corresponding vector norms.

Let X be a vector space, then a real-valued function $\|\cdot\|$ defined on X is said to be a norm on X if it satisfies the following properties:

1. $\|\mathbf{u}\| \geq 0$ (positivity);
2. $\|\mathbf{u}\| = 0$ if and only if $\mathbf{u} = 0$ (positive definiteness);
3. $\|\alpha\mathbf{u}\| = \alpha\|\mathbf{u}\|$, for any scalar α (homogeneity);
4. $\|\mathbf{u}+\mathbf{y}\| \leq \|\mathbf{u}\| + \|\mathbf{y}\|$ (triangle inequality)

for any $\mathbf{u} \in X$ and $\mathbf{y} \in X$.

The p-norm of the vector $\mathbf{u} \in \mathbb{R}^m$ is defined as

$$\|\mathbf{u}\|_p = \left(\sum_{i=1}^{m} |u_i|^p\right)^{1/p}, \quad \text{for } 1 \leq p \leq \infty.$$

For $p = 1, 2, \infty$ we have

$$\begin{aligned}
\|\mathbf{u}\|_1 &= \sum_{i=1}^{m} |u_i|; \\
\|\mathbf{u}\|_2 &= \sqrt{\sum_{i=1}^{m} |u_i|^2} = \sqrt{\mathbf{u}^T\mathbf{u}} \quad \text{(Euclidean norm)}; \\
\|\mathbf{u}\|_\infty &= \max_{1\leq i\leq m} |u_i|.
\end{aligned}$$

A norm of a vector is a measure of the vector "length", for example $\|\mathbf{u}\|_2$ is the Euclidean distance of the vector $\mathbf{u}$ from the origin. A similar kind of measure can be introduced for a matrix.

The matrix norm of the matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m\times n}$, *induced* by a vector p-norm is defined as

$$\|\mathbf{A}\|_p = \sup_{\mathbf{u}\neq 0} \frac{\|\mathbf{A}\mathbf{u}\|_p}{\|\mathbf{u}\|_p}.$$

For $p = 1, 2, \infty$, the corresponding induced matrix norm can be computed as

$$\begin{aligned}
\|\mathbf{A}\|_1 &= \max_{1\leq i\leq n} \sum_{i=1}^{m} |a_{ij}| \quad \text{(column sum)} \\
\|\mathbf{A}\|_2 &= \sqrt{\lambda_{\max}(\mathbf{A}^T\mathbf{A})} \\
\|\mathbf{A}\|_\infty &= \max_{1\leq i\leq m} \sum_{j=1} n|a_{ij}| \quad \text{(row sum)}
\end{aligned}$$

where $\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ is the maximal eigenvalue of the product $(\mathbf{A}^T\mathbf{A})$.

The matrix norm induced by vector p-norms are called *induced p-norms*, because they are defined, or induced, from a vector p-norm. Actually $\mathbf{A}$ can be viewed as a mapping from a vector space $\mathbb{R}^n$ equipped with a vector norm $\|\cdot\|_p$ to another vector space $\mathbb{R}^m$ equipped with a vector norm $\|\cdot\|_p$. So from a system theoretical point of view, the induced norms can be interpreted as input-output amplification gains.

E.4 Operators

A discrete-time operator $\mathbf{G}$ is a function from one signal space $\mathbf{u} \in l_2^m$ to another $\mathbf{y} \in l_2^p$:

$$\mathbf{y} = \mathbf{G}(\mathbf{u}).$$

The operator is linear if

$$\mathbf{G}(\mathbf{u}_1 + \mathbf{u}_2) = \mathbf{G}(\mathbf{u}_1) + \mathbf{G}(\mathbf{u}_2)$$
$$\mathbf{G}(\alpha\mathbf{u}) = \alpha\mathbf{G}(\mathbf{u})$$

where $\alpha \in \mathbb{R}$. Linear operator are, for instance, the linear systems. An operator is casual if $(\mathbf{Gu})(k)$ only depends on past values of $\mathbf{u}$. The l_2-induced norm for $\mathbf{G}$ is defined as

$$\|\mathbf{G}\| = \sup_{\mathbf{u}\in\mathcal{L}_2, \mathbf{u}\neq 0} \frac{\|\mathbf{G}(\mathbf{u})\|_2}{\|\mathbf{u}\|_2}.$$

E.5 Stability

We have already defined stability for linear time-invariant systems in terms of the eigenvalues of the $\mathbf{A}$-matrix. We will now extend the definition to uncertainty systems. Consider the feedback configuration shown in Fig. E.1. The stable LTI system $\mathbf{G}(z) \in l_2^{n\times n}$ is interconnected with the causal operator $\Omega : l_2^n \to l_2^n$ with a bounded l_2-induced norm.

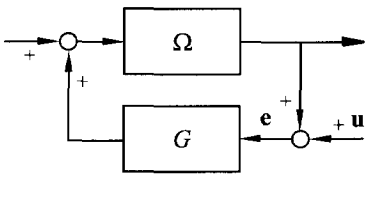


Figure E.1. Illustration of a dynamic system with uncertainty feedback.

The closed-loop system response from $\mathbf{u}$ to $\mathbf{e}$ is given by

$$\mathbf{e} = \big(\mathbf{I} - \Omega\mathbf{G}(z)\big)^{-1}\mathbf{u}.$$

We say that the closed-loop system $\big(\Omega, \mathbf{G}(z)\big)$ is bounded-input bounded-output (or BIBO) stable if $\mathbf{e} \in l_2^n$ for all $\mathbf{u} \in l_2^n$. A related definition is the following.

The feedback system defined by the pair $(\Omega, \mathbf{G}(z))$ is *well posed* if the operator $(\mathbf{I} - \Omega\mathbf{G}(z))$ is causally invertible, i.e., there exists a casual $\mathbf{F} : l_2^n \to l_2^n$ such that $\mathbf{F}(\mathbf{I} - \Omega\mathbf{G}(z)) = (\mathbf{I} - \Omega\mathbf{G}(z))\mathbf{F} = \mathbf{I}$. Assume that $\mathcal{D}$ is a set of operators $\mathcal{D} \subset \{\Omega : \Omega \in l_2^n \to l_2^n\}$. The system $(\mathcal{D}, \mathbf{G}(z))$ is *robustly stable* if $(\Omega, \mathbf{G}(z))$ is well-posed for every $\Omega \in \mathcal{D}$.

The well-known *small-gain theorem* (Vidyasagar, 1993) states that, assuming Ω and $\mathbf{G}(z)$ are stable operators, the closed-loop system $(\Omega, \mathbf{G}(z))$ is stable is $\|\mathbf{G}(z)\| \leq \gamma < 1$ and $\|\Omega\| \leq 1$. This follows, e.g., from the definition of well-posedness, since

$$\left\|\left(\mathbf{I} - \Omega\mathbf{G}^{-1}(z)\right)\right\| = \left\|\sum_{k=0}^{\infty} \left(\Omega\mathbf{G}(z)\right)^k\right\| \leq \sum_{k=0}^{\infty} \left\|\Omega\mathbf{G}(z)\right\|^k \leq \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma} < \infty.$$

Thus the gain of the closed-loop system is always bounded.

Appendix F
Distillation Columns

F.1 Distillation principle

Distillation columns operate on the principle that the vapor of a boiling mixture will be richer in the components that have lower boiling points. Therefore, when this vapor is cooled and condensed, the condensate will contain more volatile (light) components. The remaining mixture will contain more of the less volatile material.

Distillation columns are made up of several units which are used either to transfer heat energy or enhance material transfer: (i) *a vertical shell* where the separation of liquid components takes place; (ii) column stages such as *trays* or *plates* used to enhance component separations; (iii) *a reboiler* that vaporizes part of the liquid leaving the bottom of the column; (iv) *a condenser* used to cool and condense the vapor leaving the top of the column; (v) *a reflux drum* where the condensed vapor from the top of the column is stored so that liquid (reflux) can be recycled back to the column. The vertical shell contains the column trays and together with the condenser and the reboiler constitutes the distillation unit. A schematic diagram of a typical distillation unit with a single feed and two product streams is shown in Fig. 7.1.

The liquid and the vapor are in countercurrent contact inside the column: the liquid flows down and the vapor flows up. At each distillation stage, some of the vapor moving up the column is condensed and this in turn evaporates some of the liquid moving downwards. With two components in the processed mixture, a greater amount of the heavier (less volatile) component condenses and a greater amount of the lighter (more volatile) component evaporates at each tray.

The mixture containing the components to be separated is known as *feed*. The feed can be in any state from a cold liquid to a superheated vapor. It is usually introduced somewhere near the middle of the column to a tray known as *feed tray*. The feed tray divides the column into a *rectifying* (top) section and a *stripping* (bottom) section. The rectifying section comprises the stages above the feed point, where the concentration of the lighter component increases in both the liquid and the vapor. The stripping section comprises the stages below the feed point, where the concentration of the lighter component decreases in both the liquid and the vapor.

