Fault Tolerant Control Using Evolving Fuzzy Modeling

Davyd da Cruz Chivala

Dissertação para obtenção do Grau de Mestre em

Engenharia Mecânica

Júri

Presidente: Professor João Rogério Caldas Pinto
Orientador: Professor João Miguel da Costa Sousa
Co-Orientador: Professor Luís Manuel Fernandes Mendonça
Vogal: Professor Paulo Jorge Sequeira Gonçalves

Maio - 2009
Da esperança de um amanhã melhor faço
minha parte hoje sempre com auxílio D’Aquele
Que nunca nos vira as costas.
Abstract

The main goal of this dissertation is to develop fault tolerant control approach using adaptive fuzzy modeling.

The distillation column is modeled, using Takagi-Sugeno fuzzy models. The modeling task is carried out using the fuzzy modeling toolbox developed by Robert Babuska [8]. The integration of the simulink model of the distillation column, and the fuzzy models in the model based predictive control is made.

Load process faults, specifically variation in feed composition and change in heating (variation of re-boiler temperature), are considered, these faults can be abrupt or incipient. Such faults has the consequence of changing the operational ranges. The fault detection is made when the residual (difference between the process output and the model output) passes a certain threshold.

To accommodate faults effects, two different approaches are used, evolving fuzzy modeling [4, 5, 6] and adaptive fuzzy models [9, 54]. Both methods are performed on-line with current data (data with fault information). In evolving fuzzy modeling, according to certain conditions, new rules are added or the existing ones are updated. When new rules are added, their consequents are determined by using the recursive least square (RLS) method. In the adaptive fuzzy modeling approach, only the consequents are recursively updated by using RLS. Finally the proposed approaches are applied to fault tolerant control of a distillation column.

**Keywords:** Fault tolerant control, fuzzy modeling, adaptive fuzzy models, evolving fuzzy modeling, model based predictive control, distillation column.
Resumo

O principal objectivo desta tese é o desenvolvimento de abordagens para controlo tolerante a falhas utilizando modelação fuzzy adaptiva.


Falhas na alimentação, nomeadamente variação da composição da alimentação e mudanças na potência especificamente variação da temperatura fornecida ao reboiler são consideradas, sendo estas abruptas ou incipientes. Tais falhas têm como consequência a mudança da zona operacional da coluna. A detecção da falha é feita quando o resíduo (diferença entre a saída do processo e a saída do modelo) ultrapassa um determinado valor limite de resíduo.

Para acomodar o efeito das falhas, duas abordagens são utilizadas, os "evolving fuzzy modeling" [4, 5, 6] e a modelação fuzzy adaptiva [9, 54]. Quando utilizadas a abordagem "evolving fuzzy modeling", novas regras são adicionadas ou, as já existentes actualizadas. Quando adicionadas novas regras, os seus consequentes são determinados utilizando o método dos mínimos quadrados recursivo. Na modelação fuzzy adaptiva apenas os consequentes são actualizados através do método dos mínimos quadrados recursivo. Finalmente as abordagens propostas são então aplicadas no controlo tolerante a falhas de uma coluna de destilação.
Palavras-chave: Controlo tolerante a falha, modelação fuzzy, modelação fuzzy adaptativa, "evolving fuzzy modeling", controlo predictivo baseado em modelos, coluna de destilação.
Acknowledgments

I would like to sincerely express my thanks to my supervisor Professor João M. Sousa.

I would like also to express my deepest gratitude to my co-supervisor Professor Luis Mendonça for his support and orientation on this work. For his permanent availability to analyze and discuss ideas.

To Professor Plamen Angelov, my gratitudes to provide some answers related to the evolving Takagi-Sugeno fuzzy models.

To Professors Carla Pinheiro and João Silva my sincere thanks to enlight me in distillation column issues.

I would like to thank the TOTAL E&P ANGOLA for the financial support provided to realize my studies.

I need to refer to my family, specially my mother and my girlfriend for their support in this work.

At last I would like to thank my colleagues from Department of Mechanical Engineering, for all the encouragement that I have received from.

For other people that directly or indirectly helped me in this work and are not here mentioned, my thanks.
# Contents

1 Introduction 1

1.1 Objective and outline of this work 2

2 Fault Tolerant Control 5

2.1 Introduction 5

2.2 Passive and active fault tolerant control systems 7

2.2.1 Passive fault tolerant control system 7

2.2.2 Active fault tolerant control system 8

2.3 Fault detection and Isolation 8

2.4 Model based predictive control in fault tolerant control 9

2.5 Evolving fuzzy modeling in fault tolerant control 9

3 Evolving Fuzzy Modeling 11

3.1 Introduction 11

3.2 Takagi-Sugeno fuzzy models 13
3.2.1 Off-line identification .................................................. 14
3.2.2 On-line adaption of the fuzzy models .............................. 19
3.3 Evolving fuzzy modeling .................................................. 21

4 Model Based Predictive Control ........................................... 29
4.1 Introduction ................................................................. 29
4.2 Classical model based predictive control ............................. 30
4.3 Fuzzy models in predictive control .................................... 32
4.4 Branch and bound in model based predictive control .......... 33
4.4.1 Fuzzy filters .............................................................. 37

5 Predictive Fault Tolerant Control Using Evolving Fuzzy Modeling .................................................. 41
5.1 Introduction ................................................................. 41
5.2 Proposed approaches for fault tolerant control .................... 42
5.2.1 Adaptive based approach in the predictive fault tolerant control ........................................ 44
5.2.2 Evolving fuzzy modeling in the predictive fault tolerant control ........................................ 44

6 Application On The Distillation Column ................................ 47
6.1 Introduction ................................................................. 47
6.2 Distillation column ........................................................ 47
6.3 Fuzzy modeling of the distillation column ............................ 54
6.4 Predictive control applied to the distillation column ............. 57
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control ........................................ 59
CONTENTS

6.5.1 Matrix of adaptive gain and forgetting factor effects ....... 59

6.5.2 Application of fault tolerant control to the distillation column ..... 60

7 Conclusions ....... 77

7.1 Fault tolerant control approaches ....... 77

7.2 Influence of the adaptive gain matrix and the forgetting factor ....... 78

7.3 Future work ....... 78

Bibliography ....... 79
List of Tables

4.1 Complete set of discrete control actions ........................................... 35

6.1 Limits of variation of the input variables in distillation column .................. 50

6.2 Values used in modeling ................................................................. 56

6.3 Comparison of the RMS and the NDEI of the process with and without the FPF 58

6.4 Adaptive gain matrix and forgetting factor values ...................................... 61

6.5 Characteristics of the applied faults ....................................................... 61

6.6 Comparison of the RMS and NDEI for the three approaches in presence of incipient fault in feed composition ......................................................... 67

6.7 Comparison of the RMS and NDEI for the three approaches in presence of abrupt fault in feed composition ......................................................... 70

6.8 Comparison of the RMS and NDEI for the three approaches in presence of incipient fault in the re-boiler temperature ........................................ 73

6.9 Comparison of the RMS and NDEI for the three approaches in presence of abrupt fault in the re-boiler temperature ........................................ 75

7.1 Consequent parameters of the fuzzy model. ............................................... 87
LIST OF TABLES

7.2 Cluster centers. ................................................. 87
List of Figures

2.1 Abrupt and incipient faults behavior

2.2 Residual

2.3 Decomposition of fault tolerant control

4.1 Principles of model based predictive control

4.2 Fuzzy model in the MBPC scheme

4.3 Branch and bound tree

4.4 Fuzzy predictive filter

5.1 Moving window to gather the data

5.2 Adaptive FTC approach in model based predictive control scheme

6.1 Boiling point diagram

6.2 Schematic representation of the distillation column. Reprinted from [39]

6.3 Inputs and output used for modeling the distillation column

6.4 Inputs and Output of the distillation column
LIST OF FIGURES

6.5 Fuzzy model performance .................................................. 57
6.6 Control of the distillation column without fuzzy predictive filter .......... 58
6.7 Control of the distillation column with fuzzy predictive filter .................. 58
6.8 Behavior of the controller for different initializations of the matrix of adaptive gain .... 60
6.9 Behavior of the controller for different values for the forgetting factor ............. 62
6.10 Process with incipient fault in the feed composition .......................... 63
6.11 Using adaptive fuzzy models to accommodate incipient faults in the feed composition ... 64
6.12 Consequents parameters behavior for the fuzzy rules .......................... 64
6.13 Using potential based approach to accommodate incipient faults in the feed composition . 65
6.14 Consequents parameters behavior for the fuzzy rules .......................... 65
6.15 Using scatter based approach to accommodate incipient faults in the feed composition ... 66
6.16 Consequents parameters behavior for the fuzzy rules .......................... 66
6.17 Changes in the parameters of the consequents of the third rule .................. 67
6.18 Process with abrupt fault in the feed composition ............................. 68
6.19 Using adaptive fuzzy models to accommodate abrupt fault in the feed composition .... 68
6.20 Using potential based approach to accommodate abrupt fault in the feed composition . 69
6.21 Using scatter based approach to accommodate abrupt fault in the feed composition .... 69
6.22 Process with incipient fault in the re-boiler temperature .......................... 71
6.23 Using adaptive fuzzy models to accommodate Incipient faults in the re-boiler temperature 71
6.24 Using potential based approach to accommodate incipient faults in the re-boiler temperature 72
6.25 Using scatter based approach to accommodate incipient faults in the re-boiler temperature 72
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.26</td>
<td>Process with abrupt fault in re-boiler temperature</td>
<td>73</td>
</tr>
<tr>
<td>6.27</td>
<td>Using adaptive fuzzy models to accommodate abrupt fault in re-boiler temperature</td>
<td>74</td>
</tr>
<tr>
<td>6.28</td>
<td>Using potential based approach to accommodate abrupt fault in re-boiler temperature</td>
<td>74</td>
</tr>
<tr>
<td>6.29</td>
<td>Using scatter based approach to accommodate abrupt fault in re-boiler temperature</td>
<td>75</td>
</tr>
</tbody>
</table>
List of Algorithms

1. Gustafson-Kessel ......................................................... 17
2. Scatter based approach ............................................... 25
3. Potential based approach ........................................... 27
4. Branch and bound .................................................... 36
LIST OF ALGORITHMS
Notation

Abbreviations

ASI Assignation of the data to an existing cluster
Hc Control horizon
ETS Evolving fuzzy modeling
FD Fault detection
FI Fault isolation
FTC Fault tolerant control
FMT Fuzzy modeling toolbox
FPF Fuzzy predictive filter
MBPC Model based predictive control
MW Moving Window
MIMO Multiple input multiple output
MISO Multiple input single output
NDEI Non dimensional error index
PI RMS Performance of improvement using RMS
Hp Prediction horizon
RLS Recursive least square
RMS Root mean square
TS Takagi-Sugeno fuzzy models
LIST OF ALGORITHMS

Symbols

\( z^i \)  center of the cluster
\( \lambda \)  forgetting factor
\( S^G \)  Global scatter
\( S^L \)  Local scatter
\( P \)  Matrix of adaptive gain
\( \mu \)  Membership degree function
\( \varphi \)  Normalized degree of fulfillment
\( R^k \)  Number of rules
\( \theta \)  Parameters of the consequence of Takagi-Sugeno fuzzy models
\( \Phi \)  Regressand matrix containing the state vectors \( x(k)^T \)
\( \Upsilon \)  Regressand matrix containing the output vectors \( y(k+1) \)
\( \sigma_j^2 (k) \)  variance
Fault tolerant control is a technique used to automatically monitor systems and in case of fault, they aim to reduce their effect or even eliminate it by using robust controllers or by reconfigure the system. There are several approaches to deal with systems in presence of faults, including fault tolerant control based on fuzzy models [35], geometric approach to nonlinear fault detection and isolation [45], fault tolerant model based predictive control using multiple takagi-Sugeno fuzzy models [27], or fault tolerant control using an admissible model matching approach [56]. The main problem of some of these approaches is that they have to know the fault in advance, i.e., they have to characterize the fault in order to be able to handle it. This implies building controllers with enough robustness, or using fault information to select a model from a model database created off-line. The main drawback of these controllers is when a non-predicted fault occurs. In the aforementioned case, the operational range is completely different from the one for which the controller was built or prepared for. Thus, the controllers are not able to exactly characterize the fault, leading to wrong results in control strategy. Therefore, it is clear that arises the necessity of controllers that can continuously adapt themselves to different operational ranges, caused by faults or disturbances in the systems. In order to achieve this aim, this dissertation proposes a predictive fault tolerant control approach using evolving fuzzy modeling. The proposed approach has the model-based predictive control (MBPC)
1.1 Objective and outline of this work

and the evolving fuzzy modeling (ETS) [5, 6], or the adaptive fuzzy models [9, 54], as framework. Evolving fuzzy modeling are models whose rule-base and parameters continuously evolve by adding new rules with more summarization and updating the existing rules and parameters. Differently, adaptive fuzzy models, keep the rule base and the structure fixed changing only the parameters of the consequents. By using these approaches, the models in MBPC are adapted at each time that the process operational range change due to faults or disturbances. That adaption allows the model to "know" the actual operational range, and consequently good predictions of the process behavior are achieved. Differently from the approach in [35] in which, the fault is detected, isolated (fault characterization) and then accommodated. The proposed approaches does not have the stage of isolation. The fault is detected when a residual passes a certain threshold. This residual threshold is defined, by knowing the system behavior. To set this value, one analyzes the free fault system output, and chooses a value bigger than this value, and use it as the residual threshold. When the fault is detected, the data with fault are gathered with a moving window, and when the moving window is full, the proposed fault tolerant control approaches are activated. The advantage of using one of these techniques is the fact that both have fuzzy logic as background, allowing them to deal with uncertain (data with fault and disturbances) information, resulting in a more effective control. The proposed approaches are applied to a distillation column, to accommodate faults that may occur in this equipment.

