Fuzzy Systems, Modeling and Identification

Article *in* Electrical Engineering · January 2001

1 author:

Robert Babuska
Delft University of Technology

Fuzzy Systems, Modeling and Identification

Robert Babuška

Delft University of Technology, Department of Electrical Engineering
Control Laboratory, Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands
tel: +31 15 785117, fax: +31 15 2786679, e-mail: r.babuska@et.tudelft.nl

Summary

This text provides an introduction to the use of fuzzy sets and fuzzy logic for the approximation of functions and modeling of static and dynamic systems. The concept of a fuzzy system is first explained. Afterwards, the motivation and practical relevance of fuzzy modeling are highlighted. Two types of rule-based fuzzy models are described: the linguistic (Mamdani) model and the Takagi–Sugeno model. For each model, the structure of the rules, the inference and defuzzification methods are presented. Fuzzy modeling of dynamic systems is addressed, as well as the methods to construct fuzzy models from knowledge and data (measurements). Illustrative examples are given throughout the text. At the end, homework problems are included. MATLAB programs implementing some of the examples are available from the author. The reader is encouraged to study and possibly modify these examples in order to get a better insight in the methods presented.

Preface

Prerequisites: This text provides an introduction to the use of fuzzy sets and fuzzy logic for the approximation of functions and modeling of static and dynamic systems. It is assumed that the reader has basic knowledge of set and fuzzy set theory (membership functions, operations on fuzzy sets – union, intersection and complement, fuzzy relations, max-min composition, extension principle), mathematical analysis (univariate and multivariate functions, composition of functions), and linear algebra (system of linear equations, least-square solution).

Organization. The material is organized in five sections: In the Introduction, different modeling paradigms are first presented. Then, the concept of a fuzzy system is first explained and the motivation and practical relevance of fuzzy modeling are highlighted. Section 2 describes two types of rule-based fuzzy models: the linguistic (Mamdani) model and the Takagi–Sugeno model. For each model, the structure of the rules, the inference and defuzzification methods are presented. At the end of this section, fuzzy modeling of dynamic systems is addressed. In Section 3, methods to construct fuzzy models from knowledge and numerical data are presented. Section 4 reviews some engineering applications of fuzzy modeling, and the concluding Section 5 gives a short summary. Illustrative examples are provided throughout the text, and at the end, homework problems are included. Some of the numerical examples given have been implemented in MATLAB. The code is available from the author on request. The reader is encouraged to study and possibly modify these examples in order to get a better insight in the methods presented. A subject index is provided for a quick reference.

Aims: After studying the material, the reader should be able to:

- Characterize a fuzzy system and give some examples of fuzzy systems.
- Define the linguistic (Mamdani) and the Takagi-Sugeno fuzzy model in terms of their structure, inference and defuzzification mechanisms.
- Explain how dynamic systems are represented by fuzzy models, give examples.
- List the steps and choices in the knowledge-based design of fuzzy models.
- Name and briefly characterize the presented techniques for data-driven acquisition and tuning of fuzzy models.

Further reading. Readers interested in a detailed and fundamental treatment of fuzzy set theory and fuzzy logic can consult research monographs by Dubois and Prade (1980) or Klir and Yuan (1995). Basic, as well as more advanced concepts of fuzzy modeling and control, are presented, for instance, by Pedrycz (1993), Driankov, et al. (1993) or Yager and Filev (1994).

Mathematical notation. Throughout the text, the following conventions are used. Lower case characters in italics, such as x or y_i, denote scalar variables and elements of vectors. Vectors are printed in bold, i.e., $\mathbf{x}$ denotes a column vector. A row vector is denoted by the transpose operator, e.g., $\mathbf{x}^T$. Upper case bold characters denote matrices, for instance, $\mathbf{X}$ is a matrix. Upper case italic characters such as A denote crisp and fuzzy sets. A linguistic variable (a variable whose values are fuzzy sets) is denoted by $\tilde{x}$. The term “crisp” is used as an opposite to fuzzy. For instance, a fuzzy number is a normal convex fuzzy set, while a crisp number may by a real or an integer number. A list of the used mathematical symbold is included in Appendix A.

Acknowledgement. I am grateful to Piet Bruijn and Govert Monsees who read drafts of this text and contributed by their useful comments.

Contents

1 Introduction **1**

1.1 Fuzzy systems . . . 2

1.2 Practical relevance of fuzzy modeling . . . 4

2 Rule-Based Fuzzy Models **5**

2.1 Linguistic fuzzy model . . . 5

2.1.1 Relational representation of a linguistic model . . . 6

2.1.2 Max-min (Mamdani) inference . . . 8

2.1.3 Multivariable systems . . . 10

2.1.4 Defuzzification . . . 12

2.1.5 Singleton model . . . 12

2.2 Takagi–Sugeno model . . . 14

2.2.1 Inference mechanism . . . 14

2.2.2 TS model as a quasi-linear systems . . . 14

2.3 Modeling dynamic systems . . . 16

3 Building Fuzzy Models **16**

3.1 Structure and parameters . . . 17

3.2 Knowledge-based design . . . 19

3.3 Data-driven acquisition/tuning of fuzzy models . . . 19

3.3.1 Least-squares estimation of consequents . . . 19

3.3.2 Template-based modeling . . . 20

3.3.3 Neuro-fuzzy modeling . . . 22

3.3.4 Fuzzy clustering . . . 23

4 Overview of Applications **24**

5 Summary and Concluding Remarks **25**

A List of Symbols **27**

Subject index **30**

1 Introduction

Developing mathematical models of real systems is a central topic in many disciplines of engineering and science. Models can be used for simulations, analysis of the system's behavior, better understanding of the underlying mechanisms in the system, design of new processes, or design of controllers.

Traditionally, modeling is seen as a conjunction of a thorough understanding of the system's nature and behavior, and of a suitable mathematical treatment that leads to a usable model. This approach is usually termed *"white-box"* (physical, mechanistic, first-principle) modeling. However, the requirement for a good understanding of the physical background of the problem at hand proves to be a severe limiting factor in practice, when complex and poorly understood systems are considered. Difficulties encountered in conventional white-box modeling can arise, for instance, from poor understanding of the underlying phenomena, inaccurate values of various process parameters, or from the complexity of the resulting model. A complete understanding of the underlying mechanisms is virtually impossible for a majority of real systems. However, gathering an acceptable degree of knowledge needed for physical modeling may be a very difficult, time-consuming and expensive or even impossible task. Even if the structure of the model is determined, a major problem of obtaining accurate values for the parameters remains. It is the task of system identification to estimate the parameters from data measured on the system. Identification methods are currently developed to a mature level for linear systems only. Most real processes are, however, nonlinear and can be approximated by linear models only locally.