The *bottom liquid*, with the least volatile components in the feed, flows from the base of the column to *a reboiler* (Fig. F.1). The reboiler is a heat exchanger, where heat is used to vaporize some of the liquid that flows back up the column in countercurrent flow with the liquid level moving down the column. Although the source of heat input can be any suitable medium, in most chemical plants this is normally steam. The amount of the heat fed to the reboiler determines the vapor flow up the column. The liquid removed from the reboiler is known as the *bottom product* or simply, bottoms.

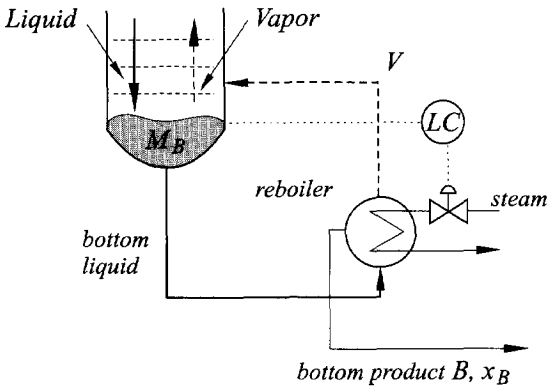


Figure F.1. Distillation column bottom operation.

The *overhead vapor* with the most volatile components in the feed moves from the top of the column to *a condenser* (Fig. F.2). This condenser is another heat exchanger, where cooling water is used to condense the vapor to a liquid. The condensed liquid is stored in a holding vessel known as the *reflux drum* or *reflux holdup*. The liquid leaving the reflux drum is split into two parts: (i) the reflux flow which is fed back to the column, where it moves downwards in countercurrent flow with the vapor flowing up the column and (ii) the condensed liquid which is removed from the system, known as the *distillate* or top product.

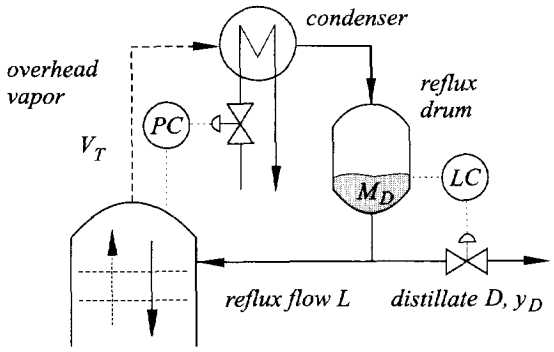


Figure F.2. Distillation column top operation.

The operation of a distillation column is affected by various factors disturbing the performance, e.g., feed conditions (feed rate, feed composition), internal liquid and fluid flow conditions, state of trays (packings), weather conditions, etc. The feed rate and feed composition affect the number of stages required for the separation. They also affect the location of the feed tray. To overcome the problems associated with the feed, some columns have multiple feed points when the feed is expected to containing varying amounts of components. Since distillation columns are open to the atmosphere, changing weather conditions can affect column operation although many of the columns are insulated. Therefore, the reboilers and the condensors must be appropriately sized to ensure that enough vapor can be generated during cold and windy operation and that it can be sufficiently cooled down during hot seasons.

F.2 Justification of closed-loop data generation

Because of the phenomenon of directionality (Section 7.1), open-loop identification only excites the process in the high-gain direction. To demonstrate why this happens, an analysis for the linear case is given. (Linear analysis suffices because the TS fuzzy model comprises linear models derived for different operating regions.) Consider the transfer function matrix $\mathbf{G}(z)$ from the inputs reflux and reboiler flows L and V, respectively, to the outputs top and bottom compositions y_D and x_B, respectively. The frequency response $\mathbf{G}(j\omega)$ has singular value decomposition (Andersen and Kümmel, 1992)

$$\mathbf{G}(j\omega) = [\mathbf{U}_1(j\omega)\ \mathbf{U}_2(j\omega)] \begin{bmatrix} \sigma_1(\omega) & 0 \\ 0 & \sigma_2(\omega) \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^*(j\omega)\ \mathbf{V}_2^*(j\omega) \end{bmatrix},$$

where $*$ denotes complex conjugate transpose and $\sigma_1(\omega) \geq \sigma_2(\omega)$. The vectors $\mathbf{U}_i$ and $\mathbf{V}_i$ specify the principal input and output directions, respectively and the singular values σ_i specify the principal gains of the process. Most distillation columns are ill-conditioned in the sense that $\sigma_1(\omega) \gg \sigma_2(\omega)$ over a large frequency range. Denoting the spectral densities of the input and output vectors by $\Phi_{\mathbf{u}}(\omega)$ and $\Phi_{\mathbf{y}}(\omega)$, we have

$$\Phi_{\mathbf{y}}(\omega) = \mathbf{G}(j\omega)\Phi_{\mathbf{u}}(\omega)\mathbf{G}^*(j\omega).$$

If the proces inputs are two independent pseudo random binary signals (PRBS) with a unit variance, then $\Phi_{\mathbf{u}}(\omega)$ is an identity matrix and

$$\Phi_{\mathbf{y}}(\omega) = [\mathbf{U}_1(j\omega)\ \mathbf{U}_2(j\omega)] \begin{bmatrix} \sigma_1^2(\omega) & 0 \\ 0 & \sigma_2^2(\omega) \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^*(j\omega) \\ \mathbf{V}_2^*(j\omega) \end{bmatrix}.$$

The interpretation of the last expression is that the output of the column has a significant component in the $\mathbf{U}_1$ direction, that is the high-gain direction. This means that by manipulating the flows L and V, one product becomes more pure and the other less pure. It is hard to make both products purer at the same time which corresponds to increasing the flows with L/V constant (low-gain). Hence, open-loop experiments do not give information about the low-gain direction, i.e., when both products are getting more pure or less pure simultaneously (Fig. 7.3, top).

Therefore, to get both low-gain and high-gain directions well excited, a closed-loop experiment is carried out. Let $\mathbf{K}$ and $\mathbf{r}$ denote the feedback controller and a set-point output reference, respectively. The frequency content of the input is given by

$$\mathbf{u}(j\omega) = (\mathbf{I} + \mathbf{K}(j\omega)\mathbf{G}(j\omega))^{-1}\mathbf{K}(j\omega)\mathbf{r}(j\omega).$$

At frequencies where the gain of the controller is (sufficiently) high, the above relation can be approximated by

$$\mathbf{u}(j\omega) \approx \mathbf{G}^{-1}(j\omega)\mathbf{r}(j\omega) = [\mathbf{V}_1(j\omega)\ \mathbf{V}_2(j\omega)] \begin{bmatrix} \frac{1}{\sigma_1(\omega)} & 0 \\ 0 & \frac{1}{\sigma_2(\omega)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^*(j\omega) \\ \mathbf{U}_2^*(j\omega) \end{bmatrix}.$$

Since $\sigma_1(\omega) \gg \sigma_2(\omega)$, the input is likely to contain mainly the low-gain component.

Appendix G
Symbols and abbreviations

Printing Conventions. Lower case characters in bold print denote column vectors. For example, $\mathbf{x}$ and $\boldsymbol{\eta}$ are column vectors. A row vector is denoted by using the transpose operator, for example $\mathbf{x}^T$ and $\boldsymbol{\eta}^T$. Lower case characters in italics denote elements of vectors and scalars. Upper case bold characters in bold print denote matrices, for instance, $\mathbf{X}$ is a matrix. Upper case calligraphic characters such as $\mathcal{A}$ denote crisp and fuzzy sets or input-output mappings, e.g., $\mathcal{F}$ depending on the context. Upper case italic characters denote domains, such as X.

No distinction is made between variables and their values, hence x may denote a variable or its value, depending on the context. No distinction is made either between a function and its value, e.g., $s_{u_j(k)}^{y_l(k+1)}$ may denote both an output sensitivity function and its value. When a signal or a system is time-dependent, the time index is included in round brackets, e.g., $\mathbf{u}(k)$.