1.1 Objective and outline of this work

The purposes of this thesis is to develop a fault tolerant control approach using evolving fuzzy modeling.

This approach is then used to control a distillation column. The outline of the thesis is as follows:

Chapter 2: Fault tolerant control is the issue of this chapter, an overview in this topic is made.
Chapter 3: In this chapter one presents evolving fuzzy modeling, the Takagi-Sugeno fuzzy models is the topic of the first part of the chapter, and in the second evolving fuzzy modeling is presented.

Chapter 4: Model based predictive control is the issue of the chapter, the classical approach is described and the advantage of using fuzzy models, and some optimization techniques in these controllers are also presented.

Chapter 5: The proposed predictive fault tolerant control using adaptive fuzzy modeling is the issue of this chapter.

Chapter 6: The application of the predictive fault tolerant control using adaptive fuzzy modeling to the distillation column is made, and the results are presented.

Chapter 7: Conclusions and recomendations for future work are the issue of this chapter.
1.1 Objective and outline of this work
Chapter 2

Fault Tolerant Control

2.1 Introduction

There are several processes that although one or more faults may occur within them, they must not stop working because of their impact in political, economical and social domains. Fault tolerant control (FTC), fault detection (FD) and fault isolation (FI) are research areas which aim to solve this problem. These research areas, which have been developing since the 80’s [42], are interested in studying methodologies for identifying and exactly characterizing possible faults that may occur in processes. When fault tolerant systems are used in control systems, it is possible to maintain the system working by accommodating the faults effects.

Figure 2.1 depicts the two main types of faults; abrupt faults and incipient faults. An abrupt fault is characterized by fast variations of the variables, giving rise to sudden faults. Normally they are modelled as a step. Incipient faults are characterized by slow variations of the variables, as depicted on the Fig. 2.2.

The fault detection may be carried out by different ways such as hardware redundancy [31], which consists of comparing the outputs of identical hardware and performing consistence cross checks. This approach is not always feasible in practice due to the cost of a solution of this kind. Analytical redundancy [22] is
2.1 Introduction

![Figure 2.1: Abrupt and incipient faults behavior](image)

![Figure 2.2: Residual](image)

an approach in which the FD can be accomplished using analytical and functional information about the system, using a mathematical model. There are two classes of model-based approaches. In the first class, quantitative models such as transfer functions, differential equations and state space models are used. These methods make use of theoretical tools such as parameter estimation techniques, state estimation techniques and parity space concepts. The gold rule for this approach is an a priori knowledge about the relationships between faults and changes in model parameters. The second approach is based on the use of artificial intelligence methods (fuzzy logic and neural networks), using qualitative reasoning and modeling. Qualitative models are used to predict the behavior of the system in nominal conditions and in different fault condition, the detection and isolation of fault is performed by the residual analysis.

Chapter summarization

In Section 2.2, passive and active fault tolerant control are both presented. In Section 2.3, an overview of fault detection and isolation is made. Finally model based predictive control in fault tolerant control is the
2.2 Passive and active fault tolerant control systems

Fault tolerant control can be divided in active and passive methods [41]. Figure 2.3 presents a classification of FTC systems. A more detailed explanation of the methods in Fig. 2.3 is presented in the following.

2.2.1 Passive fault tolerant control system

![Figure 2.3: Decomposition of fault tolerant control](image)

Passive fault tolerant control is a method in which robust control is used [34]. These controllers are made to be fault insensitive and the process with fault remains working with the same controller and structure without losing performance. In general, the FTC architecture, and fault information and location is usually required before the controller react to the fault. In passive FTC systems, the controller is made robust to faults by assuming a restrictive kind of faults and the way they affect the controlled process. A priori knowledge of the faults and their influence in the system are required for this types of controller.
2.3 Fault detection and Isolation

2.2.2 Active fault tolerant control system

Differently from passive FTC, active systems are made to be fault sensitive. Therefore, when a fault occurs the fault detection system is activated. Active FTC may be applied through reconfigurable control (reconfigurable control is an active FTC approach in which the parameters of the controller are adjusted to accommodate faults in the process)[10]. The reconfiguration is not applied only in the parameters, but the controller structure may also change. An active FTC system requires either a priori knowledge of expected faults or a mechanism for detecting and isolating anticipated faults. When the faults are detected, the decision concerning the location and nature of the faults are then used to reconfigure the control function. Active fault tolerant control is divided into two main fields, the projection based methods and the adaptive controllers as depicted in the Fig. 2.3. In projection based methods, one control law is selected in a pre-defined data base containing possibles control laws according to pre-defined faults and in adaptive controllers the model are on-line reconfigured. In the proposed FTC approaches the parameters and structure of the controller are reconfigured on-line.

2.3 Fault detection and Isolation

Since the 70’s, huge effort has been invested in fault detection and isolation (FDI) [43, 63]. In FDI, several approaches have been developed, including qualitative and quantitative model-based approaches[59, 61], and knowledge based approaches [24].

The major task of FDI is to perform real time diagnostic of the plant, reading each time the process variables and the model outputs and compute the residuals. In system failure-free (systems without fault) the residual should be close to zero. Many reconfigurable controllers use real time estimates of system parameters
provided by parameter estimation based FDI methods\cite{40, 61}. FDI based upon state estimation is said
to be less accurate in providing information to the controller than the parameter estimation approach for
on-line reconfiguration. However, in many applications it is extremelly difficult to get reasonably accurate
parameters on-line. In order to overcome the disadvantages of using parameters estimation, different
controllers reconfiguration have been developed. The FDI is expected to generate the residual, providing
detailed information about faulty conditions and take the decision if a fault has occurred or not. In the
proposed approaches the fault isolation is unnecessary once that the fault do not need to be characterized.

2.4 Model based predictive control in fault tolerant control

Model based predictive control (MBPC), has the advantage of using an objective function in which the
user can specify some requirements such as energy minimization or fast control. Such characteristic can
be used in FTC to specify some requirements.

In some FTC approaches in which the models with faults are used, fault can be treated as a modification
in MBPC constraints. Fault accommodation is implemented through the determination of the appropiated
control actions which reduce or eliminate the fault effects.

2.5 Evolving fuzzy modeling in fault tolerant control

This thesis proposes predictive fault tolerant control using evolving fuzzy modeling, the advantage of the
this approach in FTC is that the parameters and the structure of models used in the MBPC are continuously
evolving. Allowing the these controllers to be updated with information contained in new spaces created
by the faults. Evolving fuzzy modeling is the issue of the next chapter.
2.5 Evolving fuzzy modeling in fault tolerant control
Evolving Fuzzy Modeling

3.1 Introduction

The backbone for any kind of controller based on models are the models themselves. Developing good models is essential to achieve good performances. There are several tools for the modeling task, whose selection depends upon the data (linear or non-linear) that are being used. Fuzzy models are known by their ability to deal with uncertain information in data for dynamic modeling of systems.

When there is no information about the system, the rules and membership functions of fuzzy system can be obtained by the process input and output data [9]. The rules and membership functions are obtained by the following methods: fuzzy clustering [9], neuronal learning [29] or orthogonal least square [50]. Comparing with other techniques of nonlinear modeling such as neural networks, fuzzy modeling has the advantage of representing the systems with more clarity allowing an easier understanding of the linguistic interpretation by their rules. The if-then rule in fuzzy modeling has the following structure:

\[
\text{if antecedent then consequent} \tag{3.1}
\]

The fuzzy models can be considered as logical models, which use if-then rules to establish a quantitative relation between the model variables. Depending upon the particular structure of the consequent
3.1 Introduction

proposition, the fuzzy models can be distinguished in:

- Linguistic fuzzy model or Mamdani models [32], where both antecedent and consequent are fuzzy propositions.

- Takagi-Sugeno (TS) fuzzy models [9] where the consequent is a crisp function of the antecedent variables rather than a fuzzy proposition.

Mamdani fuzzy models demand a wide knowledge of the system, and when the system is nonlinear, as in most industrial processes, it is very difficult to use this type of models. Takagi-Sugeno fuzzy models are widely used to represent complex nonlinear systems. These models are also relatively easy to identify and their structure can be readily analysed.

The identification of both model structure and parameters can be handled by several approaches [9, 46, 50, 54]. The aforesaid approaches can be classified as data-driven rule/knowledge extraction.

The real problem arises with the necessity of on-line implementation, in which some practical applications called self-learning or adaptive, have been used. However, these approaches are rather self-adjusting and self-tuning normally supposing that the structure of the model is fixed. Recently, algorithms of on-line learning with evolving of the structure have been reported. Evolving fuzzy modeling [4, 5, 6] are models where rule base evolves continuously by adding new rules with more information, updating the existent, or removing rules from the rule data base.

Chapter summarization

This chapter, is divided in to two main parts. First, Section 3.2 presents a small introduction of Takagi-Sugeno fuzzy models. An overview in off-line identification is made in the Section 3.2.1, and in the Section
3.2.2 on-line adaptation of the fuzzy models is the presented. In Section 3.3, evolving fuzzy modeling are presented, where both the scatter and the potential based approaches are described.

### 3.2 Takagi-Sugeno fuzzy models

Takagi-Sugeno (TS) fuzzy models are models in which the consequents of the rules are mathematical functions

\[ R : \text{if} \ x \ \text{is} \ A \ \text{then} \ y = f^k(x) \]  \hspace{1cm} (3.2)

where \( R \) is the \( k \) rule, \( x \) are the antecedents, \( y \) are the consequents and \( A \) is the multidimensional antecedent fuzzy set of the \( k \)th rule. The multiple-input and multiple-output (MIMO) fuzzy model represented in (3.2), can be decomposed into a collection of multiple-input and single-output (MISO) fuzzy models without loss of generality [54]. The most simple and widely used function is the affine linear form:

\[ R : \text{if} \ x \ \text{is} \ A \ \text{then} \ y^k = (a)^T x + b \]  \hspace{1cm} (3.3)

where \( a \) is a parameter vector and \( b \) is a scalar offset, the consequent of the affine TS model are hyperplanes in the product space of the inputs and outputs.

Before inferring the output the degree of fulfillment of the antecedents \( \varphi_i(x) \) is computed. For rules with multivariate antecedents fuzzy sets, the degree of fulfillment is equal to the membership degree of the given input \( x \), i.e., \( \varphi_i = \mu_i(x) \). In Takagi-Sugeno fuzzy models, the inference is reduced in the following equation:

\[ y = \frac{\sum_{i=1}^{K} \varphi_i(x) y_i}{\sum_{i=1}^{K} \varphi_i(x)} \]  \hspace{1cm} (3.4)
3.2 Takagi-Sugeno fuzzy models

When considering a fuzzy modeling approach, one has to choose the type of fuzzy model a priori, which depends on the particular application. TS fuzzy models are more suitable for identification for non-linear systems from the data, while linguistic fuzzy models give a more quantitative description of the system and can also be used when dealing with knowledge of the process. For that reason, Takagi-Sugeno fuzzy models are going to be used.

3.2.1 Off-line identification

In this type of identification, it is assumed that the structure of the system is known in advance. Modeling of dynamic systems by fuzzy clustering generally entails the following steps [54]:

1. Determine the model structure suitable to the problem by identifying the relevant system variables. These may be data regarding the system state, inputs, output, errors or other.