A different approach assumes that the process under study can be approximated by using some sufficiently general *"black-box"* structure used as a general function approximator. The modeling problem then reduces to postulating an appropriate structure of the approximator, in order to correctly capture the dynamics and nonlinearity of the system. In black-box modeling, the structure of the model is hardly related to the structure of the real system. The identification problem consists of estimating the parameters of the model. If representative process data are available, black-box models usually can be developed quite easily, without requiring process-specific knowledge. A severe drawback of this approach is that the structure and parameters of these models usually do not have any physical significance. Such models cannot be used for analyzing the system's behavior otherwise than by numerical simulation, cannot be scaled up or down when moving from one process scale to another, and therefore are less useful for industrial practice.

There is a range of modeling techniques that attempt to combine the advantages of the white-box and black-box approaches, such that the known parts of the system are modeled using physical knowledge, and the unknown or less certain parts are approximated in a black-box manner, using process data and black-box modeling structures with suitable approximation properties. These methods are often denoted as hybrid, semi-mechanistic or *"gray-box"* modeling.

A common drawback of most standard modeling approaches is that they cannot make effective use of extra information, such as the knowledge and experience of engineers and operators, which is often imprecise and qualitative in its nature. The fact that humans are often able to manage complex tasks under significant uncertainty has stimulated the search for alternative modeling and control paradigms. So-called "intelligent" modeling and control methodologies, which employ techniques motivated by biological systems and human intelligence to develop models and controllers for dynamic systems, have been introduced. These techniques explore alternative representation schemes, using, for instance, natural language, rules, semantic networks or qualitative models, and possess formal methods to incorporate extra relevant information. Fuzzy modeling and control are typical examples of techniques that make use of human knowledge and deductive processes. Artificial neural networks, on the other hand, realize learning and adaptation capabilities by imitating the functioning of biological neural systems on a simplified level. The different modeling paradigms are summarized in Tab. 1.

Table 1. *Different modeling paradigms.*

modeling approach	source of information	method of acquisition	example	deficiency
mechanistic (white-box)	formal knowledge and data	mathematical (Lagrange eq.)	differential equations	cannot use “soft” knowledge
black-box	data	optimization (learning)	regression, neural network	cannot at all use knowledge
fuzzy	various knowledge and data	knowledge-based + learning	rule-based model	“curse” of dimensionality

1.1 Fuzzy systems

A static or dynamic system which makes use of fuzzy sets or fuzzy logic and of the corresponding mathematical framework is called a *fuzzy system*. There are a number of ways fuzzy sets can be involved in a system, such as:

- *In the description of the system.* A system can be defined, for instance, as a collection of if-then rules with fuzzy predicates, or as a fuzzy relation. An example of a fuzzy rule describing the relationship between a heating power and the temperature trend in a room may be:

  If *the heating power is high* **then** *the temperature will increase fast*.

- *In the specification of the system's parameters.* The system can be defined by an algebraic or differential equation, in which the parameters are fuzzy numbers instead of real numbers. As an example consider an equation: $y = \tilde{3}x_1 + \tilde{5}x_2$, where $\tilde{3}$ and $\tilde{5}$ are fuzzy number “about three” and “about five”, respectively, defined by membership functions. Fuzzy numbers express the uncertainty in the parameter values.

- *The input, output and state variables of a system may be fuzzy sets.* Fuzzy inputs can be readings from unreliable sensors (“noisy” data), or quantities related to human perception, such as comfort, beauty, etc. Fuzzy systems can process such information, which is not the case with conventional (crisp) systems.

A fuzzy system can simultaneously have several of the above attributes. Table 2 gives an overview of the relationships between fuzzy and crisp system descriptions and variables. In this text we will focus on the last type of systems, i.e., fuzzily described systems with crisp or fuzzy inputs.

[1] By means of the extension principle a crisp function can be evaluated for a fuzzy argument (Zadeh, 1975).

Table 2. *Crisp and fuzzy information in systems.*

system description	input data	resulting output data	mathematical framework
crisp	crisp	crisp	functional analysis, linear algebra, etc.
crisp	fuzzy	fuzzy	extension principle[1]
fuzzy	crisp/ fuzzy	fuzzy	fuzzy relational calculus, fuzzy inference

Fuzzy systems can be regarded as a generalization of interval-valued systems, which are in turn a generalization of crisp systems. This is depicted in Fig. 1 which gives an example of a function and its interval and fuzzy forms. The evaluation of the function for crisp, interval and fuzzy data is schematically depicted as well. Note that a function $f: X \to Y$ can be regarded as a subset of the Cartesian product $X \times Y$, i.e., as a *relation*. The evaluation of the function for a given input proceeds in three steps: 1) extend the given input into the product space $X \times Y$ (vertical dashed lines in Fig. 1), 2) find the intersection of this extension with the relation, 3) project this intersection onto Y (horizontal dashed lines in Fig. 1). This view is independent of the nature of both the function and the data (crisp, interval, fuzzy). Remember this view of function evaluation, as will help you to understand the use of fuzzy relations for inference in fuzzy modeling.