General mathematical symbols

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	system matrices in state space
$\mathbf{u}(k) \in \mathbb{R}^m$	input of a dynamic system at time k
$\mathbf{y}(k) \in \mathbb{R}^p$	output of a dynamic system at time k
$\mathcal{A}, \mathcal{B}$	fuzzy sets
K	number of rules in a rule base
$\mathcal{R}$	fuzzy rule
X, Y	domains (universes) of variables x and y
$\boldsymbol{\zeta}, \boldsymbol{\eta}, \boldsymbol{\theta}$	consequent parameters in a TS model
$\mathbf{x}_l$	state vector in the lth MISO model
ρ	dimension of the state vector $\mathbf{x}_l$
$\boldsymbol{\chi}_l(k)$	regression vector in the lth MISO model
$\vartheta = \rho + m$	dimension of the regression vector $\boldsymbol{\chi}_l(k)$
μ	membership degree
β	degree of fulfillment of a rule
w	normalized degree of fulfillment
$\mathbb{Z}$	set of integer numbers
$\mathbb{R}$	set of real numbers

$\mathbb{C}$	set of complex numbers
$\mathbf{I}$	identity matrix of appropriate dimensions

Symbols related to fuzzy clustering

$\mathbf{\Phi}$	regressor matrix
$\mathbf{\Gamma}$	regressand vector
$\mathbf{F}$	cluster covariance matrix
$\mathbf{E}$	norm-inducing matrix
$\mathbf{U} = [\mu_{ik}]$	fuzzy partition matrix
$\mathbf{V}$	matrix containing cluster prototypes (means)
$\mathbf{Z}$	data (feature) matrix
c	number of clusters
$d(\cdot,\cdot)$	distance measure
m	weighting exponent (determines fuzziness of the partition)
$\mathbf{v}$	cluster prototype (center)
$\mathbf{z}$	data vector
$\mu_{i,k}$	membership of data vector $\mathbf{z}_k$ into cluster i

Symbols related to (robust) fuzzy predictive control

Δ	signal increment
$\mathbf{f}$	gradient of the Lagrange function
$\mathbf{Hs}$	Hessian of the Lagrange function
H_p	prediction horizon
H_c	control horizon
H_min	minimum cost horizon
J	cost function
L	Lagrange function
λ	Lagrange multipliers
$M(k+i)$	LTV model extracted from the fuzzy model at time $(k+i)$
$\mathcal{M}$	LTV system comprising the LTV models $M(k+i)$
$\mathcal{U}, \mathcal{Y}$	predicted input and output trajectories
ϵ_Ω, ϵ_{Ω_Δ}	uncertainty upper bound
$\mathbf{g}(k)$	model offset at time k
$\gamma_p(G)$	l_p-gain of system G
$\mathbf{H}_G(k-\tau)$	kernels (Markov parameters) of system G
l_p	Lebesque space
$\Omega(k)$	model uncertainty at time k
$\Omega_\Delta(k)$	model uncertainty increment at time k
$N(G)$	order of system G

Operators:

$\mathbf{X}^T$	transpose of matrix $\mathbf{X}$
∂	partial derivative
det, $\lvert\cdot\rvert$	determinant of a matrix
diag	diagonal matrix
$.*$	element by element multiplication (Hadamard or Schur product)

$\text{rank}(\mathbf{X})$	rank of matrix $\mathbf{X}$
sup	suprenum
∇	*(vector) gradient*
$\|\cdot\|_p$	p-norm of a signal
$\|\cdot\|_{i,p}$	induced p-norm of a system

Abbreviations

B&B	branch-and-bound technique
BIBO	bounded-input bounded output (stability)
CSTR	continuous stirred tank reactor
d.c.	direct current
DOF	degree of fulfillment
FLC	fuzzy logic control
FLOP	floating point operations
FM	fuzzy model
GDI	gasoline direct injection (engine)
GK	Gustafson–Kessel algorithm
LMI	linear matrix inequalities
LS	line search (in MBPC)
LTI	linear time-invariant
LTV	linear time-varying
MIMO	multiple-input, multiple-output
MISO	multiple-input, single-output
MM	multiple model method (in MPC)
MPC	model predictive control
(N)ARX	(nonlinear) autoregressive with exogenous inputs
PDC	parallel distributed compensation
PID	proportional integral derivative (controller)
PRBS	Pseudo-random binary signal
QP	quadratic programme
RGA	relative gain array
RMS	root mean square
SISO	single–input, single–output
SM	single model method (in MBPC)
SQP	sequential quadratic programming
STD	standard deviation
TLS	total least-squares method
TS	Takagi–Sugeno type fuzzy model
VAF	variance accounted for

Abbreviations related to GDI engine

EOI	End of injections
IGA	Ignition advance angle
M_{air}	Mass of air in the cylinder
M_{fuel}	Mass of fuel in the cylinder
MTC	Air throttle command
N	Engine speed

N_{idle}	Idle engine speed
P_m	Manifold pressure
Q_{air}	Air flow in the cylinder
Q_{fuel}	Fuel flow in the cylinder
RAF	Air-fuel ratio
SOI	Start of injection
T_{inj}	Injection time
TDC	Top dead center
TQE	Effective torque
TQI	Indicated torque
TQL	Torque losses
ϕ_{MTC}	Air throttle position

Summary

The continual enhancement of the quality and performance of industrial processes is a primary target in the academics and industry alike. The development of new methods for control design and their utilization in the increasingly sophisticated control systems is the key to controlling the more and more sophisticated processes.

The classical linear control theory is based on greatly simplifying assumptions concerning the nature and behavior of the controlled process. In practice, however, these assumptions do not hold for complex, multivariable, not very well understood, or partially unknown systems. Hence, methods are sought which can cope with such systems.

The use of intelligent control systems is more and more often used for applications where the process to be controlled is multivariable, nonlinear and/or time varying, and for which mathematical models are difficult to obtain or to use. Automotive and chemical industries are typical examples of such applications. Multivariable processes pose difficult control problems because of their complex behavior and interacting phenomena such as directionality and ill-conditioning. Moreover, the imposed economical goals are usually conflicting. Depending on the specific process and goals, we can have a centralized control system based on a MIMO controller, or a decentralized control system using a number of SISO controllers. Multivariable controllers usually outperform decomposed controllers, but this gain in performance must be traded off against the cost of obtaining and maintaining a sufficiently detailed model.

Since its introduction in 1965, fuzzy set theory has found its application in the control systems theory, not only from the scientific community but also from industry. The interest in fuzzy algorithms for control is based on the fact that many systems are not amenable to conventional control approaches due to the lack of precise, formal knowledge about the system, due to strongly nonlinear behavior, due to the high degree of uncertainty, or due to the time varying characteristics.

We put the main accent of this thesis on the development of methods for analysis and design of control structures based on Takagi–Sugeno fuzzy models. As preliminary material, state-space and input-output forms of the TS model are presented and the inference mechanism in the Takagi–Sugeno model is discussed. For completeness of the presentation, a brief overview of the mainstream of the research efforts directed

to the state feedback and output feedback control using TS models is included. Along with this presentation, we show some of the shortcomings of the approach using TS models such as handling the offset term, or having estimated variables in the rule antecedents. As shown later, the methods proposed in this thesis can overcome solve some of these shortcomings.

In the methodology developed in this thesis, we use information derived from the TS model locally at a certain point to design off-line or on-line effective control structures, as opposed to utilizing the TS model as a global nonlinear model. This approach allows us to approximate the nonlinearity specific for a given region by linear dynamics valid only in this region. By analyzing the approximated linear dynamics we can derive properties of the original nonlinear process model. The Relative Gain Array (RGA) approach for analysis of input-output interactions has been extended to deal with TS models, in order to provide a number of arrays that can sufficiently well indicate the interactions throughout the model. Additional insight in the interactions can be gained by using the output sensitivity analysis. While the extended fuzzy RGA analysis provided an indication for the static interactions, the output sensitivity analysis can be used also to give insight in the dynamic interactions. Based on the obtained knowledge and the design specifications, we can decide whether or not a decoupler is needed that can reduce or completely remove the undesired coupling.

The same local linear approach can be used for the decoupler design. We redesign the decoupler on-line, at each sampling instant, taking into account the fuzzy rules which are currently active. If the fuzzy model is an affine one with at least as many inputs as outputs, an analytical decoupling law is be obtained. For non-affine fuzzy models, the inversion is accomplished by inverting the input-output transfer function matrix. The decoupling based on the local interpretation of TS models can be achieved computationally very efficiently. This allows for their use in situations where, due to short sampling times, more computationally involved control methods cannot be properly utilized. An alternative approach is to invert the TS model as a whole by means of a nonlinear optimization technique. This approach is called numeric decoupling. Numeric decoupling, despite of being the most general method, is computationally demanding, which limits its application in fast processes.

An important part of this thesis is devoted to the formulation of Model Predictive Control (MPC) algorithms which can be efficiently solved on-line based on TS models. By analyzing the structure of the resulting optimization problem, we propose a formulation which has a complexity comparable to linear MPC algorithms and, at the same time, significantly improves the performance. We use of the nonlinear TS model available to obtain predicted process input and output trajectories. By linearizing the TS model along these trajectories, a set of linear models is derived that can approximate the process behaviour over the prediction horizon. The accuracy of prediction can be further improved by iterating the above procedure to compensate for possible linearization errors. The convergence of the iterative scheme is guaranteed by a line search mechanism that considers reduction both in the cost function and in the constraints. The methods for MPC presented in this thesis are especially suited for applications with nonlinear multivariable processes. They can be used when the sam-

pling interval is short. For such processes, linear model predictive control does give poor performance, while nonlinear predictive control is too time consuming.