2. Collect the data and construct a set of vectors \( \{x, y\} \), where \( x \) is the input and \( y \) is the output.

3. Select the clustering algorithm and determine the values of the parameters relevant to the clustering method used.

4. Select the number of required clusters.

5. Cluster the data with the selected clustering algorithm.

6. Obtain membership functions from clusters by projection or product space.

7. Determine a fuzzy rule from each cluster by using the obtained membership functions.

8. Validate the model.
Evolving Fuzzy Modeling

Assuming that the inputs and outputs are known, the nonlinear identification steps presented previously can be compressed in two steps [54]:

1. Structure identification

2. Parameter estimation

**Structure identification**

Structure identification allows to transform the dynamic identification problem into a static nonlinear regression. For the sake of simplicity, let each MISO system be identified separately. The total MIMO system can be derived as a collection of MISO systems [54]. A MISO system can be described by

\[
\hat{y}(k+1) = f(x(k)). \tag{3.5}
\]

Let \( N \) denote the number of data samples, selected from the input and output data sequences. This number must be much larger than the number of states in the system, i.e., \( N \gg n \). Let \( N_d \) be the actual number of points used in identification and \( k_h \) is the highest order of the inputs and outputs, then \( N_d = N - k_h \). Let \( \Phi \) denote the regressand matrix in \( \mathbb{R}^{N_d \times n} \) with the state vectors \( x(k)^T \) in its rows, and \( \Upsilon \) denote the vector in \( \mathbb{R}^{N_d} \) containing the regressand \( y(k+1) \), with \( k = k_h, \ldots, N-1 \)

\[
\Phi = \begin{bmatrix}
  x(k_h)^T \\
  \vdots \\
  x(N-1)^T
\end{bmatrix}, \quad \Upsilon = \begin{bmatrix}
  y(k_h + 1)^T \\
  \vdots \\
  y(N)^T
\end{bmatrix} \tag{3.6}
\]

**Parameter estimation**

In this step, the number of the rules \( K \), the antecedent fuzzy set \( A^k \), and the consequent parameters \( a^k, b^k \) from (3.3), for all \( k \) rules are determined. Fuzzy clustering in the cartesian product space \( \Phi \times \Upsilon \)
is applied to partition the data into subsets, which can be approximated by linear models [9]. In system
identification, clustering finds relationships among the system variables. The data set \( Z \) to be clustered is
formed by appending \( \Upsilon \) to \( \Phi \)

\[
\mathbf{Z} = [\Upsilon, \Phi]^T
\]  (3.7)

The columns of \( \mathbf{Z} \) are denoted by \( z_l, l = 1, \ldots, N_d \). Let \( \mathbf{U} = [\mu_{kl}]^{K \times N_d} (\mu_{kl} \in [0, 1]) \) denote a
fuzzy partition matrix of \( \mathbf{Z} \). Let \( \mathbf{V} \) be the vector of cluster prototypes (centers) to be determined by
\( \mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K] \), and let \( \mathbf{F} \) be a set of cluster covariance matrices \( \mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_K] \), where
\( \mathbf{F}_k \) are positive defined matrices in \( \mathbb{R}^{(n+1) \times (n_d+1)} \).

The Gustafson-Kessel algorithm (GK) [9] Algorithm 1 searches for an optimal fuzzy partition \( \mathbf{U} \), the proto-
type matrix of cluster means \( \mathbf{V} \), and a set of cluster covariance matrices \( \mathbf{F} \). In other words,

\[
(Z, K) \xrightarrow{\text{clustering}} (U, V, F)
\]  (3.8)

The optimization minimizes the following objective function,

\[
J(Z, U, V) = \sum_{k=1}^{K} \sum_{l=1}^{N_d} (\mu_{kl})^q d_{kl}^2
\]  (3.9)

where \( q \) is the weighting parameter. The function \( d_{kl} \) is the distance of the data point \( z_l \) to the cluster
prototype \( \mathbf{V}_k \). In the GK clustering algorithm, the distance is computed from the covariance matrices
according to

\[
d_{kl}^2 = (z_l - \mathbf{V}_k)^T \frac{F_k^{-1}}{|F_k|/(n+1)} (z_l - \mathbf{V}_k)
\]  (3.10)

where \( |F_k| \) is the determinant of the covariance matrix \( F_k \).

Parameter estimation entails the following subparts:

1. **Number of clusters**

The number of clusters determines the number of rules in the obtained fuzzy model. This parameter
Algorithm 1 Gustafson-Kessel
Given the data set \( Z_k \), choose the number of fuzzy rules (clusters) \( 1 < k \ll N \), the weighting exponent \( q > 1 \) and the termination tolerance \( \epsilon > 0 \). Initialize the partition matrix randomly.

Repeat for \( l = 1, 2, \ldots \)

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

\[
\mathbf{v}_k^{(l)} = \sum_{i=1}^{N_d} \left( \mu_k^{(l-1)} \right)^q \mathbf{z}_i \sum_{i=1}^{N_d} \left( \mu_k^{(l-1)} \right)^q, \quad 1 \ll k \ll K.
\]

**Step 2:** Compute covariance matrices (Prototypes):

\[
\mathbf{F}_k = \sum_{i=1}^{N_d} \left( \mu_k^{(l-1)} \right)^q \left( \mathbf{z}_i - \mathbf{v}_k^{(l)} \right) \left( \mathbf{z}_i - \mathbf{v}_k^{(l)} \right)^T \sum_{i=1}^{N_d} \left( \mu_k^{(l-1)} \right)^q, \quad 1 \ll k \ll K.
\]

**Step 3:** Compute distances:

\[
d_{kl}^2 = \left( \mathbf{z}_l - \mathbf{v}_k^{(l)} \right)^T |\mathbf{F}_k|^{\frac{1}{2}} F_k^{-1} \left( \mathbf{z}_l - \mathbf{v}_k^{(l)} \right).
\]

**Step 4:** Update partition matrix:

if \( d_{kl} > 0 \) for \( 1 \leq k \leq K, 1 \leq l \leq N_d, \)

\[
\mu_{kl}^{(l)} = \frac{1}{\sum_{j=1}^{K} \left( \frac{d_{kj}}{d_{kl}} \right)^{2(q-1)}},
\]

otherwise

\[
\mu_{kl}^{(l)} = 0 \text{ if } d_{kl} > 0 \text{ and } \mu_{kl}^{(l)} \in [0, 1]
\]

with \( \sum_{k=1}^{K} \mu_{kl}^{(l)} = 1 \)

until \( \|U^{(l)} - U^{(l-1)}\| < \epsilon \).
3.2 Takagi-Sugeno fuzzy models

is an important parameter that influences the accuracy and transparency of the fuzzy models [9].

Two main strategies to determine the appropriate number of clusters in data are:

- Cluster the data for different values of \( K \) and then use a mathematical goodness of the obtained partitions. Gath and Geva proposed different validity measures related to distance adaptive clustering techniques [28].
- Start with a sufficiently large number of clusters and reduce that number successively by combining the compatible clusters. This cluster merging approach is called compatible cluster merging [54].

2. Antecedent membership functions

Each cluster represents one TS fuzzy rule. The multidimensional membership functions \( A^k \) are given analytically by computing the distance of \( x(k) \) from the projection of the cluster center \( v_k \) onto \( X \), and then computing the membership degree in an inverse proportion to the distance. Denote with \( F^x_k = [f_{jl}] 1 \leq j, l \leq n \), the submatrix of \( F^x_k \). This matrix describes the form of the cluster in the antecedent space \( X \). Let \( F^x_k = [F_{1k}, \ldots, F_{nk}]^T \) denote the projection of the cluster center onto the antecedent space \( X \), the inner-product distance norm is given by:

\[
d_{kl} = (x(k) - v^x_k)^T [((F^x_k)^T)^{-1/n}(F^x_k)^{-1}] (x(k) - v^x_k)
\]

which is converted into the membership function degree by

\[
\mu_{A^k} (x(k)) = \frac{1}{\sum_{j=1}^{K} \left( \frac{d_{kj}}{d_{kl}} \right)^{2(q-1)}}
\]

where \( q \) is the fuzziness parameter of GK algorithm.
3. Consequent parameters

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

\[
\Phi' = \left[(\Gamma^1 \Phi_e), (\Gamma^2 \Phi_e), \ldots, (\Gamma^K \Phi_e)\right]
\]  

(3.13)

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

\[
\theta' = \left[(\theta^1)^T, (\theta^2)^T, \ldots, (\theta^K)^T\right]^T
\]

(3.14)

The resulting least square problem, \(\Upsilon = \Phi' \theta' + \epsilon\) has the solution

\[
\theta' = \left[(\Phi')^T \Phi'\right]^{-1} (\Phi')^T \Upsilon
\]

(3.15)

The optimal parameters \(a^k\) and \(b^k\) are given by

\[
a^k = [\theta'_{s+1}, \theta'_{s+2}, \ldots, \theta'_{s+n}]^T
\]

\[
b^k = [\theta'_{s+n+1}]
\]

(3.16)

where \(s = (k - 1)(n + 1)\).

With the determination of the parameters \(a^k\) and \(b^k\), the fuzzy model identification procedure is completed.

3.2.2 On-line adaption of the fuzzy models

Changes in the operating conditions due to disturbances or faults, often occur in many industrial processes. To assure the desired product quality, the process control system have to deal with these changes.
3.2 Takagi-Sugeno fuzzy models

There are several adaptive control structures in fuzzy control literature among them: self-learning fuzzy control based on reinforcement learning [11], classical self-organizing linguistic controller [46] and neuro-fuzzy controller with temporal backpropagation learning [30]. The common feature of these approaches is that the controller is adapted directly without identifying the process model. Differently from those approaches, an approach consisting of adapting the fuzzy model, using the exact inversion of the model to derive the control input is proposed in [53].

In this thesis, the adaptation is performed directly in the consequent parameters of the fuzzy models. It is assumed that the antecedent partition is derived off-line and remains valid. Since the antecedent parameters are determined and fixed as in 3.11 and 3.12, the model is linear in consequents and recursive least square can be used to estimate the consequents.

The rule consequents are adapted by

$$\theta_j(k) = \theta_j(k-1) + \gamma_j(k) (y(k) - x^T(k) \theta_j(k-1))$$  \hspace{1cm} (3.17)

where $\theta_j(k)$ is the vector of consequent parameters and $\gamma$ is an intermediate variable without a specific physical meaning. $\theta$ and $\lambda$ are initialized as following $\theta_{init} = FM.th$ and $\lambda = \lambda_{init}$

$$\gamma_j(k) = \frac{P_j(k-1) x(k)}{x^T(k) P_j(k-1) x(k) + \lambda_j/\mu_A^j(x)}$$  \hspace{1cm} (3.18)

where $\lambda$ is the forgetting factor that implements forgetting of the old measurements and $P_j$ is a matrix of adaptation gain initialized as $P_{init} = FM.p1$.

$$P_j(k) = \frac{1}{\lambda} [I - \gamma_j x^T(k)] P_j(k-1)$$  \hspace{1cm} (3.19)

$FM.th$ is the value of the consequent parameters obtained offline, $\lambda_{init}$ and $P_{init}$ are values solved by a trial and error process, explanation is made in the chapter 6.
The presented Takagi-Sugeno fuzzy models are constant in structure, next Section the evolving fuzzy modeling are presented, in the fuzzy models the structure evolves continuously.

3.3 Evolving fuzzy modeling

The evolving fuzzy modeling (ETS) is a TS fuzzy model whose rule-base and parameters continually evolve by adding new rules with more summarization power and by modifying the existing rules and parameters [4, 5, 6]. The algorithm continuously evaluates the scatter or the potential contained in the new data (both the scatter and the potential are seen as measures of average distances from a data sample to an other data sample) and dynamically updates the number of rules and their antecedent parameters, combining that process with a recursive update of the consequence parameters. ETS ensures that high control performance can be achieved even with time variant process behaviour. Basically, on-line learning of ETS entails the following stages:

1. Initialization of the rule-base structure (antecedent part of the rule).

2. Reading the next data sample at the next time step.

3. Recursive calculation of the scatter (average distance from a data sample to other data sample) of the new data.

4. Recursively update the scatter at the focal point (rule center) of the existing clusters.

5. Modification or rule-base up-grade based on the scatter of the new data in comparison to scatter of the existing rule.

6. Recursive calculation of the consequent parameters.
7. Prediction of the model output of the next time step.

For on-line clustering, the data normalization is required, this normalization is made by solving recursively the mean $\bar{z}_j(k)$ and the variance $\sigma_j^2(k)$ for each element $z_j$ of the input/output vector $z(k) = [x(k), y(k)]$, where $k$ is the sampling time.

\[ \bar{z}_j(k) = \frac{k-1}{k}z_j(k-1) + \frac{1}{k}z_j(k) \quad (3.20) \]

\[ \sigma_j^2(k) = \frac{k-1}{k}\sigma_j^2(k-1) + \frac{1}{k-1}(z_j(k) - \bar{z}_j(k))^2 \quad (3.21) \]

The normalized is given by

\[ z_{nt}(k) = \frac{(z_j(k) - \bar{z}_j(k))}{\sigma_j(k)} \quad (3.22) \]

With normalized data that establish the focal point of the first cluster ($i = 1$), the scatter of the first focal point is assumed to be $S_1 = 0$

\[ R = 1; x^{1*} = x(k); S_1(z^{1*}) = 0; \quad (3.23) \]

where $z^{1*}$ is the center of the first cluster; $x^{1*}$ is the focal point (projection of $z^{1*}$ in the $x$ axis) $R$ is the number of initial rules.