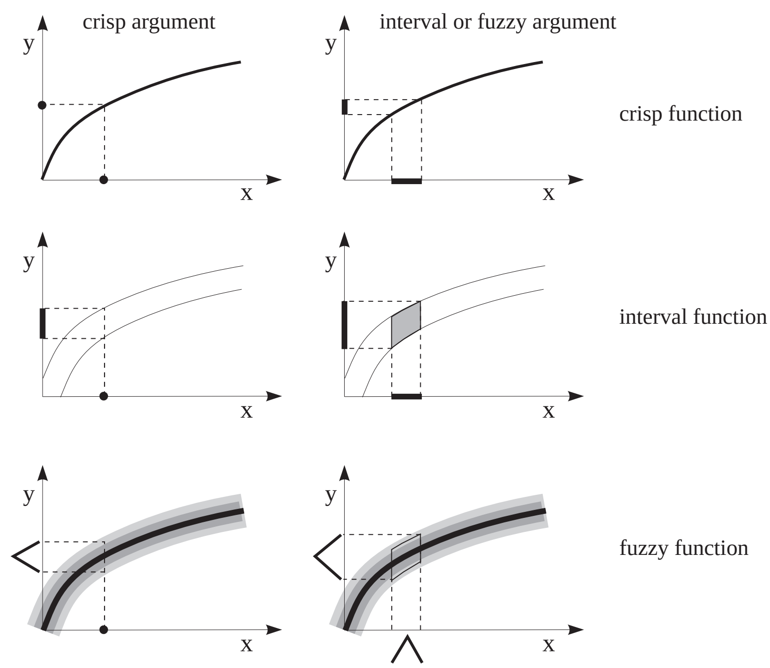


Figure 1. *Evaluation of a crisp, interval and fuzzy function for crisp, interval and fuzzy arguments.*

Most common are fuzzy systems defined by means of if-then rules: rule-based fuzzy systems. In the rest of this text we will focus on these systems only. Fuzzy systems can serve different purposes, such as modeling, data analysis, prediction or control. In this text a fuzzy rule-based system is for simplicity called a *fuzzy model*, regardless of its eventual purpose.

1.2 Practical relevance of fuzzy modeling

Incomplete or vague knowledge about systems. Conventional system theory relies on crisp mathematical models of systems, such as algebraic and differential or difference equations. For some systems, such as electro-mechanical systems, mathematical models can be obtained. This is because the physical laws governing the systems are well understood. For a large number of practical problems, however, the gathering of an acceptable degree of knowledge needed for physical modeling is a difficult, time-consuming and expensive or even impossible task. In the majority of systems, the underlying phenomena are understood only partially and crisp mathematical models cannot be derived or are too complex to be useful. Examples of such systems can be found in the chemical or food industries, biotechnology, ecology, finance, sociology, etc. A significant portion of information about these systems is available as the knowledge of human experts, process operators and designers. This knowledge may be to vague and uncertain to be expressed by mathematical functions. It is, however, often possible to describe the functioning of systems by means of natural language, in the form of if-then rules. Fuzzy rule-based systems can be used as knowledge-based models constructed by using knowledge of experts in the given field of interest (Pedrycz, 1990; Yager and Filev, 1994). From this point of view, fuzzy systems are similar to *expert systems* studied extensively in the "symbolic" artificial intelligence (Buchanan and Shortliffe, 1984; Patterson, 1990).

Adequate processing of imprecise information. Precise numerical computation with conventional mathematical models only makes sense when the parameters and input data are accurately known. As this is often not the case, a modeling framework is needed which can adequately process not only the given data, but also the associated uncertainty. The stochastic approach is a traditional way of dealing with uncertainty. However, it has been recognized that not all types of uncertainty can be dealt with within the stochastic framework. Various alternative approaches have been proposed (Smets, et al., 1988), fuzzy logic and set theory being one of them.

Transparent (gray-box) modeling and identification. Identification of dynamic systems from input-output measurements is an important topic of scientific research with a wide range of practical applications. Many real-world systems are inherently nonlinear and cannot be represented by linear models used in conventional system identification (Ljung, 1987). Recently, there is a strong focus on the development of methods for the identification of nonlinear systems from measured data. Artificial neural networks and fuzzy models belong to the most popular model structures used. From the input-output view, fuzzy systems are flexible mathematical functions which can approximate other functions or just data (measurements) with a desired accuracy. This property is called *general function approximation* (Kosko, 1994; Wang, 1994; Zeng and Singh, 1995). Compared to other well-known approximation techniques such as artificial neural networks, fuzzy systems provide a more transparent representation of the system under study, which is mainly due to the possible linguistic interpretation in the form of rules. The logical structure of the rules facilitates the understanding and analysis of the model in a semi-qualitative manner, close to the way humans reason about the real world.

2 Rule-Based Fuzzy Models

In rule-based fuzzy systems, the relationships between variables are represented by means of fuzzy if–then rules of the following general form:

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

The antecedent proposition is always a fuzzy proposition of the type "$\tilde{x}$ **is** A" where $\tilde{x}$ is a linguistic variable and A is a linguistic constant (term). The proposition's truth value (a real number between zero and one) depends on the degree of match (similarity) between $\tilde{x}$ and A. Depending on the form of the consequent two main types of rule-based fuzzy models are distinguished:

- *Linguistic fuzzy model*: both the antecedent and the consequent are fuzzy propositions.
- *Takagi–Sugeno (TS) fuzzy model*: the antecedent is a fuzzy proposition, the consequent is a crisp function.

These two types of fuzzy models detailed in the subsequent sections.

2.1 Linguistic fuzzy model

The linguistic fuzzy model (Zadeh, 1973; Mamdani, 1977) has been introduced as a way to capture available (semi-)qualitative knowledge in the form of if–then rules:

$$\mathcal{R}_i\colon \textbf{ If } \tilde{x} \textbf{ is } A_i \textbf{ then } \tilde{y} \textbf{ is } B_i, \qquad i = 1, 2, \ldots, K\,. \tag{1}$$

Here $\tilde{x}$ is the input (antecedent) linguistic variable, and A_i are the antecedent linguistic terms (constants). Similarly, $\tilde{y}$ is the output (consequent) linguistic variable and B_i are the consequent linguistic terms. The values of $\tilde{x}$ ($\tilde{y}$) and the linguistic terms A_i (B_i) are fuzzy sets defined in the domains of their respective base variables:[2] $\mathbf{x} \in X \subset \mathbb{R}^p$ and $\mathbf{y} \in Y \subset \mathbb{R}^q$. The membership functions of the antecedent (consequent) fuzzy sets are then the mappings: $\mu(\mathbf{x})\colon X \to [0,1]$, $\mu(\mathbf{y})\colon Y \to [0,1]$. Fuzzy sets A_i define fuzzy regions in the antecedent space, for which the respective consequent propositions hold. The linguistic terms A_i and B_i are usually selected from sets of predefined terms, such as *Small*, *Medium*, etc. By denoting these sets by $\mathcal{A}$ and $\mathcal{B}$ respectively, we have $A_i \in \mathcal{A}$ and $B_i \in \mathcal{B}$. The rule base $\mathcal{R} = \{\mathcal{R}_i | i = 1, 2, \ldots, K\}$ and the sets $\mathcal{A}$ and $\mathcal{B}$ constitute the knowledge base of the linguistic model.