The feasibility of the optimization problem on the one hand, and the use of the Internal Model Control (IMC) scheme on the other hand, guarantee the stability of the closed-loop system in the nominal case i.e., when the process model is an accurate approximation of the process. However, a certain model-plant mismatch will always be present due to unmodeled dynamics, time-varying aging phenomena, etc. This model-plant mismatch will not only deteriorate the control performance; it may even destabilize the closed-loop system. A reasonable approach is to consider the process deviation from the available model as a model uncertainty, such that the process behavior is always contained within the set of behaviors described by the model combined with the associated uncertainty. Then the robust stability of the closed loop system is ensured by means of extra constraints on the control signal that guarantee stability for any model-plant mismatch within certain given bounds. To derive these bounds, the TS model is viewed as a linear time-varying model rather than as a nonlinear one. We demonstrated that the stability constraints robustify the system performance in the case of a model-plant mismatch without deteriorating the nominal performance. The stability constraints, rather than the weights in the cost function smooth out the control signal. The application of the stability constraints, however, is computationally demanding.

We can conclude that the methodologies proposed in this thesis offer novel tools for the analysis and design of control structures for complex MIMO processes. This has been demonstrated by several simulation studies and real-time laboratory experiments. An application of fuzzy MPC to a simulation model of a GDI engine illustrates the utilization of a data-driven TS model in a control structure comprising a multivariable MPC optimizer and a complex switching logic. A real-time application of the fuzzy MPC methods is demonstrated on a laboratory-scale cascaded-tanks system. This setup is also used to test the robust stability constrains in real time. The smoothing effect of the stability constraints has been demonstrated. The techniques for analysis of input-output interactions and input-output decoupling have been illustrated on a simulation model of a binary distillation column and again on the laboratory cascaded-tanks setup.

Finally, the main contributions of this thesis to the field of fuzzy MIMO control design can be summarized as follows. New methods have been proposed for the analysis of input-output coupling and interactions, and for decoupling design. An effective formulation of the optimization problem in fuzzy model predictive control has been devised. Constraints on the control signal in MPC have been introduced which guarantee closed-loop robust asymptotic stability for open-loop BIBO stable processes with an additive l_1-norm bounded model uncertainty.

Samenvatting

De continue verbetering van de kwaliteit en prestatie van industriële processen staat voorop bij zowel de academische als de industriële wereld. De ontwikkeling van nieuwe regelmethoden en het gebruik hiervan voor het ontwerpen van steeds meer geperfectioneerde regelsystemen is essentieel voor het regelen van steeds ingewikkelder processen.

De klassieke lineaire regeltheorie is gebaseerd op sterk vereenvoudigde veronderstellingen over de eigenschappen en het gedrag van het te regelen proces. In de praktijk gaan deze veronderstellingen echter niet op voor complexe, multivariabele, slecht begrepen of gedeeltelijk onbekende systemen. Daarom is men op zoek naar methodes die zulke systemen aankunnen.

Steeds vaker worden intelligente regelsystemen gebruikt voor toepassingen waar het te regelen proces multivariabel, niet-lineair, en/of variabel in de tijd is, en waarvoor wiskundige modellen moeilijk te verkrijgen of te gebruiken zijn. De auto industrie en de chemische industrie zijn typische voorbeelden van zulke toepassingen. Multivariabele processen geven aanleiding tot moeilijke regelproblemen vanwege hun complexe gedrag en hun wisselwerking met fenomenen als richtingsafhankelijkheid en slechte geconditioneerdheid. Daarnaast zijn de beoogde economische doelen meestal in strijd met elkaar. Afhankelijk van het specifieke proces en de doelen is sprake van een gecentraliseerd regelsysteem dat is gebaseerd op een MIMO systeem, of van een gedecentraliseerd regelsysteem dat een aantal SISO systemen gebruikt. Multivariabele regelaars presteren over het algemen beter dan separaat ontwikkelde regelaars, maar deze verbetering in prestatie moet worden afgewogen tegen de kosten van het bepalen en onderhouden van een model dat voldoende gedetailleerd is.

Sinds haar introductie in 1965 is de fuzzy set theorie toegepast in de theorie van regelsystemen, zowel in de academische wereld als in de industrie. De belangstelling voor fuzzy regel-algoritmen komt door het feit dat de conventionele regelmethodes vaak niet gebruikt kunnen worden voor een systeem omdat er geen precieze, formele kennis over het systeem is, het systeem sterk niet-lineair gedrag vertoont, er een hoge mate van onzekerheid is, of het systeem in de tijd variërende eigenschappen heeft.

In dit proefschrift kijken we vooral naar het ontwikkelen van methodes voor het analyseren en ontwerpen van regelstructuren die zijn gebaseerd op Takagi–Sugeno

fuzzy modellen. Er wordt uitgegaan van state-space en input-output vormen van het TS model en de berekeningsmethode van het Takagi–Sugeno model wordt besproken. Voor de volledigheid geven we ook een kort overzicht van de belangrijkste onderzoeken naar state feedback en output feedback regelaars waarbij gebruikt wordt gemaakt van TS modellen. Ook laten we een paar tekortkomingen zien van de aanpak met TS modellen zoals het omgaan met de "offset term", of met geschatte variabelen in de antecedents waaraan de regels die het systeem beschrijven moeten voldoen. Zoals we later laten zien, kunnen de methodes die we hier voorstellen sommige van deze tekortkomingen oplossen.

In de onderzoeksmethode die we in dit proefschrift ontwikkelen, gebruiken we informatie die plaatselijk op een bepaald punt van het TS model is afgeleid om off-line of on-line effectieve regelstructuren te ontwerpen, in plaats van dat we het TS model als een globaal niet-lineair model gebruiken. Deze aanpak maakt het mogelijk om de niet-lineariteit die specifiek is voor een bepaald gebied te benaderen met lineaire dynamica die alleen voor dit gebied geldt. Door de benaderde lineare dynamica te analyseren kunnen we karakteristieken van het originele niet-lineaire procesmodel afleiden. De "Relative Gain Array" (RGA) benadering voor het analyseren van input-output interacties is uitgebreid om TS modellen aan te kunnen, om zo een aantal rijen te verkrijgen die voldoende goed kunnen aangeven welke interacties in het model plaatsvinden. Extra inzicht in de interacties kan worden verkregen door de "output sensitivity" te analyseren. Terwijl de uitgebreide fuzzy RGA analyse alleen de statische interacties kan aangeven, kan de output sensitivity analyse ook worden gebruikt om inzicht te krijgen in de dynamische interacties. Op grond van de verkregen kennis en de ontwerpspecificaties kunnen we beslissen of er al dan niet een ontkoppelaar nodig is die ongewenste koppeling kan verminderen of geheel opheffen.

Dezelfde plaatselijk lineaire aanpak kan worden gebruikt voor het ontwerp van een ontkoppelaar. We ontwerpen de ontkoppelaar opnieuw on-line, tijdens iedere bemonsteringsperiode, rekening houdend met de fuzzy regels die op dat moment van kracht zijn. Als het fuzzy model affien is met tenminste evenveel inputs als outputs dan volgt er een analytische ontkoppelingswet. Voor niet-affiene fuzzy modellen verkrijgen we de inversie door de input-output matrix van overdrachtsfuncties te inverteren. De ontkoppeling gebaseerd op de plaatselijke interpretatie van TS models kan efficiënt worden berekend. Dit maakt het mogelijk ze te gebruiken in situaties waar, door de korte tijd tussen de bemonsteringen, rekentijd-intensievere regelmethoden niet goed kunnen worden toegepast. Een andere benadering is het TS model in zijn geheel inverteren door middel van een niet-lineaire optimalisatietechniek. Deze methode wordt numerieke ontkoppeling genoemd. Alhoewel deze methode het meest algemeen wordt gebruikt, is hij ook reken-intensief, waardoor hij minder geschikt is voor snelle processen.

Een belangrijk deel van dit proefschrift wordt gewijd aan het formuleren van Model Predictive Control (MPC) algoritmen die efficiënt on-line kunnen worden berekend op basis van TS modellen. We analyseren de structuur van het resulterende optimalisatieprobleem en op basis hiervan formuleren we een benadering die ongeveer net zo complex is als die van lineaire MPC algoritmen, maar tegelijkertijd veel beter presteert. We gebruiken het niet-lineaire TS model om voorspelde input en output

trajectorieën te verkrijgen. Door het TS model langs deze trajectorieën te linearis-eren, leiden we een set lineaire modellen af die het gedrag van het proces binnen de voorspellingshorizon kunnen benaderen. De nauwkeurigheid van de voorspelling kan verder worden verbeterd door bovenstaande procedure te itereren om mogelijke linearisatiefouten te corrigeren. De convergentie van het iteratie schema wordt gegarandeerd door een line-search mechanisme dat reductie bewerkstelligt in zowel de kostenfunktie als in de restricties. De methodes voor MPC die in dit proefschrift worden gepresenteerd zijn speciaal geschikt voor toepassingen met niet-lineaire multivariabele processen. Ze kunnen worden gebruikt wanneer er weinig tijd zit tussen de bemonsteringstijdstippen. Voor zulke processen geeft voorspellend regelen gebaseerd op lineaire modellen slechte resultaten, terwijl niet-lineair voorspellend regelen teveel tijd kost.