The global scatter of the new data is solved by (3.24), which is the distance from a data sample to all other data samples.

\[ S_k^G(z(k)) = \frac{1}{(N)(n+m)} \sum_{l=1}^{N} \sum_{j=1}^{n+m} (z_j(l) - z_j(k))^2. \quad (3.24) \]

The values of $S_k^G(z(k))$ have the range $[0, 1]$, with 0 meaning that all of the data sample coincide (which is extremely improbable) and 1 meaning that all of the data are on the vertices of the hypercube formed.
Evolving Fuzzy Modeling

as a result of normalizing the data.

Recursively, equation (3.24) is given by:

\[
S_k(z(k)) = \frac{1}{(k-1)(n+m)} \left( (k-1) \sum_{j=1}^{n+m} z_j^2(k) - 2 \sum_{j=1}^{n+m} z(k) b_j(k) + h(k) \right)
\]  (3.25)

where \(b_j(k) = \sum_{j=1}^{k-1} z_j(l)\) and \(h(k) = \sum_{l=1}^{k-1} \sum_{j=1}^{n+m} z_j^2(l)\).

The parameters \(b_j(k)\) and \(h(k)\) are recursively updated by

\[
b_j(k) = b_j(k-1) + z_j(k-1) ; h(k) = h(k-1) + \sum_{j=1}^{n+m} z_j^2(k-1)
\]  (3.26)

The new data influence the scatter at the centers of the clusters \((z_i^*, i = 1, 2, \ldots, R)\). The reason is that by definition, the global scatter depends on the distance to all points, including the new ones.

After the scatter of the existing clusters is updated, a comparison between the scatter of two consecutive steps, \(k - 1\) and \(k\) is made:

\[
S_{k-1}(z^*) = \frac{1}{(k-2)(n+m)} \sum_{l=1}^{k-2} \sum_{j=1}^{n+m} (z_j(l) - z_j(k-1))^2
\]  (3.27)

\[
S_k(z^*) = \frac{1}{(k-1)(n+m)} \sum_{l=1}^{k-1} \sum_{j=1}^{n+m} (z_j(l) - z_j(k))^2
\]  (3.28)

simplifying and comparing (3.27) and (3.28) the recursive equation is then obtained:

\[
S_k(z^*) = \frac{k-2}{k-1} S_{k-1}(z^*) + \sum_{j=1}^{n+m} (z_j(k) - z_j(k-1))^2
\]  (3.29)

The scatter of the new data sample is compared to the updated scatter of the centers of the existing clusters. The decision whether to modify or upgrade is taken based on the following rule:

\[
\text{IF} \quad \left[ S_k(z(k)) < \min_{i=1}^{R} S_k(z_i^*) \right] \text{ OR } \left[ S_k(z(k)) > \max_{i=1}^{R} S_k(z_i^*) \right] \quad (3.30)
\]

and \(z(k)\) is close to an existing rule center given by

\[
\delta_{\min}(k) < 0.5r
\]  (3.31)
3.3 Evolving fuzzy modeling

\[
\delta_{\min}(k) = \min_{i=1}^{R} \| x(k) - x^i* \|^2 \tag{3.32}
\]

where \( \delta_{\min} \) denotes the distance from the new data point to the closest of the existing rule centers.

Then the new data point \( z(k) \) replaces this center \( z^i* \)

\[
x^i* = x(k); \quad S_k(z^i*) = S_k(z(k)); \quad N^i(k) = N^i(k) + 1; \quad ASI^i(k) = ASI^i(k) + k \tag{3.33}
\]

ELSE The new data is added to the rule base, new focal point and new rule cluster is formed

\[
R = R + 1; \quad N^l(k) = 1; \quad x^{R*} = x(k); \quad S_k(z^{R*}) = S_k(z(k)) \tag{3.34}
\]

ELSEIF equation (3.31) is not satisfied

The data sample is assigned to the nearest existing cluster/focal point of the rule

\[
ASI^l(k) = ASI^l(k) + k; \quad N^l(k) = N^l(k) + 1; \quad i = \arg\min_{i=1}^{R} \| x(k) - x^i* \|^2 \tag{3.35}
\]

where \( ASI \) means assignation of the data to an existing cluster. It should be noted that the distance \( \delta_{\min} \) is calculated over the input only, disallowing rules with similar antecedents to co-exist. The rule base evolve incrementally, when a new rule is created, it is judged by the existing rules, not the future ones. As the data pattern can be less and less relevant in the future, the population of each cluster is monitored if it amounts to less than 1% of the total data samples at that moment the cluster/rule is ignored from the rule base by setting its firing level to 0.

\[
\text{IF } \frac{N(l)}{N} < 0.01 \text{ THEN } (\phi_i = 0) \tag{3.36}
\]

The rule-base gradually evolves. Therefore, the normalized firing strengths of the rules change, which affects all the data.

Once the antecedents are determined and fixed, the model is linear in parameters and, recursive least square (RLS) by 3.17, 3.18 and 3.19.

The presented steps are resumed in Algorithm 2. The other approach of the evolving fuzzy modeling is
Algorithm 2 Scatter based approach

**Step 1:** Read the existing FM:

**Step 2:** Initializations:

$$R = 1; x^{i*} = x (k); S_1 (z^{i*}) = 0;$$ (3.37)

where $z^{i*}$ is the center of the first cluster; $x^{i*}$ is the focal point (projection of $z^{i*}$ in the x axis).

**Step 3:** Calculate the scatter of new data:

$$S_k (z (k)) = \frac{1}{(k-1) (n+m)} \left[ (k-1) \sum_{j=1}^{n+m} z_j^2 (k) - 2 \sum_{j=1}^{n+m} z (k) b_j (k) + h (k) \right]$$ (3.38)

where $b_j (k) = \sum_{j=1}^{k-1} z_j (l)$ and $h (k) = \sum_{l=1}^{k-1} \sum_{j=1}^{n+m} z_j^2 (l)$.

The parameters $b_j (k)$ and $h (k)$ are recursively updated by

$$b_j (k) = b_j (k-1) + z_j (k-1); h (k) = h (k-1) + \sum_{j=1}^{n+m} z_j^2 (k-1)$$ (3.39)

**Step 4:** Update the scatter of the existing clusters:

$$S_k (z^{i*}) = \frac{k-2}{k-1} S_{k-1} (z^{i*}) + \sum_{j=1}^{n+m} (z_j (k) - z_j (k-1))^2$$ (3.40)

**Step 5:** Compare the scatter of the new data with the scatter of the existing centers:

**IF**

$$S_k (z (k)) < \min_{i=1}^{R} S_k (z^{i*}) \ldots OR \ldots S_k (z (k)) > \max_{i=1}^{R} S_k (z^{i*})$$ (3.41)

**AND** $z (k)$ is close to an existing rule center given by

$$\delta_{\text{min}} (k) < 0.5 r$$ (3.42)

**THEN** the new data point $z (k)$ replaces this center

$$x^{i*} = x (k); S_k (z^{i*}) = S_k (z (k)), N^l (k) = N^l (k) + 1 \text{ and } ASI^l (k) = ASI^l (k) + k$$ (3.43)

**ELSEIF** equation 3.41 is satisfied and equation 3.42 is not.

The data sample is assigned to the nearest existing cluster/focal point of the rule

$$ASI^l (k) = ASI^l (k) + k; N^l (k) = N^l (k) + 1; i = \arg \min_{i=1}^{R} \| x (k) - x^{i*} \|^2$$ (3.45)

**Step 6:** Calculate the consequents of the parameters by 3.17, 3.18 and 3.19 and update FM

END IF
3.3 Evolving fuzzy modeling

the potential based approach, it is applied, according to the Algorithm 3. According to [5] the threshold presented in this algorithm are calculated by 3.46 or 3.47. In which the upper threshold $\chi$ ensures that data samples with potential above half of the best are accepted only and the lower one define a gray zone [18] where the data are checked on the proximity to the existing centers.

\[
\chi = 0.15 \cdot \max_{i=1}^{R} p_i^*; \quad \bar{\chi} = 0.5 \cdot \max_{i=1}^{R} p_i^* \tag{3.46}
\]

\[
\chi = \bar{\chi} = \frac{1}{R} \sum_{i=1}^{R} p_i^* \tag{3.47}
\]

The presented evolving fuzzy modeling are going to be used in the model based predictive control (MBPC). MBPC is the issue of next chapter.
Algorithm 3 Potential based approach

IF buffer is full

**Step 1:** Read the existing FM:

**Step 2:** Initializations:

\[
R = FM.c; \; x_i^1 = FM.V; \; P_i^* = 1; \; i = 1, 2, \ldots, R; \; \theta_{init} = FM.th; \; P_{init} = P_{init}
\] (3.48)

**Step 3:** Calculate the potential of the new data:

\[
P^k = \frac{1}{1 + (k - 1) a - 2b + c_k}
\] (3.49)

where

\[
a = \sum_{j=1}^{n+1} (z_{jk})^2, \; v_j^i = \sum_{i=1}^{k-1} z_{ij}^i, \; c_k = \sum_{i=1}^{k-1} \sum_{j=1}^{n+1} (z_{ij}^i)^2 \quad \text{and} \quad b = \sum_{j=1}^{n+1} z_{jk} v_j^k
\]

**Step 4:** Update the potential of the existing clusters:

\[
[p_i^*]^k = \frac{[p_i^*]^{k-1}}{1 + [p_i^*]^{k-1} \sum_{j=1}^{n+1} (d_{ki})^2}
\] (3.50)

where \([p_i^*]^k\) denotes the potential of the \(i\)th center of the \(k\)th data.

**Step 5:** Compare the potential of the new data with the potential of the existing centers:

IF \(P^k\) is higher than certain threshold \(\chi\) AND is true the equation 3.51

\[
\frac{\arg \min_{i=1}^{R} \|z^k - z_i^*\|}{R} + \frac{p(z^k)}{\max_{i=1}^{R} p(z_i^*)} < 1
\] (3.51)

THEN the new data \(z^k\) replaces the old center

\[
z_j^* = z^k, \; p_j^* = p^k, \; \arg \min_{i=1}^{R} \|z^k - z_i^*\|
\] (3.52)

and GO TO : Step 6

ELSE IF \(P^k\) is higher than certain threshold \(\chi\) AND is false the equation 3.51

THEN it is accepted as a new rule’s center

\[
R = R + 1; \; z_R^* = z^k, \; p_R^* = p^k
\] (3.53)

**Step 6:** Calculate the consequents of the parameters by 3.17, 3.18 and 3.19 and update FM

END IF
3.3 Evolving fuzzy modeling
Model Based Predictive Control

4.1 Introduction

Model based predictive control (MBPC) can be applied to linear and nonlinear models, and is a control technique with multiple applications in process industry [17]. The use of this kind of controllers that take into account the nonlinearities of the process, implies an improvement in the performance of the process by reducing the impact of disturbances.

The performance of the MBPC depends on the predictive accuracy of the model. For the aforementioned, one may conclude that the major part of MBPC design and cost is related to modeling the process.

MPBC uses the process model to estimate the inputs for the process by minimizing an objective function. When the process is linear without constraints, the objective function is quadratic, and the optimization problem has an analytic solution. When the process is nonlinear with constraints, the optimization problem becomes a convex optimization problem, solved by methods such as:

- Gradient search methods
- Quadratic programming [33]
4.2 Classical model based predictive control

- Gradient free search methods
  - Discrete search (dynamic programming (DP)) [2]
  - Tree search (branch and bound) [15]
  - Stochastic (Simulated annealing [62], Genetic algorithms [23])

In MBPC, the future outputs can be estimated by using a model of the process to be controlled.

Chapter summarization

In this chapter, an overview of model based predictive control is done, starting in Section 4.2 with classical model based predictive control. In Section 4.3, the advantages of using fuzzy models in predictive control are presented and Section 4.4 describes optimization techniques in MBPC, such as branch and bound. The branch and bound optimization technique introduces a trade-off between accuracy and computational complexity. Fuzzy predictive filters to avoid chattering are in Section 4.4.

4.2 Classical model based predictive control

The strategy in predictive control is based on a receding horizon optimization, calculated on-line at each sampling time [21]. As depicted in the Fig. 4.1. This control strategy entails the following steps:

1. The future output at a determined horizon, called the prediction horizon, is predicted at each instant \( k \) using the model of the process. These predictions of the output \( \hat{y}(k) \) depends on the values known until the instant \( k \) (known inputs and outputs) and the control signals \( u(k) \), to be calculated and sent to the system.
2. The sequence of future control signals is calculated by minimizing an objective function to keep the process as close as possible to the reference trajectory. Usually, the objective function takes the form of a quadratic function of the error between the predicted output and the trajectory of future reference. In most cases it also includes the control effort within the objective function.