Example 2.1 Consider a simple fuzzy model which qualitatively describes how the heating power of a gas burner depends on the oxygen supply (assuming a constant gas supply). We have a scalar input, the oxygen flow rate (x), and a scalar output, the heating power (y). Define the set of antecedent linguistic terms: $\mathcal{A} = \{\textit{Low}, \textit{OK}, \textit{High}\}$, and the set of consequent linguistic terms: $\mathcal{B} = \{\textit{Low}, \textit{High}\}$. The qualitative relationship between the model input and output can be expressed by the following rules:

$\mathcal{R}_1$: **If** O_2 flow rate **is** *Low* **then** heating power **is** *Low*.
$\mathcal{R}_2$: **If** O_2 flow rate **is** *OK* **then** heating power **is** *High*.
$\mathcal{R}_3$: **If** O_2 flow rate **is** *High* **then** heating power **is** *Low*.

[2] Base variable is the domain variable in which fuzzy sets are defined.

The meaning of the linguistic terms is defined by their membership functions, depicted in Fig. 2. The numerical values along the base variables are selected somewhat arbitrarily. Note that no universal meaning of the linguistic terms can be defined. For this example, it will depend on the type and flow rate of the fuel gas, type of burner, etc. Nevertheless, the qualitative relationship expressed by the rules remains valid. □

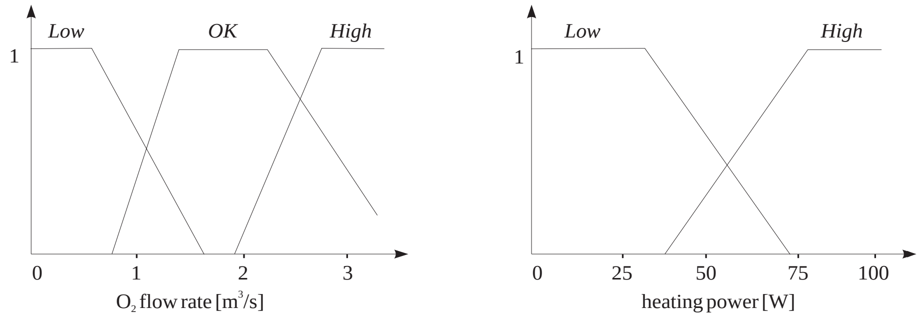


Figure 2. *Membership functions.*

In order to be able to use the linguistic model, we need an algorithm which allows us to compute the output value, given some input value. This algorithm is called the *fuzzy inference* algorithm (or mechanism). For the linguistic model, the inference mechanism can be derived by using fuzzy relational calculus, as shown in the following section.

2.1.1 Relational representation of a linguistic model

Each rule in (1) can be regarded as a fuzzy relation (fuzzy restriction on the simultaneous occurrences of values $\mathbf{x}$ and $\mathbf{y}$): $R_i\colon (X \times Y) \to [0,1]$. This relation can be computed in two basic ways: by using fuzzy conjunctions (Mamdani method) and by using fuzzy implications (fuzzy logic method), see for instance (Driankov, et al., 1993). Fuzzy implications are used when the if-then rule (1) is strictly regarded as an implication $A_i \to B_i$, i.e., "A implies B". In classical logic this means that if A holds, B must hold as well for the implication to be true. Nothing can, however, be said about B when A does not hold, and the relationship also cannot be inverted.

When using a conjunction, $A \wedge B$, the interpretation of the if-then rules is "it is true that A and B simultaneously hold". This relationship is symmetric and can be inverted. For simplicity, in this text we restrict ourselves to the Mamdani (conjunction) method. The relation R is computed by the minimum ($\wedge$) operator:

$$R_i = A_i \times B_i, \quad \text{that is,} \quad \mu_{R_i}(\mathbf{x},\mathbf{y}) = \mu_{A_i}(\mathbf{x}) \wedge \mu_{B_i}(\mathbf{y})\,. \tag{2}$$

Note that the minimum is computed on the Cartesian product space of X and Y, i.e., for all possible pairs of $\mathbf{x}$ and $\mathbf{y}$. The fuzzy relation R representing the entire model (1) is given by the disjunction (union) of the K individual rule's relations R_i:

$$R = \bigcup_{i=1}^{K} R_i, \quad \text{that is,} \quad \mu_R(\mathbf{x},\mathbf{y}) = \max_{1\le i\le K}\left[\mu_{A_i}(\mathbf{x}) \wedge \mu_{B_i}(\mathbf{y})\right]. \tag{3}$$

Now the entire rule base is encoded in the fuzzy relation R and the output of the linguistic model can be computed by the relational max-min composition ($\circ$):

$$\tilde{y} = \tilde{x} \circ R\,. \tag{4}$$

Example 2.2 Let us compute the fuzzy relation for the linguistic model of Example 2.1. First we discretize the input and output domains, for instance: $X = \{0, 1, 2, 3\}$ and $Y = \{0, 25, 50, 75, 100\}$. The (discrete) membership functions are given in Tab. 3 for the antecedent linguistic terms, and in Tab. 4 for the consequent terms.