De haalbaarheid van het optimalisatieprobleem aan de ene kant en het gebruik van het "Internal Model Control" (IMC) schema aan de andere kant garanderen de stabiliteit van het gesloten-lus systeem in een ideaal geval, dat wil zeggen, wanneer het procesmodel een accurate benadering is van het te regelen proces. Er zal echter altijd een zekere discrepantie zijn tussen model en systeem door ongemodelleerde dynamica, in de tijd veranderende verouderingsverschijnselen, etc. Het verschil tussen model en systeem kan niet alleen resulteren in een slechtere regelprestatie; het kan zelfs zorgen voor destabilisatie van het gesloten-lus systeem. Een redelijke aanpak is om de afwijking van het proces van het beschikbare model als een onzekere factor in het model te beschouwen, zodat het gedrag van het proces altijd valt binnen de set gedragingen die het model beschrijft samen met de daarmee geassocieerde onzekerheid. Dan wordt de robuuste stabiliteit van het gesloten-lus systeem verzekerd door middel van extra restricties op het regelsignaal die stabiliteit garanderen voor welk verschil tussen model en systeem dan ook binnen zekere gegeven grenzen. Om deze grenzen af te leiden, beschouwen we het TS model als een lineair in de tijd variërend model in plaats van als een niet-lineair model. We hebben aangetoond dat de stabiliteitsrestricties de prestatie van het systeem robuuster maken wanneer er een verschil is tussen model en systeem zonder dat de nominale prestatie wordt beinvloed. De stabiliteitsrestricties, en niet zozeer het wegen van factoren in de kostenfunctie, zorgen dat problemen met het regelsignaal worden opgelost. De toepassing van de stabiliteitsrestricties is echter rekenintensief.

We kunnen concluderen dat met de methodes die in dit proefschrift worden voorgesteld, regelstructuren voor complexe MIMO processen op een nieuwe manier kunnen worden geanalyseerd en ontworpen. Dit is aangetoond met verschillende simulaties en real-time laboratoriumexperimenten. De toepassing van een fuzzy MPC op een simulatiemodel van een GDI motor illustreert het gebruik van een data-gestuurd TS model in een regelstructuur bestaand uit een multivariabele MPC optimizer en complexe schakel-logica. Een real-time toepassing van de fuzzy MPC methodes is beschreven uitgaande van een cascaded-tanks systeem op laboratoriumschaal. Deze opstelling is ook gebruikt om de robuuste stabiliteitsrestricties in real-time te testen. Het effect die de stabiliteitsrestricties op het regelgedrag hebben is aangetoond. De technieken voor het analyseren van input-output interacties en input-output ontkop-

peling zijn geïllustreerd met een simulatiemodel van een binaire distillatiekolom en daarna met een opstelling van cascaded tanks in het laboratorium.

Tenslotte vatten we de belangrijkste bijdragen tot het gebied van fuzzy MIMO regelontwerp samen. Nieuwe methodes zijn gepresenteerd voor het analyseren van input-output koppeling en interacties, en voor het ontwerp van de ontkoppelaar. Een effectieve formulering van het optimalisatieprobleem in voorspellend regelen met gebruik van fuzzy modellen is gegeven. Restricties voor het regelsignaal in MPC zijn geïntroduceerd die gesloten-lus robuuste asymptotische stabiliteit garanderen voor open-lus BIBO stabiele processen met een extra onzekerheidsfactor die het model begrenst: de l_1 norm.

Curriculum Vitae

Stanimir Mollov was born on April 10, 1972 in Sliven, Bulgaria. In 1995 he graduated from the Technical University (TU) in Sofia, with specialization in systems and control engineering. In 1994 he worked six months as programmer at Great Bear Inc., Sofia and from 1995 till 1998 was employed in the Department of Information Technologies of the government Agency for Economic Analysis and Forecasting. In 1997, he started his doctoral research program which was focused on control of time-varying processes at the Department of Control Systems at the Faculty of Automatics, TU Sofia. In the period between March 1998 and February 2002, he was associated with the Control Laboratory at the Department of Information Technologies and Systems of the Delft University of Technologies as a research assistant (AIO) and Ph.D. student in the field of fuzzy control. Since March 2002 he has been working at FCS Control Systems B.V.

He is married to Vania and they have two children, Michail and Simona.

Acknowledgments

The research reported in this thesis was supervised by Prof. Henk Verbruggen and Prof. Robert Babuška at the Control Laboratory, Faculty of Information Technology and Systems, Delft University of Technology. To both of them I wish to express my gratitude for their support and guidance, numerous discussions and suggestions.

Parts of this thesis are based on articles co-authored by Ton van den Boom of the Control Laboratory, TU Delft, by Hans Roubos, currently with Halotec B.V., by Peter van der Veen, currently with Huisman-Itrec, by Janos Anonyi of University of Veszprém, Hungary, and by Aníbal Ollero and Federico Cuesta Rojo, Universidad de Sevilla, Spain. The application reported in Chapter 6 was realized in cooperation with Siemens Automotive SA, Toulouse, France.

I would like to express my sincere thanks to all my colleagues and friends who contributed with their comments and suggestions on part of this work. Among them, a special word of thanks to Bart de Schutter, Marcel Oosterom, Bart Wams and Hayco Bloemen, currently with TPD-TNO. I also want to thank to Dan Noteboom and Will van Geest for their resourcefulness regarding the software and hardware support. Last, but not least, I am indebted to Mrs. M. J. Nieman for revisions to my English text.

I am grateful to my wife, Vania, for her understanding, patience and continual support during the entire period of my Ph.D. research and especially in difficult mounts of writing the manuscript.

References

Abonyi, J. and R. Babuška (2000). Local and global identification and interpretation of parameters in Takagi-Sugeno fuzzy models. *Proceedings of FUZZ-IEEE*, San Antonio, Texas, USA, pp. 835–840. IEEE.

Abonyi, J., R. Babuška, M. Setnes, H. B. Verbruggen and S. Szeifert (1999). Constrained parameter estimation in fuzzy modeling. *Proceedings of FUZZ-IEEE'99*, Seoul, Korea, pp. 951–956.

Abonyi, J., L. Nagy and F. Szeifert (2001). Fuzzy model based predictive control by instantaneous linearization. *Fuzzy Sets and Systems 120*(1), 109–122.

Andersen, H. W. and M. Kümmel (1992). Evaluating estimation of gain directionality. part 1: Metodology. *Journal of Process Control 2*(2), 59–66.

Åström, K. J. and B. Wittenmark (1989a). *Adaptive Control*. Addison.

Åström, K. J. and B. Wittenmark (1997b). *Computer-Controlled Systems: Theory and Design*. Upper Saddle River, New Jersey 07458: Prentice Hall. Third Edition.

Babuška, R. (1998). *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, MA, USA.

Babuška, R., C. Fantuzzi, U. Kaymak and H. B. Verbruggen (1996). Improved inference for takagi-sugeno models. *Proceedings of FUZZ-IEEE'96*, New Orleans, USA, pp. 701–706.

Babuška, R. and H. B. Verbruggen (1995). Identification of composite linear models via fuzzy clustering. *Proceedings of ECC'95*, Rome, Italy, pp. 1207–1212.

Bard, Y. (1974). *Nonlinear Parameter Estimation*. New York, USA: Academic Press.

Bemporad, A., F. Borrelli and M. Morari (2001). Robust model predictive control: piecewise linear explicit solution. *Proceedings of ECC*. Session We-M09.

Bergsten, P. (2001). *Observers and Controllers for Takagi-Sugeno Fuzzy Systems*. PhD dissertation, Örebro University, Örebro, Sweden.

Bergsten, P. and R. Palm (2000a). Sliding mode observers for TS fuzzy system. *Proceedings of FUZZ-IEEE*, San Antonio, Texas. IEEE. Session TP1-2.

Bergsten, P. and R. Palm (2000b). Thau-luenberger observers for TS fuzzy systems. *Proceedings of FUZZ-IEEE*, San Antonio, Texas. IEEE. Session TP1-3.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function*. Plenum Press, New York.