3. The control signal $u(k)$ is sent to the process while the rest of the signals calculated are not considered, because in the next sampling instant $y(k+1)$ is already known and the above steps are repeated with this new value.

**Predictive control algorithm**

- Sample the output of the plant.

- Use the model of the plant to predict its future behavior over a prediction horizon $H_p$, the control action is applied along a control horizon $H_c$.

- Calculate the optimal control sequence $\{u(k), \ldots, u(k + H_c)\}$ that minimizes:

$$
\min_{u(k), \ldots, u(k + H_c)} J(u(k), y(k), w(k))
$$

(4.1)
where $J$ is the cost function given by:

$$J(u(k), y(k), w(k)) = \sum_{i=1}^{H_p} \alpha_i (w(k + i) - \hat{y}(k + i))^2 + \beta_i (\Delta u(k + i - 1))^2 \quad (4.2)$$

Where $u(k)$ represents the inputs, $y(k)$ the outputs and $w(k)$ is the reference signal. The first term minimizes the output errors, and the second term minimizes the control effort.

The predictive control knows the desired reference \textit{a priori} and the system can react before the change has been made, thus delay effects can be avoided.

### 4.3 Fuzzy models in predictive control

The use of nonlinear models increase the complexity of the problem and demands more information from the process. Raise time, gains and data at different operating points are part of the information demanded to construct the model of the process. When nonlinear systems are considered, fuzzy model present a major advantage due to, their ability to handle imprecise information whose complexity can gradually increase as more information is gathered. MBPC controllers take into account the nonlinearities of the plant, implying an improvement in the control performance, reducing also the impact of the disturbances. The application of fuzzy models together with the concept of predictive control allows the achievement of a good control performance.

The defuzzification (turning the fuzzy set into crisp data set) of the output allows the use of fuzzy models for numerical prediction of process behavior, as depicted in Fig. 4.2. In this figure, fuzzy model is acting
as a numerical predictors of the output, and can be directly integrated in MBPC.

Figure 4.2: Fuzzy model in the MBPC scheme

4.4 Branch and bound in model based predictive control

Branch and bound (can be visualized by a search tree, as depicted in the Fig. 4.3) is a general algorithm for finding optimal solutions for optimization problems, especially in discrete and combinatorial optimization. The algorithm can be characterized by three the following rules

1. Branching rule - defines how to divide the problem into sub-problems.
4.4 Branch and bound in model based predictive control

2. Bounding rule - establishes lower and upper bound in the optimal solutions of a sub-problem. These bounds allow for the elimination of sub-problems that do not constitute an optimal solution.

3. Selection rule - defines the next sub-problem to branch from.

Branch and bound (B&B) is applied recursively. When the control actions are discretized, the B&B method can be applied to predictive control. Let us assume that the MIMO model given by

\[ \hat{y}(k+1) = f(x(k)) \] (4.3)

where \( x(k) \subset \mathbb{R}^n \) are the states and \( \hat{y}(k+1) \) are the estimated outputs. The states of the system can be obtained by:

\[ x(k) = [y_1(k), \ldots, y_p(k-p_p+1), u_1(k), \ldots, u_m(k-m_m+1)]^T \] (4.4)

where \( u(k) \subset \mathbb{R}^m \) are the control actions with \( u(k) = [u_1(k), \ldots, u_m(k)] \). The orders of control actions, outputs and states are denoted by \( m, p \) and \( n \), respectively. The function \( f \) relates the state at time \( k+1 \) with the state and the control actions at time \( k \). The parameters \( m_1, \ldots, m_m \) are the order of the inputs \( u_1, \ldots, u_m \), and the parameters \( p_1, \ldots, p_p \) are the orders of the outputs \( y_1, \ldots, y_p \), respectively. Note that the dimension of the state vector is given by \( n = \sum_{j=1}^{m} m_j + \sum_{j=1}^{p} p_j \). Each input \( u_i \) of the system is discretized into \( M \) discrete control actions. Consequently, a discrete control action is represented by \( u_{ij} \), with \( i = 1, \ldots, m \) and \( j = 1, \ldots, M \). The discrete set \( \Psi \) with all possible control actions (as depicted in the Table 4.1) is given by

\[ \Psi = \Psi_1 \times \Psi_2 \times \cdots \times \Psi_m \] (4.5)

where each \( \Psi_i \) represents the complete set of discretized control action for the input \( u_i \)

\[ \Psi = \{ \omega_i | i = 1, 2, \ldots, n_d \} \] (4.6)
The whole number of possible discrete control actions is given by \( S = M^m \). At each time step, control alternatives are considered, yielding a maximum of \( s \) branches. Let \( i = 1, \ldots, H_p \) denote the \( i \)th level of the tree (\( i = 0 \) at initial node), as depicted in the Fig. 4.3, and let \( j \) denote the branch corresponding to the control alternative \( \omega_j \). The cumulative cost at node \( i \) is given by \( J^{(i)} \). Note that no branching takes place beyond the control horizon, and the last control action \( u(k + H_c - 1) \) is applied successively until \( H_p \) is reached. Bounding is applied to reduce the number of alternatives, a particular branch \( j \) at level \( i \) is followed only if the cumulative cost \( J^{(i)} \) is summed with the lower bound on the cost from the level \( i \) to the terminal level \( H_p \). Denoted \( J_L^{(i)} \) is lower than an upper bound of the total cost, denoted \( J_U \)

\[
J^{(i)} + J_L^{(i)} < J_U \tag{4.7}
\]

The lower bound must be estimated, if no estimated is available, it is simply set to zero \( J_L^{(i)} = 0 \). The branch and bound algorithm is presented in Algorithm 4.

The branch and bound optimization, normally introduces oscillations in the controller, known as the chattering effects. In next section a solution for this problem is proposed.
4.4 Branch and bound in model based predictive control

Algorithm 4 Branch and bound

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

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

(4.8)

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

**Step 2: Estimate lower bound.** The algorithm is at level $i$. The branch $j$ with the best cost function $J_j^{(i)}$ found so far, and that is not fully explored, is chosen. This means that if the best cost function at level $i$ has already branches to level $i+1$, the lower bound on the cost over the remaining steps must be estimated. If no estimation is available, it is simply set to zero: $J_L^{(i)} = 0$.

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

$$J_j^{(i)} + J_L^{(i)} < J_U$$

(4.9)

is applied to the considered branch at level $i$ for all $j$, $j = 1,2,\ldots,s$ discretized control actions $\omega_j$. This procedure generates $j$ branches. If no branch is generated go to step 6.

If $i + 1 = H_c$,

**Step 4: Compute a new optimal solution.** Compute the outputs from $H_c$ to $H_p$ and the respective costs. Compare the optimal cost found so far, $J_U$, with this $N$ new costs. If a new optimal solution is found, $J_U$ is replaced by the new $J^{(i)}$. Update the best solution found so far. Eliminate the nodes with non-optimal solutions.

Else

**Step 5: Branch from the best generated node.** Choose the smallest cost $J_j^{(i)}(\omega_j)$ and go to the next level ($i \rightarrow i + 1$). Go to Step 2.

**Step 6: Go up in the tree.** Go up in the levels of tree until a non totally explored branch is found and go to Step 2 afterwards. If all the branches are explored, the optimal solution is the optimal solution found so far, and the algorithm stops.
4.4.1 Fuzzy filters

Basic concepts

The discretization of the control space, allows the branch-and-bound to transform the optimization problem into a searching the best control action problem, this transformation introduces a compromise between the number discrete alternatives and the computational complexity [54]. In order to scale the gain $\xi(k) \in [0, 1]$ for the discrete incremental control actions, the predictive rules consider the errors between the system’s outputs and the desired references. The gain is decreased when the system is close to the steady state situation, and increased when the errors are big or if the references contains sudden changes, allowing the control actions to vary much more.

Figure 4.4 depicts the fuzzy predictive filter [9, 55]. The aforementioned filter reduces the accuracy problem introduced by the discretetization of the control actions, while at the same time the number of necessary control actions are kept low.

Adaptive control alternatives

The fixed set of incremental control alternatives $\Psi = \{\omega_i | i = 1, 2, ..., n_d\}$ is replaced by an adaptive set of control. Let $u_i(k - 1) \in U_i$ represent the control action at time instance $k - 1$, where $U_i = [U_i^-, U_i^+]$ is the domain of each manipulated variable $u_i$. The upper and lower bounds of the possible change in the
4.4 Branch and bound in model based predictive control

The adaptive set of incremental control alternatives for each input is defined as

$$\Psi_k^* = \{0, \sigma_l u^+_k, \sigma_l u^-_k | l = 1, 2, ..., N\}$$

(4.11)

The designer must choose the value for $\sigma_l$. The parameter $l$ determine the number of possible control actions. The maximum number of discrete control actions for each input $u_i$ is given by $V = 2 \times S + 1$ with zero included.

Gain filter

The fuzzy predictive filter applies factors, or gains $\xi(k)$, to the adaptive set of control actions $\Psi_k^*$ in order to obtain a scaled version $\Psi_k$ that is presented to the optimization routine,

$$\Psi_k = \xi(k) \cdot \Psi_k^*$$

(4.12)

The gain $\xi(k)$ must be chosen based on at least the predicted error between the reference and the system’s output. The predicted error is defined as

$$\hat{e}_n(k + H_p) = r_n(k + H_p) - \hat{y}_n(k + H_p)$$

(4.13)

Where $r_n(k + H_p)$ is the reference to be followed at time $k + H_p$. Another important information is the change in the errors, which give an indication on the evolution of the system, and are defined as

$$\Delta e_n(k) = e_n(k) - e_n(k - H_p), n = 1, \ldots, p$$

(4.14)
Considering the predicted error and the change in error, simple heuristic rules can be constructed for the gain. When both $\hat{e}(k + H_p)$ and $\Delta e(k)$ are small, the system is close to a steady state situation. The set of control alternatives should then be scaled down to allow finer control actions, i.e., $\xi(k) \rightarrow 0$, in order to approach zero steady state error without introducing oscillations around the set-point. When the predicted error and the change in error are both high, bigger corrective steps should be taken, i.e., $\xi(k) \rightarrow 1$. The two fuzzy criteria, 'small predicted error' and 'small change in error', are defined by the membership functions $\mu_e(\hat{e}(k + H_p))$ and $\mu_{\Delta e}(\Delta e(k))$, respectively.

Although the performance that MBPC presented here may achieve, they can become not useful when the process change their operational range. The reason is that the models are obtained in a specific operational range, and when the process change, the model can not describe the process behavior in these new operational range. To overcome this situation, next chapter proposes the use of the evolving fuzzy modeling in MBPC. Which allows the model to adapt itself to a new operational range.
4.4 Branch and bound in model based predictive control
Chapter 5

Predictive Fault Tolerant Control Using Evolving Fuzzy Modeling

5.1 Introduction

The concept of using the model based predictive control (MBPC) with evolving Takagi-Sugeno fuzzy modeling (ETS) [4, 5, 6] is proposed in this thesis.

Model based predictive control as presented in Chapter 4, are control techniques which use the process model to predict future outputs. Using MBPC in fault tolerant control (FTC) can allows to use different control specifications in faulty conditions.

The backbone of MBPC techniques is the accuracy of the models, used to predict future outputs. Even using modeling techniques such as neural networks [29], fuzzy logic [7, 47] or neuro-fuzzy [30], the mismatch between the obtained models and the processes are not completely eliminated. What if one could have the guarantee that the mismatches between the models and the processes could be reduced?

The answer for the last question in the fuzzy logic field arises with adaptive fuzzy models and evolving Takagi-Sugeno fuzzy models [4, 5, 6]. These techniques aim to increase the accuracy of model, by reducing the mismatch between the model and the process. This increment in performance is done with recursive
update of the model parameters. The difference between the adaptive fuzzy model and ETS is that in adaptive fuzzy models the number of rules is constant, the clusters centers are not updated and only the parameters of the consequents are recursively updated. Contrarily, in ETS according to certain conditions, new rules are added, the clusters centers are constantly updated and the consequents parameters are also recursively updated. Should be noted that, although one speak about adaptive and evolving, the evolving can be seen as a kind of adaptive and both names are used to distinguish the classical adaptation to the new one the evolving.

In fault conditions or ‘heavy’ disturbances in the process, ETS seems to be more realist, because the addiction of new rules allows to increase the operational ranges (space created by the faults or disturbances) of the model. As the new rules are going to be created in these spaces, their firing strength in these spaces are bigger than the existing ones.