Table 3. *Antecedent membership functions.*

	domain element			
linguistic term	0	1	2	3
Low	1.0	0.6	0.0	0.0
OK	0.0	0.4	1.0	0.4
High	0.0	0.0	0.1	1.0

Table 4. *Consequent membership functions.*

	domain element				
linguistic term	0	25	50	75	100
Low	1.0	1.0	0.6	0.0	0.0
High	0.0	0.0	0.3	0.9	1.0

The fuzzy relations R_i corresponding to the individual rule, can now be computed by using eq. (2). For rule $\mathcal{R}_1$, we have $R_1 = Low \times Low$, for rule $\mathcal{R}_2$, we obtain $R_2 = OK \times High$, and finally for rule $\mathcal{R}_3$, $R_3 = High \times Low$. The fuzzy relation R, which represents the entire rule base, is the union (element-wise maximum) of the relations R_i:

$$\left.\begin{array}{l} R_1 = \begin{bmatrix} 1.0 & 1.0 & 0.6 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \\ R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0.4 & 0.4 \\ 0 & 0 & 0.3 & 0.9 & 1.0 \\ 0 & 0 & 0.3 & 0.4 & 0.4 \end{bmatrix} \\ \\ R_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0 & 0 \\ 1.0 & 1.0 & 0.6 & 0 & 0 \end{bmatrix} \end{array}\right\} \quad R = \begin{bmatrix} 1.0 & 1.0 & 0.6 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0.4 & 0.4 \\ 0.1 & 0.1 & 0.3 & 0.9 & 1.0 \\ 1.0 & 1.0 & 0.6 & 0.4 & 0.4 \end{bmatrix}. \tag{5}$$

Graphical visualization of these steps is given in Fig. 3. In this figure, the relations are computed on a finer discretization by using the membership functions of Fig. 2. This example can be run under MATLAB by calling the script `ling`. See the file `ling.m` for details of the implementation.

Now consider an input fuzzy set to the model, $A' = [1, 0.6, 0.3, 0]$, which can be denoted as *Somewhat Low* flow rate, as it is close to *Low* but does not equal *Low*. The result of max-min composition is the fuzzy set $B' = [1, 1, 0.6, 0.4, 0.4]$, which gives the expected approximately *Low* heating power. For $A' = [0, 0.2, 1, 0.2]$ (approximately *OK*), we obtain $B' = [0.2, 0.2, 0.3, 0.9, 1]$, i.e., approximately *High* heating power. Verify these results as an exercise. □

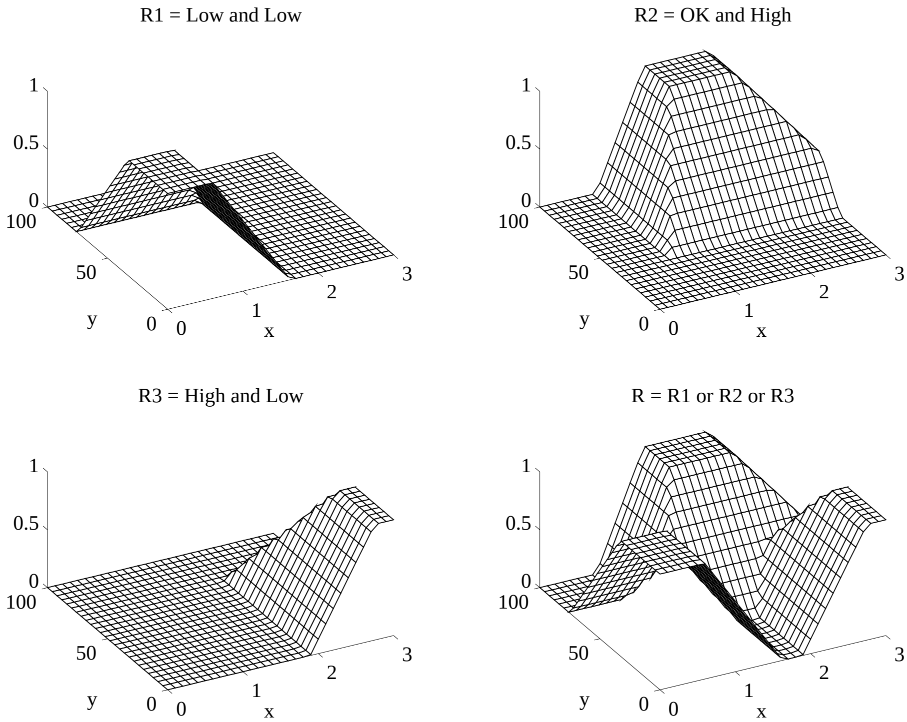


Figure 3. *Fuzzy relations R_1, R_2, R_3 corresponding to the individual rules, and the aggregated relation R corresponding to the entire rule base.*

Because of the relational representation, the linguistic fuzzy model is sometimes called a *fuzzy graph*. Figure 4 shows the fuzzy graph for our example (contours of R, where the shading corresponds to the membership degree). The relational composition (4) can be regarded as a function evaluation on the fuzzy graph, see also Fig. 1.

2.1.2 Max-min (Mamdani) inference

In the previous section, we have seen that a rule base can be represented as a fuzzy relation. The output of a rule-based fuzzy model is then computed by the max-min relational composition. In this section, it will be shown that the relational calculus can be by-passed. This is advantageous, as the discretization of domains and storing of the relation R can be avoided. To show this, suppose an input fuzzy value $\tilde{x} = A'$, for which the output value B' is given by the relational composition:

$$\mu_{B'}(\mathbf{y}) = \max_X [\mu_{A'}(\mathbf{x}) \wedge \mu_R(\mathbf{x}, \mathbf{y})] \,. \tag{6}$$

After substituting for $\mu_R(\mathbf{x}, \mathbf{y})$ from (3), the following expression is obtained:

$$\mu_{B'}(\mathbf{y}) = \max_X \left\{ \mu_{A'}(\mathbf{x}) \wedge \max_{1 \le i \le K} [\mu_{A_i}(\mathbf{x}) \wedge \mu_{B_i}(\mathbf{y})] \right\} \,. \tag{7}$$

Since the max and min operation are taken over different domains, their order can be changed as follows:

$$\mu_{B'}(\mathbf{y}) = \max_{1 \le i \le K} \left\{ \max_X \left[\mu_{A'}(\mathbf{x}) \wedge \mu_{A_i}(\mathbf{x}) \right] \wedge \mu_{B_i}(\mathbf{y}) \right\} \,. \tag{8}$$

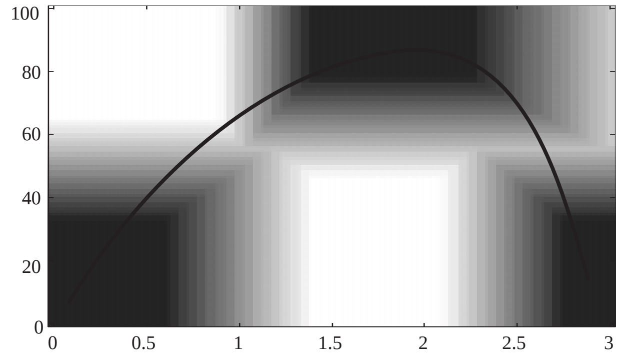


Figure 4. *A fuzzy graph for the linguistic model of Example 2.2. Darker shading corresponds to higher membership degree. The solid line is a possible crisp function representing a similar relationship as the fuzzy model.*