Bloemen, H. H. J., C. T. Chou, T. J. J. van den Boom, V. Verdult, M. Verhaegen and T. C. Backx (2001). Wiener model identification and predictive control for dual composition control of a distillation column. *Journal of Process Control 11*(6), 601–620.

Bogatyrev, A. V. and E. S. Pyatnitskii (1987). Construction of piecewise-quadratic lyapunov functions for nonlinear control systems. *Automation and Remote Contol 48*(10), 1292–1299.

Boggs, P. T., A. J. Kearsley and J. W. Tolle (1999). A global convergence analysis of an algorithm for large-scale nonlinear optimization problems. *SIAM Journal of Optimization 9*(4), 833–862.

Boggs, P. T. and J. W. Tolle (1995). Sequential quadratic programming. *Acta numerica 4*, 1–52.

Bortolet, P., D. Passaquay and S. Boverie (1998). Engine benchmark architecture decomposition. Technical report, Siemens Automotive, Toulouse, France.

Botto, M. A., T. J. J. van den Boom, A. Krijgsman and J. Sa da Costa (1999). Predictive control based on neural network models with i/o feedback linearization. *International Journal of Control 72*(17), 1538–1554.

Boverie, S. (2000). Final control results for the engine benchmark. FAMIMO deliverable 4.1 justification and presentation of hybrid fuzzy controllers in real conditions. Technical report, Siemens Automotive S.A.

Boyd, S. P., L. E. Ghaoui, E. Feron and V. Balakrishnan (1994). *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia: SIAM.

Bristol, E. H. (1966). On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control 11*, 133–134.

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms. *J. Inst. Maths. Applics 6*, 76–90.

Bryson, A. E. and Yu-Chi Ho (1981). *Applied Optimal Control. Optimization, Estimation and Control*. Taylor&Francis; Hemishepe publishing Corporation. Revised printing.

Campo, P. J. and M. Morari (1987). Robust model predictive control. *Proceedings of ACC*, Volume 2, pp. 1021–1026.

Chen, H.-G. and K.-W. Han (1994). Improved quantitative measures of robustness for multivarible systems. *IEEE Transactions on Automatic control 39*(4), 807–810.

Chou, C. T., H. H. J. Bloemen, V. Verdult, T. J. J. van den Boom, T. Backx and M. Verhaegen (2000). Nonlinear identification of high purity identification columns. *Proceedings of the Symposium on System Identification*, Santa Barbara, CA, USA.

Clarke, D. W. and R. Scattolini (1991). Constrained receding horizon predictive control. *IEE Proc-D 138*(4), 347–354.

Commission, EU (1993). Commission directive 93/116/ec of 17 december 1993 adapting to technical progress council directive 80/1268/eec relating to the fuel consumption of motor vehicles. *Official Journal L 329*, 39–53. Web adress http://europa.eu.int/eur-lex/en/lif/reg/en_register_133010.html, or http://www.environment.fgov.be/Root/tasks/atmosphere/klim/pub/wet/eu/93-116-EG_en.htm.

Coleman, T., M. A. Branch and A. Grace (1999). *Optimization Toolbox For Use with* MATLAB. The MathWorks.

Cuesta, F., F. Gordillo, J. Aracil and A. Ollero (1999). Stability analysis of nonlinear multivariable Takagi–Sugeno fuzzy control systems. *IEEE Transactions on Fuzzy Systems 7*(5), 508–520.

De Schutter, B. and T. J. J. van den Boom (2000a). Model predictive control for max-plus-linear systems. *Proceedings of American Control Conference*, Chicago, Illinois, pp. 4046–4050.

De Schutter, B. and T. J. J. van den Boom (2000b). Model predictive control for max-min-plus systems. R. Boel and G. Stremersch (Eds.), *Discrete Event Systems: Analysis and Control, vol. 569 of The Kluwer International Series in Engineering and Computer Science*, pp. 201–208. Boston, USA: Kluwer Academic Publishers.

Driankov, D., H. Hellendoorn and M. Reinfrank (1993). *An Introduction to Fuzzy Control*. Springer, Berlin.

Economou, C. G., M. Morari and B. O. Palsson (1986). Internal model control: Extension to non-linear systems. *Ind. Eng. Chem. Process Des. Dev. 25*, 403–411.

Espinosa, J. J., M. L. Hadjili, V. Wertz and J. Vandewalle (1999). Predictive control using fuzzy models - comparative study. *Proceedings of ECC*, pp. F0547.

Farinwata, S. (2000). An approach to to the analysis of robust stability of fuzzy control systems. S. Farinwata, D Filev and R. Langari (Eds.), *Fuzzy Control. Synthesis and Analysis*, pp. 165–202. Chichester, England: Jhon Wiley & Sons, Ltd.

Farinwata, S., D Filev and R. Langari (Eds.) (2000). *Fuzzy Control. Synthesis and Analysis*. Chichester, England: Jhon Wiley & Sons, Ltd.

Fischer, M., O. Nelles and R. Isermann (1998b). Adaptive predictive control of a heat exchanger based on a fuzzy model. *Control Engineering Practice 6*(2), 259–269.

Fischer, M., O. Nelles and R. Isermann (1998a). Predictive control based on local linear fuzzy models. *International Journal of Systems Science 29*(7), 679–697.

Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal 13*, 317–332.

Fliess, M., M. Lamnabhi and F. Lamnabhi-Lagarrigue (1983). An algebraic approach to nonlinear functional. *IEEE Transactions on Circuits and Systems 30*, 554–570.

Fuentes, C. and W.-L. Luyben (1983). Control of high-purity distillation columns. *Industrial and Engineering Chemistry, Process Design and Development 25*(3), 335–348.

Gäfvert, M., L. M. Pedersen, K.-E. Årzén and B. Bernhardsson (2000). Simple feedback contol and mode switching strategies for GDI engines. *Proceedings of SAE World Congress*. SAE Paper 2000–01–0263.

Ganesan, V. (1994). *Internal Combustion Engines*. McGraw-Hill.

García, C. E., D. M. Prett and M. Morari (1989). Model predictive control: Theory and practice – a survey. *Automatica 25*(3), 335–348.

Gegov, A., R. Babuška and H. B. Verbruggen (1999). Linguistic analysis of interactions in MIMO fuzzy systems. *Proceedings of 14th IFAC World Congress*, Volume K, Bejing, China, pp. 249–254.

Gerkšič, S., D. Juričić, S. Strmčnik and D. Matko (2000). Wiener model based nonlinear predictive control. *International Journal of Systems Science 31*(2), 189–202.

Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computing 24*, 23–26.

Golub, G. H. and C. F. van Loan (1996). *Matrix Computations*. Baltimor and London: The Jonhs Hopkins University Press.

Goodwin, G. C., S. F. Graebe and Salgado M. E. (2001). *Control System Design*. Upper Saddle River, New Jersey 07458: Prentice Hall.

Grossdidier, P., M. Morari and B. R. Holt (1985). Closed-loop properties from steady-state gain information. *Ind. and Eng. Chem. Process. Design and Development 24*, 221–235.

Gunawardena, J. (1994). Cycle times and fixed points of min-max functions. G. Cohen and J. Quadrat (Eds.), *Proceedings of the 11th International Conference on Analysis and Optimization of Systems*, pp. 266–272. London, UK: Springer-Verlag. vol. 199 of Lecture Notes in Control and Information Sciences.

Gustafson, D. E. and W. C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. *Proc. IEEE CDC*, San Diego, CA, USA, pp. 761–766.

Hoekstra, P., T. J. J. van den Boom and Botto M. A. (2001). Design of an analytic constrainted predictive controller using neural networks. *Proceedings of European Control Conference*. EUCA. Paper ID Fr-M09-03.

Hovd, M. and S. Skogestad (1992). Simple frequency-dependent tools for control system analysis, structure selection, and design. *Automatica 28*, 989–996.

Hu, J. Q. and T. Rose (1999). Generalized predictive control using a neuro-fuzzy model. *Int. J. of Systems Science 30*(1), 117–122.

Huang, Y. L., H. H. Lou, J. P. Gong and E. F. Thomas (2000). Fuzzy model predictive control. *IEEE Transactions on fuzzy systems 8*(6), 665–678.

Hui, L. C. (1983). *General Decoupling Theory of Multivariable Process Control Systems*. Lecture Notes in Control and Information Series No. 53. Springer-Verlag, Berlin Heidelberg.

Isidori, A. (1995). *Nonlinear Control Systems*. Communications and Control Engineering Series. Berlin: Springer–Verlag, 3 ed.

Johansen, T. A. (1994a). Fuzzy model based control: stability, robustness, and performance issues. *IEEE Transactions on Fuzzy Systems 2*, 221–234.

Johansen, T. A. (1994b). *Operating Regime Based Process Modelling and Identification*. PhD dissertation, The Norwegian Institute of Technology – University of Trondheim, Trondheim, Norway.