Chapter summarization

In this chapter, both adaptive and ETS fuzzy modeling are used in predictive FTC. The proposed approaches are the issue of Section 5.2. The moving window is the issue of the first subsection, that moving window is used to gather the data, to be used in the FTC approaches. The adaptive based approach in the predictive fault tolerant control scheme proposed is presented in Section 5.2.1 and the evolving fuzzy modeling in the predictive fault tolerant control scheme proposed is the issue of the section 5.2.2.

5.2 Proposed approaches for fault tolerant control

The controller depicted in the Fig. 4.2, may be used as a starting point, for these proposed controller, the reason is that when the system is fault free the model obtained offline are able to describe it, behavior. The presence of a fault or of heavy disturbances may lead the system to a different operational range,
disallowing these models (obtained offline) to perform good predictions. By using ETS, the models are adapted at each time that the process operational range change. That adaption allows the model to “know” the actual operational range, and consequently good predictions of the process behavior are achieved. Differently from the approach in [35] in which, the fault is detected, isolated (fault characterization) and then accommodated. The proposed approach does not have the stage of isolation. The fault is detected when residual passes a certain residual threshold. This residual threshold is defined, by knowing the system behavior. To set this value, one analyze the free fault system output, and choose a value bigger than this value, and use it as the residual threshold. When the fault is detected, the data with fault are gathered with a moving window. The moving window is the issue of the next subsection.

Moving window

When the residual passes the residual threshold, the moving window (MW) begin gathering the system data (output and inputs). If the fault is solved before the MW is full, it is cleaned and it stop gathering data.

The MW as depicted in figure 5.1 has the function of gathering the data. Its size and the number of data removed each time were defined by a trial and error process. The MW size was set to 60 sample data and the data that comes in (points a, b, c and d, these points are only representative) and the removed out (points e, f, g and h) have the size of 10 samples.

When the MW is full, the collected data are used in the FTC in order to adapt the parameters of the fuzzy models. The residual is computed again and, if the fault is not accommodate the process is repeated.

In next Sections the proposed evolving fuzzy modeling approaches in MBPC is presented.
5.2 Proposed approaches for fault tolerant control

5.2.1 Adaptive based approach in the predictive fault tolerant control

Figure 5.2 depicts the proposed adaptive based approach in model based predictive control scheme, applied in to FTC. The fuzzy models presented in this scheme is actually the same. When fault is detected and the MW is full, the collected data are used to adapt the controller parameters. This adaption is carried-out using the adaptive based approach.

5.2.2 Evolving fuzzy modeling in the predictive fault tolerant control

The scheme depicted in the figure 5.2, still be useful, with the difference that in, this approaches instead of using the adaptive based approach algorithm, the adaptation is carried-out by the scatter based approach and the potential based approach algorithms.
Predictive Fault Tolerant Control Using Evolving Fuzzy Modeling

Note that the adaptive based approach can be seen as a part of the ETS. Despite this fact, they are considered different approaches. In the next chapter the FTC approaches proposed in this thesis are applied to a distillation column with the aim of accommodating different faults that may occur in that equipment. Some parameters such as the matrix of adaptive gain $P$ and forgetting factor $\lambda$ are studied.
5.2 Proposed approaches for fault tolerant control
Chapter 6

Application On The Distillation Column

6.1 Introduction

In this chapter, proposed FTC approach is applied to a distillation column. The chapter is organized in the following way. Section 6.2, presents an introduction to the distillation column [16, 20], where the overall theory behind the distillation column is presented. Types of faults and their effects in the distillation column are also presented here. Section 6.3, the fuzzy models of the distillation column are here described. Different fuzzy model structures are tested in order to find appropriate models. Section 6.4, model based predictive control is applied to the distillation column. The effects of the fuzzy filters are presented. Abrupt and incipient faults are simulated on the distillation column. Section 6.5 presents the effects of the forgetting factor and the matrix of the adaptive gain in the adaptive and evolving fuzzy modeling. further, the results of applying adaptive and evolving fuzzy modeling on the distillation column are also presented.

6.2 Distillation column

In chemical industry and petroleum refineries the distillation column is the most important tool to perform the separation of components into more or less pure product streams. This separation is based on the
6.2 Distillation column

difference of volatility among the components. The more volatiles are removed from the top of the column and the less volatiles are removed from the bottom part of the column. The steady state is obtained when the total amount of the feed that is added and the product that is removed are in equal proportions.

Figure 6.1 depicts the boiling point diagram. It is shown how the equilibrium compositions of the components in a liquid mixture vary with temperature at the pressure. In this figure, the upper curve in the diagram (dew-point curve) is the temperature at which the saturated vapor starts to condense and the lower curve (bubble-point curve) is the temperature at which the liquid starts to boil.

According to the nature of the feeding process, distillation column can be:

- Binary column - feed contains two components.
- Multi-component column - feed contains more than two components

The one that is going to be used in this thesis is a binary distillation column, used to separate water and ethanol.

In this thesis, the considered distillation column was the one located in the chemical technology laboratory at "Instituto Superior de Engenharia de Lisboa" (ISEL) with the following features.
• Inside diameter 43\text{mm}

• Structured filling

• Filling has 2\text{m} of height corresponding to 60 trays

• Maximum temperature is 200\degree C

• Allowed pressure 1013 \text{atm} - 0.98 \times 10^{-3} \text{atm}

Figure 6.2 depict the schematic representation of the distillation column located at ISEL. Distillation columns are large scale processes and are regarded as MIMO system [16]. Our interest is the control of the re-boiler temperature and, as the system has multiple inputs, the control problem is going to be a MISO, with the following variables:

Inputs

• Feed flow rate (F0)

• Fraction of feed liquid (q0)

• Reflux valve (F_{\text{reflux}})

• Molar feed fraction of ethanol (x_{F0})

• Re-boiler heat (QR)

Output

• Re-boiler composition (x_{B})
6.2 Distillation column

Figure 6.2: Schematic representation of the distillation column. Reprinted from [39]

Input and output signal values

According to [39], each variable has the area of variation as shown in Table 6.1. Differently to other variables, the reflux valve is a discrete variable, it assumes only the following values: [0.3; 0.4; 0.5].

<table>
<thead>
<tr>
<th>F0 (ml/s)</th>
<th>QR</th>
<th>Freflux</th>
<th>xF0</th>
<th>q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1-1.5]</td>
<td>[0.25-0.29]</td>
<td>[0.3-0.5]</td>
<td>[7-20]%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.1: Limits of variation of the input variables in distillation column
Mathematical model of distillation column

This mathematical model is the one used as the process in the Fig. 5.2. According to [39], for modeling task some assumptions were done:

- Constant pressure (since the top of the distillation column runs in atmospheric pressure and there are low losses of load, make all sense the assumption that the pressure is constant and equal to the atmospheric).

- Constant molar flow (that assumption is only applied when the vaporization enthalpy of the components are similar).

- Equilibrium on all stages (this assumption lead to the consideration that the column efficiency is 100%).

- Total condenser (applied only if the condenser works with temperatures lower than the ebullition point of the light component).

- No vapor holdup (that assumption is acceptable because the vapor molar volume is 1000 times smaller than the liquid molar volume).

- Linear liquid dynamics (the fluid dynamic is considered linear).

With these assumptions the basic equations of the distillation column are:

**Overall material balance on stage i:**

\[
\frac{dM_i}{dt} = L_{i+1} - L_i + V_{i-1} - V_i
\]
6.2 Distillation column

where $L_i$ and $V_i$ are the liquid and the vapor flow from stage $i$

**Material balance for light component on each stage $i$:**

$$\frac{d(M_i x_i)}{dt} = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} + L_i x_i + V_i y_i$$

(6.2)

where $M_i$, $x_i$ and $y_i$ are the liquid holdup, the liquid composition and the vapor composition stage $i$, respectively. The vapor $y_i$ and the liquid $x_i$ compositions are related in the same stage through the following vapor-liquid equation:

$$efp_i = \left(1 - \frac{1}{L_i/V_i + 1}\right)^{ef}$$

(6.3)

$$y_i^r = ef_i y_i + (1 - ef_i) x_i$$

(6.4)

where $efp_i$ is the efficiency in the stray $i$, $y_i^r$ is the real vapor composition in the stray $i$, and $ef$ is a parameter that represent the influence of the ratio $L/V$ in the stray efficiency.

The previous equations are not applied in the re-boiler, feed stage and condenser. To apply them to the aforementioned stages, the effect of any of these stages must be taken into account. Doing so, the following equations are obtained:

**Feed stage**

$$\frac{dM_i}{dt} = L_{i+1} - L_i + V_{i-1} - V_i + F$$

$$\frac{d(M_i x_i)}{dt} = L_{i+1} x_{i+1} + V_{i-1} y_{i-1} + L_i x_i + V_i y_i + F_0$$

(6.5)

**Total condenser**

$$\frac{dM_i}{dt} = V_{i-1} - L_i - D$$

$$\frac{d(M_i x_i)}{dt} = V_{i-1} y_{i-1} - L_i x_i - D x_i$$

(6.6)
Re-boiler

\[ \frac{dM_i}{dt} = L_{i+1} - V_i - B \]

\[ \frac{d(M_i x_i)}{dt} = L_{i+1} x_{i-1} - V_i y_i - B x_i \]  \hspace{1cm} (6.7)

where \( D \) is the distillate flow rate, \( B \) the bottom flow rate, and \( F0 \) is the feed flow rate. In order to control the temperature in the re-boiler, the ethanol composition must be transformed in temperature [39].

\[ T_{\text{reboiler}} = -111.42 x_B + 96.217 \]  \hspace{1cm} (6.8)

where \( T_{\text{reboiler}} \) means temperature of the composition in the re-boiler (which will be the variable to be controlled). This temperature is different from the temperature in the re-boiler due to the heat provided by the electrical resistance. Equation 6.8 was obtained experimentally.

Faults in the distillation column

As the time passes, the performance of the equipments gradually degrades due to deterioration of plant components, by the excessive usage or by bad usage. Those factors lead to unexpected values of process variables. Since the distillation column is a large-scale system, it is therefore normal to expect several numbers of faults, which tend to degrade the overall system performance.

The fault in distillation column may be characterized as [20]:

- **Process loads**: these faults consists of change in
  - Feed flow rate
  - Feed composition
  - Top-product flow rate
  - Base-product flow rate
6.3 Fuzzy modeling of the distillation column

- **Changes in heating and cooling**: changes in the heat input to the re-boiler and changes in the heat output from the condenser.

- **Equipment failing**: heat equipment fails with extensive use. Since controller parameters are typically functions of process parameter, and since failing can cause the change of these parameters, the performance of the control system associated with these devices can deteriorate with time if the controllers are not updated with the failing effects.

For the case study the considered faults are going to be:

- Variation in the feed composition

- Variation in the re-boiler temperature

More details about these faults are presented in Section 6.5.

6.3 Fuzzy modeling of the distillation column

In order to extract the fuzzy models of the distillation column using the fuzzy modeling toolbox (FMT) [8], the following steps are needed:

1. Extract the data (input/output) from the process.

2. Configure the fuzzy model parameters according to the problem in hand.

3. Extract fuzzy models.

To extract the input/output data from the model of the distillation column, the black-box approach is used (in the distillation column), as depicted in the Fig. 6.3. Although in the previous section the distillation
column was presented with 5 inputs, in Fig. 6.3 only the variables used for control are presented. These variables were selected by knowing in advance the effect of each of them in the variable that one is interested in.

The input values for the variables $f_{\text{reflux}}$ and $F_0$ were set according to [39]. The input values are presented in the figure 6.4(a), and the output on the figure 6.4(b).

In order to choose a model to be used in the model based predictive control scheme, different fuzzy models structure. The parameters used in the fuzzy models are shown in the Table 6.2. where $N_y$ is the number of delays in the output, $N_u$ is the number of delays in input, $N_d$ is the number of transport delays and the variables \{a_1, b_1, b_2, c_1, c_2, offset\} are the elements of the consequents parameters vectors presented in 6.11.