Denote $\beta_i = \max_X [\mu_{A'}(\mathbf{x}) \wedge \mu_{A_i}(\mathbf{x})]$ the *degree of fulfillment* of the ith rule's antecedent. The output fuzzy set of the linguistic model is thus:

$$\mu_{B'}(\mathbf{y}) = \max_{1 \le i \le K} [\beta_i \wedge \mu_{B_i}(\mathbf{y})], \quad \mathbf{y} \in Y\,. \tag{9}$$

The entire algorithm, called the *max-min* or *Mamdani inference*, is summarized in Algorithm 2.1 and visualized in Fig. 5.

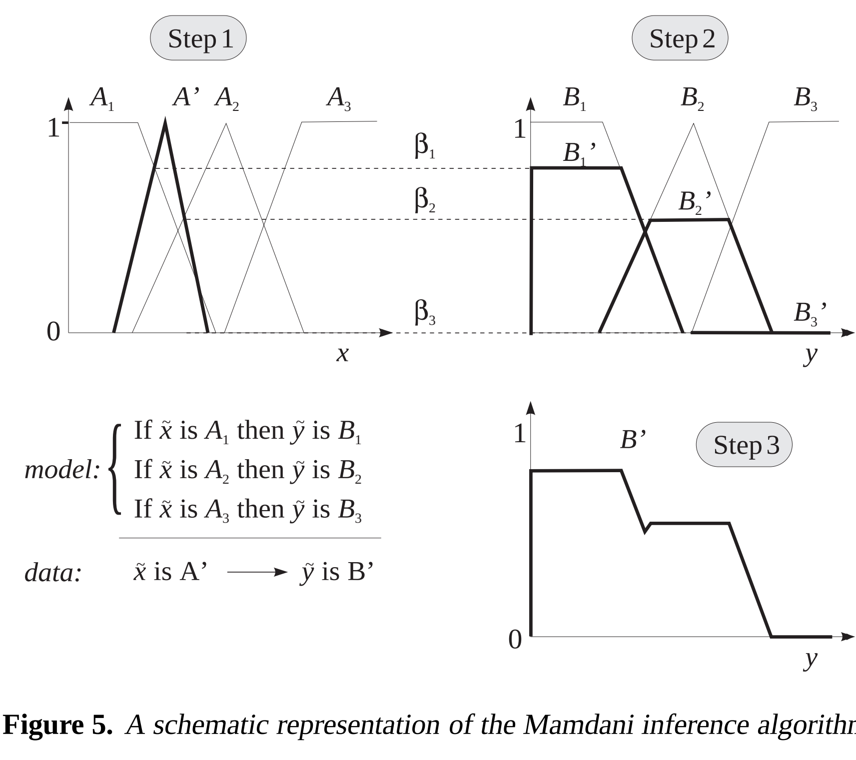


Figure 5. *A schematic representation of the Mamdani inference algorithm.*

Algorithm 2.1 *Mamdani (max-min) inference*

1. Compute the degree of fulfillment by: $\beta_i = \max_X [\mu_{A'}(\mathbf{x}) \wedge \mu_{A_i}(\mathbf{x})]$, $1 \leq i \leq K$. Note that for a singleton fuzzy set ($\mu_{A'}(\mathbf{x}) = 1$ for $\mathbf{x} = \mathbf{x}_0$ and $\mu_{A'}(\mathbf{x}) = 0$ otherwise) the equation for β_i simplifies to $\beta_i = \mu_{A_i}(\mathbf{x}_0)$.
2. Derive the output fuzzy sets B_i': $\mu_{B_i'}(\mathbf{y}) = \beta_i \wedge \mu_{B_i}(\mathbf{y})$, $\mathbf{y} \in Y$, $1 \leq i \leq K$.
3. Aggregate the output fuzzy sets B_i': $\mu_{B'}(\mathbf{y}) = \max_{1 \leq i \leq K} \mu_{B_i'}(\mathbf{y})$, $\mathbf{y} \in Y$.

Example 2.3 Let us take the input fuzzy set $A' = [1, 0.6, 0.3, 0]$ from Example 2.2 and compute the corresponding ouput fuzzy set by the Mamdani inference method. Step 1 yields the following degrees of fulfillment:

$$\beta_1 = \max_X [\mu_{A'}(x) \wedge \mu_{A_1}(x)] = \max([1, 0.6, 0.3, 0] \wedge [1, 0.6, 0, 0]) = 1, \tag{10}$$

$$\beta_2 = \max_X [\mu_{A'}(x) \wedge \mu_{A_2}(x)] = \max([1, 0.6, 0.3, 0] \wedge [0, 0.4, 1, 0.4]) = 0.4, \tag{11}$$

$$\beta_3 = \max_X [\mu_{A'}(x) \wedge \mu_{A_3}(x)] = \max([1, 0.6, 0.3, 0] \wedge [0, 0, 0.1, 1]) = 0.1\,. \tag{12}$$

In step 2, the individual consequent fuzzy sets are computed:

$$B_1' = \beta_1 \wedge B_1 = 1 \wedge [1, 1, 0.6, 0, 0] = [1, 1, 0.6, 0, 0], \tag{13}$$

$$B_2' = \beta_2 \wedge B_2 = 0.4 \wedge [0, 0, 0.3, 0.9, 1] = [0, 0, 0.3, 0.4, 0.4], \tag{14}$$

$$B_3' = \beta_3 \wedge B_3 = 0.1 \wedge [1, 1, 0.6, 0, 0] = [0.1, 0.1, 0.1, 0, 0]\,. \tag{15}$$

Finally, step 3 gives the overall output fuzzy set:

$$B = \max_{1 \leq i \leq K} \mu_{B_i'} = [1, 1, 0.6, 0.4, 0.4],$$

which is identical to the result from Example 2.2. Verify the result for the second input fuzzy set as an exercise. □

From a comparison of the number of operations in examples 2.2 and 2.3, it may seem that the saving with the Mamdani inference method with regard to relational composition is not significant. This is, however, only true for a rough discretization (such as the one used in Example 2.2) and for a small number of inputs (one in this case). Note that the Mamdani inference method does not require any discretization and thus can work with analytically defined membership functions. It also can make use of learning algorithms, as discussed in Section 3.3.3.