Johansen, T. A., R. Shorten and R. Murray-Smith (2000). On the interpretation and identification of dynamic TakagiŰ-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems 8*(3), 297–313.

Johansson, M., A. Rantzer and K.-E. Årzen (1999). Piecewise quadratic stability of fuzzy systems. *IEEE Transactions on Fuzzy Systems 7*(6), 713–722.

Kacprzyk, J. (1997). *Multistage Fuzzy Control: A Model-Based Approach to Fuzzy Control and Decision Making*. Chichester, New York, USA: John Wiley & Sons.

Kang, F. and Y. Lei (1996). Application of multivariable fuzzy control in heating system of injection moulding machine. *Proceedings of the IEEE International Conference on Industrial Technology*, New York, NY, USA, pp. 603–606. IEEE.

Kavsek, B. K., I. Skrjanc and D. Matko (1997). Fuzzy predictive control of a highly nonlinear pH process. *Computers Chem. Eng. 21*, S613–618.

Keerthi, S. S. and E. G. Gilbert (1988). Optimal infinite-horizon feedback laws for a general class of contrained discrete-time systems: stability and moving-horizon approximations. *Journal of Optimization Theory and Applications 57*, 265–293.

Khalil, H. K. (1992). *Nonlinear systems*. New York, USA: Macmillan.

Kim, J.-H. (1995). Comments on "improved quantitative measures of robustness for multivarible systems". *IEEE Transactions on Automatic control 40*(9), 1619–1620.

Klir, G. J. and T. A. Folger (1988). *Uncertainty and information*. New Jersey: Prentice Hall.

Korba, P. (2000). *A Gain-scheduling approach to model-based fuzzy control*. PhD dissertation, Gerhard Mercator University of Duisburg, Duisburg, Germany.

Kothare, M. V., V Balakrishnan and M. Morari (1996). Robust constrained predictive control using linear matrix inequalities. *Automatica 32*(10), 1361–1379.

Kotta, Ü. (1995). *Inversion Method in the Discrete-time Nonlinear Control Systems Synthesis Problems. Lecture Notes in Control and Information Sciences*, Volume 205. Berlin: Springer-Verlag.

Kuhn, H. and A. W. Tucker (1951). Nonlinear programming. *Proceedings of Second Berkeley Symposium of Mathematical Statistics and Probability*. University of California Press.

Kwakernaak, H. and R. Sivan (1991). *Modern Signals and Systems*. Englewood Cliffs, NJ 07632, USA: Prentice Hall Information and System Science Series.

Lee, J. H. and Z. H. Yu (1994). Tuning of model predictive controllers for robust performance. *Computers Chem. Eng. 18*(1), 15–37.

Lewis, F. L. (1995). *Optimal Control* (Second ed.). New York: Wiley.

Maciejowski, J. M. (1989). *Multivariable feedback design*. Great Britain: Addison-Wesley.

Maciejowski, J. M. (2002). *Predictive control with constraints*. Harlow, Essex, England: Prentice-Hall.

Mamdani, E. H. (1974). Applications of fuzzy algorithms for control of simple dynamic plant. *Proceedings IEE*, Volume 121, pp. 1585–1588.

Mamdani, E. H. (1977). Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems 26*, 1182–1191.

Mamdani, E. H. and S. Assilian (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies 7*, 1–13.

Marino, R. and P. Tomei (1995). *Nonlinear control design: geometric, adaptive, and robust*. Hemel Hempstead, Hertfordshire, England: Prentice-Hall.

Masubuchi, I., A. Kume and E. Shimemura (1998). Spline-type solution to parameter dependent lmis. *Proceedings of the 37th IEEE Conference on Decision and Control*, pp. 1753–1758. IEEE.

Mayne, D. Q. (2001). Control of constrained dynamic systems. *European Journal of Control 7*(2-3), 87–99.

Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica 36*, 789–814.

McAvoy, T. J., E. Hsu and S. Lowenthal (1972). Dynamics of pH in controlled stirred tank reactor. *Ind. Eng. Chem Process Des. Develop. 11*(1), 68–70.

Michalska, H. and D. Q. Mayne (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control 38*(11), 1623–1633.

Mollov, S. and R. Babuška (2002). Analysis of interactions and multivariable fuzzy control design for a binary distillation column. *International Journal of Fuzzy Systems*. Submitted.

Mollov, S., R. Babuška, J. Abonyi and H. B. Verbruggen (2002). Effective optimization for fuzzy model predictive control. *IEEE Transactions on fuzzy systems*. Submitted.

Mollov, S., R. Babuška and P. van der Veen (2001). Nonlinear model based predictive control of a GDI engine. *Benelux Quarterly Journal of Automatic Control 42*(1), 31–37.

Mollov, S., R. Babuška and H. B. Verbruggen (2000a). Decoupling MBPC using fuzzy models. *Proceedings of FUZZ-IEEE*, San Antonio, Texas, USA, pp. 417–422. IEEE.

Mollov, S., R. Babuška and H. B. Verbruggen (2001b). Analysis of interactions in MIMO Takagi–Sugeno fuzzy models. *Proceedings of FUZZ-IEEE*, Melbourne, Australia. IEEE. Session T4d-6.

Mollov, S., T. J. J. van den Boom, F. Cuesta, A. Ollero and R. Babuška (2002). Robust stability constraints for fuzzy model predictive control. *IEEE Transactions on fuzzy systems 10*(1), 50–64.

Mollov, S., J. A. Roubos, R. Babuška and H. B. Verbruggen (1998a). MIMO predictive control by multiple-step linearization of Takagi–Sugeno fuzzy models. *Proceedings of AIRTC*, Arizona, US. Session I.

Mollov, S., J. A. Roubos, R. Babuška and H. B. Verbruggen (1999b). Predictive control by multiple-step linearization of Takagi–Sugeno fuzzy models. Y. H. Pao and S. R. LeClair (Eds.), *Artificial Intelligence in Real-Time Control 1998*, IFAC Proceedings Volumes, pp. 197–202. Pergamon Press.

Mollov, S., P. van der Veen and R. Babuška (2001). Nonlinear model based predictive control of a GDI engine. *Proceedings of European Control Conference*, Porto, Portugal. Session WE-ISO4-03.

Mollov, S., P. van der Veen, R. Babuška, J. Abonyi, J. A. Roubos and H. B. Verbruggen (1999). Extraction of local linear models from Takagi–Sugeno fuzzy model with application to model-based predictive control. *Proceedings of EUFIT*, Aachen, Germany. Session BA 8.

Monaco, S. and D. Normand-Cyrot (1995). *Nonlinear Systems in Discrete-time. Lecture Notes in Control and Information Sciences*, Volume 83. Berlin: Springer-Verlag.

Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice Hall, New Jersey.

Mosca, E. and J. Zhang (1992). Stable redesign of predictive control. *Automatica 28*(6), 1229–1233.

Murray-Smith, R. and T. A. Johansen (Eds.) (1997). *Multiple Model Approaches to Nonlinear Modeling and Control*. London, UK: Taylor & Francis.

Na, M. G. (1998). Design of a genetic fuzzy controller for the nuclear steam generator water level control. *IEEE Transactions on Nuclear Science 45*(4), 2261–2271.

Nevistić, V. and M. Morari (1995). Constrained control of feedback-linearizable systems. *Proceedings of 3rd European Control Conference*, pp. 1726–1731.

Nichols, R. A., R. T. Reichert and W. J. Rugh (1993). Gain cheduling for H_∞ controllers for a MS760 Paris aircraft. *IEEE Trans. on Control Systems Technology 1*, 69–75.

Nijmeijer, H. and A. J. van der Schaft (1990). *Nonlinear Dynamical Control Systems*. New York, USA: Springer-Verlag.

Norquay, S. J., A. Palazoglu and J. A. Romagnoli (1998). Model predictive control based on Wiener models. *Chemical Engineering Science 53*(1), 75–84.

Nounou, H. N. and K. M. Passino (1999). Fuzzy model predictive control: techniques, stability issues, and examples. *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics*, pp. 423–428.

Oliveira, de, S. L., V. Nevistić and M. Morari (1995). Control of nonlinear systems subject to input constraints. *Proceedings of Nonlinear Control Design Symp., NOLCOS*, Volume 1, pp. 15–20.

Oliveira, de, V. J. and J. M. Lemos (2000). A comparison of some adaptive-predictive fuzzy-control strategies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 30*(1), 138–145.

Olsder, G. (1994). On structural properties of min-max systems. G. Cohen and J. Quadrat (Eds.), *Proceedings of the 11th International Conference on Analysis and Optimization of Systems*, pp. 237–246. London, UK: Springer-Verlag. vol. 199 of Lecture Notes in Control and Information Sciences.