The performance indices used to select the structure are:
### 6.3 Fuzzy modeling of the distillation column

<table>
<thead>
<tr>
<th>Structure</th>
<th>Number of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N = 1$</td>
</tr>
<tr>
<td></td>
<td>VAF</td>
</tr>
<tr>
<td>1</td>
<td>$N_y = 2; N_u = [1 1]; N_d = [1 1]$</td>
</tr>
<tr>
<td>2</td>
<td>$N_y = 2; N_u = [3 3]; N_d = [1 1]$</td>
</tr>
<tr>
<td>3</td>
<td>$N_y = 3; N_u = [1 1]; N_d = [2 2]$</td>
</tr>
<tr>
<td>4</td>
<td>$N_y = 1; N_u = [1 1]; N_d = [2 2]$</td>
</tr>
<tr>
<td>5</td>
<td>$N_y = 1; N_u = [2 2]; N_d = [2 1]$</td>
</tr>
<tr>
<td>6</td>
<td>$N_y = 1; N_u = [2 2]; N_d = [1 1]$</td>
</tr>
</tbody>
</table>

Table 6.2: Values used in modeling

- Variance accounted for ($VAF$) which is given by 6.9

- Root mean square ($RMS$) which is given by 6.10

**Variance accounted for ($VAF$)**

\[
VAF = \left[1 - \frac{\text{cov}(y_i - \hat{y}_i)}{\text{cov}(y_i)}\right] \times 100\% \tag{6.9}
\]

**Root mean square ($RMS$)**

\[
RMS = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{6.10}
\]

where $y_i$ is the system output, $\hat{y}_i$ is the predicted output and $N$ the number of samples in data.

By inspecting Table 6.2, it become clear that the structure in row five presents the better values for $VAF$ and $RMS$ for both cases ($N = 1$, $N = 2$). The selected structure is the one in this row with two clusters with the parameters $N_y = 1$, $N_u = [2 2]$, $N_d = [2 1]$.

A fuzzy rule is then given by:

If $y(k-1)$ is $A_{11}$ and $u_1(k-1)$ is $A_{12}$ and $u_1(k-2)$ is $A_{13}$ and $u_2(k-1)$ is $A_{14}$ and $u_2(k-2)$ is $A_{15}$ then $y(k) = a_1 y(k-1) + b_1 u_1(k-1) + b_2 u_1(k-2) - c_1 u_2(k-1) + c_2 u_2(k-2) + offset \tag{6.11}$
It is important to note that the number of rules is equal to the number of clusters.

With the previously parameters values, the obtained model outputs are the ones presented in the Fig. 6.5.

![Fuzzy model performance](image)

**Figure 6.5: Fuzzy model performance**

### 6.4 Predictive control applied to the distillation column

The model selected in the previous section is going to be used to control a distillation column in a MBPC scheme as presented in Chapter 4. Two scenarios are going to be considered. The first one do not use fuzzy predictive filter (FPF) and in second one FPF are considered. The results for both scenarios are depicted in the Fig. 6.6 and 6.7. To evaluate the controller performance, two performance indices are used, the RMS presented in (6.10) and the Non-Dimensional Error Index (NDEI) presented in (6.12). Table 6.3 presents the performance for both scenarios.

\[
NDEI = \frac{RMS}{\text{std}(y)} \quad (6.12)
\]

where $\text{std}$ is the standard deviation of the system output.

The FPF reduces the RMS in 11\% and the NDEI in 13\% and shown in equations 6.13 and 6.14 respectively.
6.4 Predictive control applied to the distillation column

<table>
<thead>
<tr>
<th></th>
<th>RMS</th>
<th>NDEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>without FPF</td>
<td>0.1916</td>
<td>1.2458</td>
</tr>
<tr>
<td>with FPF</td>
<td>0.1705</td>
<td>1.0830</td>
</tr>
</tbody>
</table>

Table 6.3: Comparison of the RMS and the NDEI of the process with and without the FPF

\[
\left[ 1 - \left( \frac{RMS_{with FPF}}{RMS_{without FPF}} \right) \right] \times 100 \quad ; \quad \left[ 1 - \left( \frac{0.1705}{0.1916} \right) \right] \times 100 = 11\% \quad (6.13)
\]

\[
\left[ 1 - \left( \frac{NDEI_{with FPF}}{NDEI_{without FPF}} \right) \right] \times 100 \quad ; \quad \left[ 1 - \left( \frac{1.0830}{1.2458} \right) \right] \times 100 = 13\% \quad (6.14)
\]

Clearly, the FPF introduce an improvement in the controller. Thus now fuzzy predictive filters are going to be used.

Figure 6.6: Control of the distillation column without fuzzy predictive filter

Figure 6.7: Control of the distillation column with fuzzy predictive filter
The present section introduced the controller that is going to be used in evolving FTC. The advantages of using FPF were shown. In Section 6.5, adaptive fuzzy models and the evolving fuzzy modeling (scatter based approach and potential based approach) are going to be applied to the distillation column in faulty conditions.

6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

This Section applied adaptive fuzzy models and evolving fuzzy modeling to the distillation column, a study of the matrix of adaptive gain ($P$) and the forgetting factor ($\lambda$), is performed previously.

6.5.1 Matrix of adaptive gain and forgetting factor effects

In this subsection, the effects of the matrix of adaptive gain and the forgetting effects are presented. There is no specific rules to select these values. According to [54], the initial values for the matrix of adaptive gain is a big value, and they have presented an example using 100, and in the example presented by [5, 6] they used 750. For our case study it is shown to be different as depicted on the Fig. 6.8.

The problem in hand, that one has is to simultaneously determine the matrix of adaptive gain and the forgetting factor, the forgetting factor was fixed to 0.99, and different values were tried for the matrix of adaptive gain, as shown on the Fig. 6.8. Figure 6.9 shows the control behavior for a fixed value of the matrix of adaptive gain and different values for the forgetting factor. Table 6.4 presents the values that are going to be used in FTC approaches. Note that these values were set by a trial and error process.
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

6.5.2 Application of fault tolerant control to the distillation column

Two faults are considered as presented in Section 6.2. A process load fault (variation in feed composition), where the feed composition is characterized by the percentage of ethanol in the composition. Initially the percentage of ethanol in the feed is 9% and the fault is made by decreasing this value to 8%. The second fault is a change in the heating (a variation in the re-boiler temperature). The power in the re-boiler heat is a value with the range [0.25 – 0.29] * 4000W and to simulate a fault in this variable the initial value
0.27 is going to be reduced to 0.25. Although in both faults, the value of the variables were decreased, one could do the inverse process (increasing these values). The reason to present as fault, the decreasing of both variables, is that decreasing the feed composition has the effect of change operational range of the distillation column to high temperatures, and decreasing the re-boiler heat has the effect of change the operational range of the distillation column to lower values. These faults can be seen as actuators malfunctioning. In both cases, incipient and abrupt faults are considered.

Table 6.5 presents the fault characteristics and the control performance. The fault intensity for both faults (abrupt and incipient) are the same, the rising time for the incipient fault, was considered to be 240 s. The process RMS and NDEI in presence of these faults (without using FTC) are also presented.

The following procedure is going to be followed: in the beginning of each Section a figure of the process with fault is depicted. Then, the FTC approaches are applied to the process to accommodate the faults, with the following order: adaptive fuzzy models, potential based approach and scatter based approach.

<table>
<thead>
<tr>
<th></th>
<th>without fault</th>
<th>fault in the feed composition</th>
<th>fault in the re-boiler temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>incipient fault</td>
<td>abrupt fault</td>
</tr>
<tr>
<td>Fault intensity</td>
<td>-</td>
<td>9% → 8%</td>
<td>(0.27 → 0.25) × 4000 W</td>
</tr>
<tr>
<td>RMS</td>
<td>0.1705</td>
<td>0.4328</td>
<td>0.5406</td>
</tr>
<tr>
<td>NDEI</td>
<td>1.0830</td>
<td>1.6162</td>
<td>2.0179</td>
</tr>
</tbody>
</table>

Table 6.5: Characteristics of the applied faults
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Figure 6.9: Behavior of the controller for different values for the forgetting factor

(a) Forgetting factor 0.85  
(b) Forgetting factor 0.90  
(c) Forgetting factor 0.99

Next, a figure containing the process using FTC is depicted another figure contained the manipulated variables behavior is also shown. For the sake of simplicity, the consequents parameters behavior for the three FTC approaches, are only presented for the incipient fault in the feed composition. For abrupt fault, the consequents parameters behavior, the initial rule bases and the final rule bases are presented in the appendix B for the scatter approach.

The reason to present initial rules and the final rules for the scatter approach is that in this approach a new rule is added and is convenient to observe its influence.
Incipient fault conditions in feed concentration

From now on, FTC approaches are going to be applied on the distillation column. In Fig. 6.10, an incipient fault was applied to the distillation column. The fault starts at time 120 s. and raised in 240 s. It can be seen in this figure that the operational range of the distillation column has changed.

Adaptive fuzzy models in incipient fault conditions

Applying adaptive fuzzy models in the FTC approach on the aforementioned situation, the results are presented in the Fig. 6.11 and 6.12. From the Fig. 6.12 one may see that the controller adjusted five times the parameters of the consequents until it achieved constant values. However, these adjustment were not done in all parameters. The first adjustment of the consequents occurred at 160 s. The reason to happen in that time is, as the fault is incipient, it takes too long until the residual passes the threshold and the FTC, become activated.
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Figure 6.11 show that, although the variations on the parameters consequents, the reflux variable does not change. However, the feed flow rate changed its value, from the moment which happens the first change in the parameters of the consequents, until constant parameters were achieved. As a conclusion one may say that the presented FTC approach accommodates the fault.

Figure 6.11: Using adaptive fuzzy models to accommodate incipient faults in the feed composition

Figure 6.12: Consequents parameters behavior for the fuzzy rules
Potential based approaches in incipient fault conditions

Differently from the adaptive fuzzy modeling approach, in the potential based approach the controller

![Figure 6.13: Using potential based approach to accommodate incipient faults in the feed composition](image)

(a) Changes in the parameters of the consequents of the first rule
(b) Changes in the parameter of the consequents of the second rule

![Figure 6.14: Consequents parameters behavior for the fuzzy rules](image)

adjusted only four times the parameters of the consequents until it achieved the good values. The fault was also accommodated.
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Scatter based approaches in incipient fault conditions

In this approach, a new rule was added (the third rule). As this rule did not exist until the controller

![Figure 6.15: Using scatter based approach to accommodate incipient faults in the feed composition](image)

(a) Changes in the parameters of the consequents of the first rule

(b) Changes in the parameters of the consequents of the second rule

![Figure 6.16: Consequents parameters behavior for the fuzzy rules](image)

adjusted the consequents, from the Fig. 6.17 one may see that until it happens, the parameters were set to zero.

With this approach, until the good values for the consequents were achieve, the controller adjusted the
parameter eight times. Although the controller accommodated the faults, it show to be less stable than the others. From the Fig. 6.15, one may see that the there are variations in the feed flow rate, with turn the controller impracticable.

![Image of Figure 6.17: Changes in the parameters of the consequents of the third rule](image)

Table 6.6: Comparison of the RMS and NDEI for the three approaches in presence of incipient fault in feed composition

<table>
<thead>
<tr>
<th>Approach</th>
<th>RMS</th>
<th>NDEI</th>
<th>PI RMS</th>
<th>New rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process without fault</td>
<td>0.1705</td>
<td>1.0830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process with fault</td>
<td>0.4328</td>
<td>1.6162</td>
<td>61%</td>
<td>0</td>
</tr>
<tr>
<td>Using the adaptive fuzzy modeling</td>
<td>0.1756</td>
<td>1.0428</td>
<td>3%</td>
<td>0</td>
</tr>
<tr>
<td>Using the potential based approach</td>
<td>0.1715</td>
<td>1.0663</td>
<td>0.6%</td>
<td>0</td>
</tr>
<tr>
<td>Using the scatter based approach</td>
<td>0.1834</td>
<td>1.1303</td>
<td>7%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.6: Comparison of the RMS and NDEI for the three approaches in presence of incipient fault in feed composition

Table 6.6 presents the used approaches with the RMS and NDEI performances indices, the improvement is given by PI RMS, meaning percentage of improvement using the RMS. On analyzing these values, it should be noted that, the better performances are achieved, with the PI RMS close to 0, because although one call it improvement, in fact what is computed is the deteriorate percentage of FTC approaches. From this table one conclude that the potential based approach presents the better performance.
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Abrupt fault conditions in feed concentration

![Graph showing re-boiler temperature over time with reference and process lines]

Figure 6.18: Process with abrupt fault in the feed composition

Adaptive fuzzy models in abrupt fault conditions

![Graphs showing adaptive fuzzy models to accommodate abrupt fault in the feed composition]

Figure 6.19: Using adaptive fuzzy models to accommodate abrupt fault in the feed composition
Potential based approach in abrupt fault conditions

Figure 6.20: Using potential based approach to accommodate abrupt fault in the feed composition

Scatter based approach in abrupt fault conditions

Figure 6.21: Using scatter based approach to accommodate abrupt fault in the feed composition
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

<table>
<thead>
<tr>
<th></th>
<th>RMS</th>
<th>NDEI</th>
<th>PI RMS</th>
<th>New rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process without fault</td>
<td>0.1705</td>
<td>1.0830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process with fault</td>
<td>0.5406</td>
<td>2.0179</td>
<td>68.5%</td>
<td></td>
</tr>
<tr>
<td>Using the adaptive fuzzy modeling</td>
<td>0.2161</td>
<td>1.0359</td>
<td>21%</td>
<td>0</td>
</tr>
<tr>
<td>Using the potential based approach</td>
<td>0.2478</td>
<td>1.1332</td>
<td>31%</td>
<td>0</td>
</tr>
<tr>
<td>Using the scatter based approach</td>
<td>0.2036</td>
<td>1.0520</td>
<td>16%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.7: Comparison of the RMS and NDEI for the three approaches in presence of abrupt fault in feed composition

Faults in the feed composition are now considered. Different approaches were used to accommodate these faults, from the Fig. 6.11 to 6.21 one may see that, effectively, these approaches can accommodate the fault. With the percentage of improvement shown in Tables 6.6 and 6.7 one can not conclude which approach it best in FTC. In both cases (incipient and abrupt) a new rule was created when the scatter based approach is used. The reason is that the scatter based approach showed to be more sensitive to new data.