2.1.3 Multivariable systems

So far, the linguistic model was presented in a general manner covering both the SISO and MIMO cases. In the MIMO case, all fuzzy sets in the model are defined on vector domains by multivariate membership functions. It is, however, usually, more convenient to write the antecedent and consequent propositions as logical combinations of fuzzy propositions with univariate membership functions. Fuzzy logic operators, such as the conjunction, disjunction and negation (complement), can be used to combine the propositions. Furthermore, a MIMO model can be written as a set of MISO models. Therefore, for the ease of notation, we will write the rules for MISO systems. Most common is the *conjunctive form* of the antecedent, which is given by:

$$\mathcal{R}_i\colon\ \textbf{If } x_1 \textbf{ is } A_{i1} \textbf{ and } x_2 \textbf{ is } A_{i2} \textbf{ and } \ldots \textbf{ and } x_p \textbf{ is } A_{ip} \textbf{ then } y \textbf{ is } B_i, \quad i = 1, 2, \ldots, K\,. \tag{16}$$

Note that the above model is a special case of (1), as the fuzzy set A_i in (1) is obtained as the Cartesian product of fuzzy sets A_{ij}: $A_i = A_{i1} \times A_{i2} \times \cdots \times A_{ip}$. Hence, the degree of fulfillment (step 1 of Algorithm 2.1) is given by:

$$\beta_i = \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \cdots \wedge \mu_{A_{ip}}(x_p), \quad 1 \le i \le K\,. \tag{17}$$

Other conjunction operators, such as the product, can be used. A set of rules in the conjunctive antecedent form divides the input domain into a lattice of fuzzy hyperboxes, parallel with the axes. Each of the hyperboxes is an Cartesian product-space intersection of the corresponding univariate fuzzy sets. This is shown in Fig. 6a. The number of rules in the conjunctive form, needed to cover the entire domain, is given by:

$$K = \Pi_{i=1}^{p} N_i,$$

where p is the dimension of the input space and N_i is the number of linguistic terms of the ith antecedent variable.

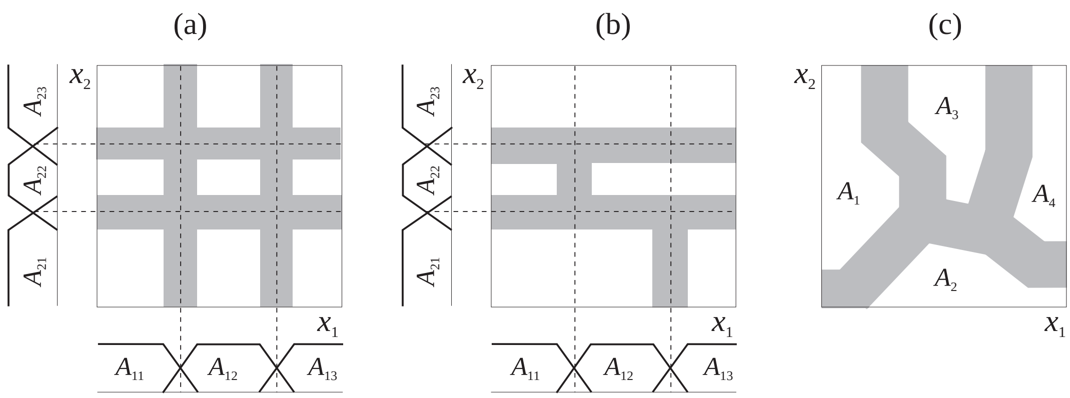


Figure 6. *Different partitions of the antecedent space. Gray areas denote the overlapping regions of the fuzzy sets.*

By combining conjunctions, disjunctions and negations, various partitions of the antecedent space can be obtained, the boundaries are, however, restricted to the rectangular grid defined by the fuzzy sets of the individual variables, see Fig. 6b. As an example consider the rule antecedent covering the lower left corner of the antecedent space in this figure:

$$\textbf{If } x_1 \textbf{ is not } A_{13} \textbf{ and } x_2 \textbf{ is } A_{21} \textbf{ then } \ldots$$

The degree of fulfillment of this rule is computed using the complement and intersection operators:

$$\beta = [1 - \mu_{A_{13}}(x_1)] \wedge \mu_{A_{21}}(x_2)\,. \tag{18}$$

The antecedent form with multivariate membership functions (1) is the most general one, as there is no restriction on the shape of the fuzzy regions. The boundaries between these regions can be arbitrarily curved and opaque to the axes, as depicted in Fig. 6c. Also the number of fuzzy sets needed to cover the antecedent space may be much smaller than in the previous cases. Hence, for complex multivariable systems, this partition may provide the most effective representation. Note that the fuzzy sets A_1 to A_4 in Fig. 6c still can be projected onto x_1 and x_2 to obtain an approximate linguistic interpretation of the regions described.

Another way to reducing the complexity of multivariable fuzzy systems is the decomposition into subsystems with fewer inputs per rule base. The subsystems can be inter-connected in a flat or hierarchical (multi-layer) structure. In such a case, an output of one rule base becomes an input to another rule base, as depicted in Fig. 7. This cascade connection will lead to the reduction of the total number of rules. As an example, suppose five linguistic terms for each input. Using the conjunctive form, each of the two sub-rule bases will have $5^2 = 25$ rules. This is a significant saving compared to a single rule base with three inputs which would have $5^3 = 125$ rules.