Palm, R., D. Driankov and H. Hellendoorn (1997). *Model Based Fuzzy Control: Fuzzy Gain Schedulers and Sliding Mode Fuzzy Controllers*. Springer, Berlin.

Palm, R. and K. Storjohann (1997). Slip control and torque optimization using fuzzy logic. M. Jamshidi, A. Titli, S. Boverie and L. A. Zadeh (Eds.), *Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient Systems*, pp. 245–263. Prentice Hall, New York.

Parsini, T. and R. Zopolli (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica 31*, 1443–1451.

Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems (second, extended,edition)*. John Wiley and Sons, New York.

Piegat, A. (2001). *Fuzzy Modeling and Control*. Studies in fuzzyness and soft computing No. 69. Physica-Verlag.

Pottmann, M. and D. E. Seborg (1997). A nonlinear predictive control strategy based on radial basis function models. *Computers chem. Engng 21*, 965–980.

Powell, M. J. D. (1978). A fast algorithm for nonlinearly constrained optimization calculations. G. A. Watson (Ed.), *Numerical Analysis*, Lecture Notes in Mathematics, Vol. 630, pp. 144–157. Springer Verlag.

Powell, M. J. D. (1983). Variable metric methods for constrained optimization. A. Bachem, M. Grotschel and B. Korte (Eds.), *Mathematical programming: The state of the art*, pp. 288–311. Springer Verlag.

Pshenichnyj, B. N. (1994). *The Linearization Method for Constrained Optimization.* Heidelberg, Berlin, Germany: Springer-Verlag.

Rantzer, A. and M. Johansson (1997). Piecewise linear quadratic optimal control. *Proceedings of American Control Conference*, Volume 3, pp. 1749–1753.

Rawlings, J. B. and K. R. Muske (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control 38*, 1512–1516.

Reay, D. S., M. M. Mirkazemi, T. C. Green and B. W. Williams (1995). Switched reluctance motor control via fuzzy adaptive systems. *IEEE Control Systems Magazine 15*(3), 8–15.

Reichert, R. T. (1992). Dynamic scheduling of modern-robust-control autopilot designs for missiles. *IEEE Control Systems Magazine 12*(5), 35–42.

Rivera, D. E., M. Morari and S. Skogestad (1986). Internal model control. 4 pid controller design. *Ind. Eng. Chem. Process Des. Dev. 25*, 252–265.

Roubos, J. A., S. Mollov, R. Babuška and H. B. Verbruggen (1999). Fuzzy model based predictive control by using Takagi–Sugeno fuzzy models. *Int. Journal of Approximate Reasoning 22*(1), 3–30.

Rugh, W. J. (1991). Analytical framework for gain scheduling. *IEEE Control Systems Magazine 11*(1), 79–84.

Saez, D. and A. Cipriano (1997). Design of fuzzy model based predictive controllers and its application to an inverted pendulum. *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Volume 2, Barcelona, Spain, pp. 915–919.

Sain, M. K. and S. Yurkovich (1982). Controller scheduling: A possible algebraic viewpoint. *Proceedings of the American Control Conference*, pp. 261–269.

Scokaert, O. M. (1997). Infinite horizon generalized predictive controller. *Int. J. Control 66*(1), 161–175.

Scokaert, O. M., D. Q. Mayne and J. B. Rawlings (1999). Suboptimal model predictive controller (feasibility implies stability). *IEEE Transactions on Automatic Control 44*(3), 648–654.

Setnes, M., H. R. van Nauta Lemke and U. Kaymak (1998). Fuzzy arithmetic-based interpolative reasoning for nonlinear dynamic fuzzy systems. *Engineering Applications of Artificial Intelligence 11*, 781–789.

Shamma, J. S. and M. Athans (1990). Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control 35*(8), 898–907.

Shamma, J. S. and M. Athans (1991). Guaranteed properties of gain scheduled control of linear parameter-varying systems. *Automatica 27*(3), 559–564.

Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computing 24*(24), 647–656.

Shinskey, F. G. (1996). *Process Control Systems. Application, Design and Tuning. Forth edition.* New York, NY, USA: McGraw-Hill.

Skogestad, S. (1997). Dynamics and control of distillation columns. A tutorial introduction. *Chem. Eng. Res. Des. (Trans. IChemE) 75*, 539–562.

Skogestad, S. and M. Morari (1988). Understanding the dynamic behaviour of distillation columns. *Industrial and Engineering Chemistry Research 27*(10), 1848–1862.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control.* Chichester, England: John Wiley & Sons.

Soeterboek, A. R. M. (1992). *Predictive Control; A Unified Approach.* Prentice Hall.

Sousa, J. M. (1998). *Fuzzy Model-Based Control.* Ph. D. thesis, Delft University of Technology, Department of Electrical Engineering, Delft, The Netherlands.

Sriniwas, G. R., Y. Arkun, I.-L. Chien and B. A. Ogunnaike (1995). Nonlinear identification and control of a high-purity distillation column: a case study. *Journal of Process Control 5*(3), 149–162.

Stein, G. (1980). Adaptive flight control – a pragmatic view. K. S. Narendra and R. V. Monopoli (Eds.), *Applications of Adaptive control.* Academics, New York.

Sugeno, M. and K. Tanaka (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems 42*, 315–334.

Sun, J., I. Kolmanovski, D. Brehob, J. A. Cook, J Buckland and M. Haghgooie (1999). Modeling and control of gasoline direct injection stratified charge DIDC engines. *Proceedings of 1999 IEEE Conference on Control Applications*, pp. 471–477. IEEE.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics 15*, 116–132.

Tanaka, K., T. Ikeda and H. Wang (1998). Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based desings. *IEEE Transactions on Fuzzy Systems 6*(2), 250–264.

Tanaka, K., T. Kosaki and H. Wang (1997). Fuzzy control of an articulated vehicle and its stability analysis. *Proceedings of 13th Triennial World Congress*, pp. 115–120. IFAC.

Tanaka, K. and M. Sano (1994). A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer. *IEEE Transactions on Fuzzy Systems 2*(2), 119–134.

Tanaka, K. and M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems 45*(2), 135–156.

Titli, A. and S. Boverie (1997). Fuzzy control approach for the design of active and semi-active suspension (comparison with optimal control and variable structure control). M. Jamshidi, A. Titli, S. Boverie and L. A. Zadeh (Eds.), *Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient Systems*, pp. 25–48. Prentice Hall, New York.

Vidyasagar, M. (1993). *Nonlinear Systems Analysis.* Englewood Cliffs, NJ: Prentice Hall.

Vries, de, R. A. J. and T. J. J. van den Boom (1997). Robust stability constraints for predictive control. *Proceedings of European Control Conference*, Volume 6, (FR A B2).

Wang, H. O., K. Tanaka and M. F. Griffin (1996). An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Trans. on Fuzzy systems 4*, 14–23.

Whatley, M.-J. and D. C. Pott (1984). Adaptive gain aimproves reactor control. *Hydrocarbon precessing*, 75–78.

Yager, R. R. and D. P. Filev (1994). *Essentials of Fuzzy Modeling and Control.* John Wiley, New York.

Yaochu, J., J. Jingping and Z. Jing (1995). Adaptive fuzzy modelling and identification with its applications. *Int. J. Systems Science 26*, 197–212.

Yoneyama, J. and M. Nishikawa (2001). Design of output feedback controllers for Takagi-Sugeno fuzzy system. *Fuzzy Sets and Systems 121*(1), 127–148.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control 8*, 338–353.

Zadeh, L. A. (1968). Fuzzy algorithms. *Information and control 12*, 94–102.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man, and Cybernetics 1*, 28–44.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci. 8*, 199–257.

Zeng, A. and M. Morari (1993). Robust stability of constrained model predictive control. *Proceedings of American Control Conference*.

Zeng, A. and M. Morari (1994). Global stabilization of discrete-time systems with bounded controls - A model predictive control approach. *Proceedings of American Control Conference*, pp. 2847–2851.

Zeng, A. and M. Morari (1995). Robust stability of linear systems with constraints. Technical report, California Institute of Technology, Pasadena.

Zhao, J. (1995). *Fuzzy logic in modeling and control*. PhD dissertation, CESAME, Louvain la Neuve, Belgium.

Zhao, J., R Gorez and V. Wertz (1997). Synthesis of control systems based on linear Takagi–Sugeno fuzzy models. R. Murray-Smith and T. A. Johansen (Eds.), *Multiple Model Approaches to Nonlinear Modeling and Control*, pp. 307–336. London, UK: Taylor & Francis.

Zhou, K., J. C. Doyle and K. Glover (1995). *Robust and optimal control*. Upper Saddle River, New Jersey 07458: Prentice-Hall.

Zimmermann, H.-J. (1991). *Fuzzy Set Theory and its Application* (Second ed.). Boston: Kluwer.

Author Index

Subject Index