Clearly, one may say that the proposed FTC approaches are useful for fault accommodation.

In the following Section faults in re-boiler temperature are going to be considered and as the explanation are similar, one is going to explain only the results in the end of the subsection.
Incipient fault conditions in re-boiler temperature

![Graph of re-boiler temperature over time](image)

Figure 6.22: Process with incipient fault in the re-boiler temperature

Adaptive fuzzy models in incipient fault conditions

![Graphs of adaptive fuzzy models](image)

Figure 6.23: Using adaptive fuzzy models to accommodate Incipient faults in the re-boiler temperature
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Potential based approach in incipient fault conditions

Figure 6.24: Using potential based approach to accommodate incipient faults in the re-boiler temperature

Scatter based approach in incipient fault conditions

Figure 6.25: Using scatter based approach to accommodate incipient faults in the re-boiler temperature
Table 6.8: Comparison of the RMS and NDEI for the three approaches in presence of incipient fault in the re-boiler temperature

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>NDEI</th>
<th>PI RMS</th>
<th>New rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process without fault</td>
<td>0.1705</td>
<td>1.0830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process with fault</td>
<td>0.2293</td>
<td>1.0913</td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>Using the adaptive fuzzy modeling</td>
<td>0.1753</td>
<td>1.0076</td>
<td>3%</td>
<td>0</td>
</tr>
<tr>
<td>Using the potential based approach</td>
<td>0.1714</td>
<td>1.0019</td>
<td>0.5%</td>
<td>0</td>
</tr>
<tr>
<td>Using the scatter based approach</td>
<td>0.1721</td>
<td>1.0019</td>
<td>0.9%</td>
<td>1</td>
</tr>
</tbody>
</table>

Abrupt fault conditions in re-boiler temperature

Figure 6.26: Process with abrupt fault in re-boiler temperature
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control

Adaptive fuzzy models in abrupt fault conditions

![Graphs showing adaptive fuzzy models in re-boiler temperature](image1.png)

Figure 6.27: Using adaptive fuzzy models to accommodate abrupt fault in re-boiler temperature

Potential based approach in abrupt fault conditions

![Graphs showing potential based approach in re-boiler temperature](image2.png)

Figure 6.28: Using potential based approach to accommodate abrupt fault in re-boiler temperature
Scatter based approach in abrupt fault conditions

Figure 6.29: Using scatter based approach to accommodate abrupt fault in re-boiler temperature

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>NDEI</th>
<th>PI RMS</th>
<th>New rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process without fault</td>
<td>0.1705</td>
<td>1.0830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process with fault</td>
<td>0.2725</td>
<td>1.2654</td>
<td>37%</td>
<td></td>
</tr>
<tr>
<td>Using the adaptive fuzzy modeling</td>
<td>0.1828</td>
<td>1.0216</td>
<td>7%</td>
<td>0</td>
</tr>
<tr>
<td>Using the potential based approach</td>
<td>0.1779</td>
<td>1.0160</td>
<td>4%</td>
<td>0</td>
</tr>
<tr>
<td>Using the scatter based approach</td>
<td>0.1847</td>
<td>1.0112</td>
<td>8%</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.9: Comparison of the RMS and NDEI for the three approaches in presence of abrupt fault in the re-boiler temperature

Faults in the re-boiler temperature were considered, Fig. 6.23 to 6.29. depicts how the FTC approaches accommodate the faults. Tables 6.8 and 6.9 present the percentage of improvement of these approaches, and clearly, one may say that the potential based approach accommodates the faults in the re-boiler better than the others approaches.

The conclusions of the developed thesis are the issue of the next chapter, and the recommendation for future work is also made in the next chapter.
6.5 Adaptive and evolving fuzzy modeling in the fault tolerant control
Chapter 7

Conclusions

The conclusions of the developed work are presented in this final chapter.

7.1 Fault tolerant control approaches

This thesis proposed the use of adaptive and evolving fuzzy modeling approaches in fault tolerant control, and from the results presented one concludes that the main objective was achieved.

The adaptive fuzzy modeling was easier to be used because the values of the matrix of adaptive gain were the same in all cases. The potential based approach, one variant of evolving Takagi-Sugeno fuzzy models, showed to accommodate the faults without having to add new rules, and from the results, one may conclude that, in general, this approach led to better results.

The scatter based approach, the other variant of evolving Takagi-Sugeno fuzzy models, leaded also to good control performance. However, this approach shown to be more sensitive to new data than the potential based approach, and in all cases in which it was used, new rules were added.
7.2 Influence of the adaptive gain matrix and the forgetting factor

Both adaptive gain matrix and forgetting factor have important roles in the controller, as presented in Section 6.5.1. Adaptive fuzzy models as presented in the table 6.4, used the same values of the matrix of adaptive gain. The potential approach used same values of the matrix of adaptive gain for each fault. However, in the scatter approach the values of the matrix of adaptive gain were different. For all the approaches, the value for the forgetting factor was the same. However, we believe that this value might vary slightly.

7.3 Future work

Despite the results presented by the scatter based approach, we believe that these results can be improved. This improvement can be made by searching better values for the matrix of adaptive gain and for the forgetting factor. In order to search for better values for these parameters, optimization algorithms such as genetic algorithms, ant colony and swarm optimization could be useful.

Unfortunately, the fault tolerant controller presented here was not applied in to real distillation column. Efforts have been done in order to make it possible.

The proposed FTC approaches were only applied to two different faults. However, other faults may be considered. The control performance can be tested in the future in faults as the ones described in Section 6.2.
Bibliography


BIBLIOGRAPHY


[59] D.O Thompson and Jr L.W Schumerr. Quantitative model-based approaches to inspectability and


Appendix

Appendix A

Rules and center cluster for the used model

1. If \( y(k-1) \) is \( A_{11} \) and \( u_1(k-1) \) is \( A_{12} \) and \( u_1(k-2) \) is \( A_{13} \) and \( u_2(k-1) \) is \( A_{14} \) and \( u_2(k-2) \) is \( A_{15} \) then
\[
y(k) = 0.93y(k-1) + 0.0u_1(k-1) + 0.59u_1(k-2) + 0.22u_2(k-1) + 0.044u_2(k-2) + 5.6
\]

2. If \( y(k-1) \) is \( A_{21} \) and \( u_1(k-1) \) is \( A_{22} \) and \( u_1(k-2) \) is \( A_{23} \) and \( u_2(k-1) \) is \( A_{24} \) and \( u_2(k-2) \) is \( A_{25} \) then
\[
y(k) = 0.95y(k-1) + 0.17u_1(k-1) + 0.23u_1(k-2) + 0.0u_2(k-1) + 0.27u_2(k-2) + 3.8
\]

(7.1)

| Table 7.1: Consequent parameters of the fuzzy model. |
|---|---|---|---|---|---|
| rule | \( y(k-1) \) | \( u_1(k-1) \) | \( u_1(k-2) \) | \( u_2(k-1) \) | \( u_2(k-2) \) | offset |
| 1 | \( 9.3 \cdot 10^{\cdot -1} \) | \( 0.0 \cdot 10^{\cdot +1} \) | \( 5.9 \cdot 10^{\cdot -1} \) | \( +2.2 \cdot 10^{\cdot -1} \) | \( 4.4 \cdot 10^{\cdot -2} \) | \( 5.6 \cdot 10^{\cdot +} \) |
| 2 | \( 9.5 \cdot 10^{\cdot -1} \) | \( 1.7 \cdot 10^{\cdot -1} \) | \( 2.3 \cdot 10^{\cdot -1} \) | \( 0.0 \cdot 10^{\cdot +} \) | \( +2.7 \cdot 10^{\cdot -1} \) | \( 3.8 \cdot 10^{\cdot +} \) |

| Table 7.2: Cluster centers. |
|---|---|---|---|---|
| rule | \( y(k-1) \) | \( u_1(k-1) \) | \( u_1(k-2) \) | \( u_2(k-1) \) | \( u_2(k-2) \) |
| 1 | \( 8.9 \cdot 10^{\cdot +1} \) | \( 3.8 \cdot 10^{\cdot -1} \) | \( 3.8 \cdot 10^{\cdot -1} \) | \( 1.3 \cdot 10^{\cdot +} \) | \( 1.3 \cdot 10^{\cdot +} \) |
| 2 | \( 9.1 \cdot 10^{\cdot +1} \) | \( 4.5 \cdot 10^{\cdot -1} \) | \( 4.5 \cdot 10^{\cdot -1} \) | \( 1.0 \cdot 10^{\cdot +} \) | \( 1.0 \cdot 10^{\cdot +} \) |
Appendix B

Initial rules:

Equal to appendix A.

Final rules:

Adaptive approach

1. If $y(k-1)$ is $A_{11}$ and $u_1(k-1)$ is $A_{12}$ and $u_1(k-2)$ is $A_{13}$ and $u_2(k-1)$ is $A_{14}$ and $u_2(k-2)$ is $A_{15}$ then

   $y(k) = 0.93g(k-1) + 2.3 \cdot 10^{-6}u_1(k-1) + 0.59u_1(k-2) + 0.22u_2(k-1) + 0.045u_2(k-2) + 5.6$

2. If $y(k-1)$ is $A_{21}$ and $u_1(k-1)$ is $A_{22}$ and $u_1(k-2)$ is $A_{23}$ and $u_2(k-1)$ is $A_{24}$ and $u_2(k-2)$ is $A_{25}$ then

   $y(k) = 0.95g(k-1) + 0.17u_1(k-1) + 0.23u_1(k-2) + 0.021u_2(k-1) + 0.24u_2(k-2) + 3.8$

Scatter approach

1. If $y(k-1)$ is $A_{11}$ and $u_1(k-1)$ is $A_{12}$ and $u_1(k-2)$ is $A_{13}$ and $u_2(k-1)$ is $A_{14}$ and $u_2(k-2)$ is $A_{15}$ then

   $y(k) = 0.93g(k-1) - 1.5 \cdot 10^{-4}u_1(k-1) + 0.59u_1(k-2) + 0.22u_2(k-1) + 0.047u_2(k-2) + 5.6$

2. If $y(k-1)$ is $A_{21}$ and $u_1(k-1)$ is $A_{22}$ and $u_1(k-2)$ is $A_{23}$ and $u_2(k-1)$ is $A_{24}$ and $u_2(k-2)$ is $A_{25}$ then

   $y(k) = 0.95g(k-1) + 0.15u_1(k-1) + 0.20u_1(k-2) + 0.044u_2(k-1) + 0.21u_2(k-2) + 3.8$

3. If $y(k-1)$ is $A_{31}$ and $u_1(k-1)$ is $A_{32}$ and $u_1(k-2)$ is $A_{33}$ and $u_2(k-1)$ is $A_{34}$ and $u_2(k-2)$ is $A_{35}$ then

   $y(k) = 1.0g(k-1) + 0.0043u_1(k-1) + 0.0015u_1(k-2) + 0.0058u_2(k-1) + 0.0061u_2(k-2) + 0.010$

Potential approach

1. If $y(k-1)$ is $A_{11}$ and $u_1(k-1)$ is $A_{12}$ and $u_1(k-2)$ is $A_{13}$ and $u_2(k-1)$ is $A_{14}$ and $u_2(k-2)$ is $A_{15}$ then

   $y(k) = 0.93g(k-1) + 1.8 \cdot 10^{-4}u_1(k-1) + 0.59u_1(k-2) + 0.22u_2(k-1) + 0.045u_2(k-2) + 5.6$

2. If $y(k-1)$ is $A_{21}$ and $u_1(k-1)$ is $A_{22}$ and $u_1(k-2)$ is $A_{23}$ and $u_2(k-1)$ is $A_{24}$ and $u_2(k-2)$ is $A_{25}$ then

   $y(k) = 0.95g(k-1) + 0.15u_1(k-1) + 0.23u_1(k-2) + 0.0068u_2(k-1) + 0.23u_2(k-2) + 3.8$