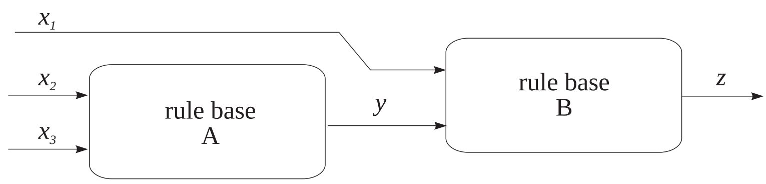


Figure 7. *Cascade connection of two rule bases.*

2.1.4 Defuzzification

In many applications, a crisp output $\mathbf{y}$ is desired. To obtain a crisp value, the output fuzzy set must be *defuzzified*. With the Mamdani inference scheme, the *center of gravity* (COG) defuzzification method is used. This methods computes the y coordinate of the center of gravity of the area under the fuzzy set B':

$$y' = \mathrm{cog}(B') = \frac{\sum_{j=1}^{F} \mu_{B'}(y_j) y_j}{\sum_{j=1}^{F} \mu_{B'}(y_j)}, \tag{19}$$

where F is the number of elements y_j in Y. Continuous domain Y thus must be discretized to be able to compute the center of gravity.

Example 2.4 Consider the output fuzzy set $B' = [0.2, 0.2, 0.3, 0.9, 1]$ from Example 2.2, where the output domain is $Y = [0, 25, 50, 75, 100]$. The defuzzified output obtained by applying formula (19) is:

$$y' = \frac{0.2 \cdot 0 + 0.2 \cdot 25 + 0.3 \cdot 50 + 0.9 \cdot 75 + 1 \cdot 100}{0.2 + 0.2 + 0.3 + 0.9 + 1} = 72.12\,.$$

The heating power of the burner, computed by the fuzzy model, is thus 72.12 W. □

2.1.5 Singleton model

A special case of the linguistic fuzzy model is obtained when the consequent fuzzy sets B_i are singleton fuzzy sets. These sets can be represented simply as real numbers b_i, yielding the following rules:

$$R_i\colon\ \textbf{If } \tilde{\mathbf{x}} \textbf{ is } A_i \textbf{ then } y = b_i, \quad i = 1, 2, \ldots, K\,. \tag{20}$$

This model is called the *singleton model*. A simplified inference/defuzzification method is usually used with this model:

$$y = \frac{\sum_{i=1}^{K} \beta_i \, b_i}{\sum_{i=1}^{K} \beta_i} \, . \tag{21}$$

This defuzzification method is called the *fuzzy mean*. The singleton fuzzy model belongs to a general class of general function approximators, called the basis functions expansion (Friedman, 1991) taking the form:

$$y = \sum_{i=1}^{K} \phi_i(\mathbf{x}) b_i \, . \tag{22}$$

Most structures used in nonlinear system identification, such as artificial neural networks, radial basis function networks, or splines, belong to this class of systems. Connections between these types of models have been investigated (Jang and Sun, 1993; Brown and Harris, 1994). In the singleton model, the basis functions $\phi_i(\mathbf{x})$ are given by the (normalized) degrees of fulfillment of the rule antecedents, and the constants b_i are the consequents. Multilinear interpolation between the rule consequents is obtained if

- the antecedent membership functions are trapezoidal, pairwise overlapping and the membership degrees sum up to one for each domain element,
- the product operator is used to represent the logical **and** connective in the rule antecedents.

The input-output mapping of the singleton model is then piecewise (multi-)linear, as shown in Fig. 8a.

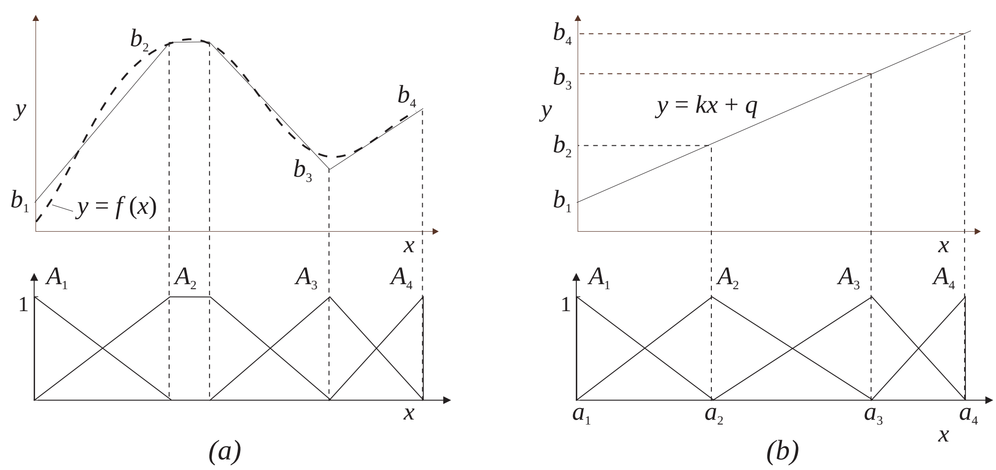


Figure 8. *Singleton model with triangular or trapezoidal membership functions results in a piecewise linear input-output maping (a), of which a linear mapping is a special case (b).*

Clearly, a singleton model can also represent any linear mapping of the form:

$$y = \mathbf{p}^T \mathbf{x} + q = \sum_{i=1}^{p} p_i x_i + q \, . \tag{23}$$

In this case, the antecedent membership functions must be triangular. The consequent singletons can be computed by evaluating the desired mapping (23) for the cores a_{ij} of the antecedent fuzzy sets A_{ij}:

$$b_i = \sum_{j=1}^{p} p_j a_{ij} + q \,. \tag{24}$$

This situation is depicted in Fig. 8b. This property is useful, as the (singleton) fuzzy model can always be initialized such that it mimics a given (perhaps inaccurate) linear model and can later be optimized.

2.2 Takagi–Sugeno model

The linguistic model, introduced in the previous section, describes a given system by means of linguistic if-then rules with fuzzy proposition in the antecedent as well as in the consequent. The Takagi–Sugeno (TS) fuzzy model (Takagi and Sugeno, 1985), on the other hand, uses crisp functions in the consequents. Hence, it can be seen as a combination of linguistic and mathematical regression modeling in the sense that the antecedents describe fuzzy regions in the input space in which consequent functions are valid. The TS rules have the following form:

$$\mathcal{R}_i\text{:}\ \textbf{If } \mathbf{x} \textbf{ is } A_i \textbf{ then } \mathbf{y}_i = \mathbf{f}_i(\mathbf{x}), \qquad i = 1, 2, \ldots, K \,. \tag{25}$$

Contrary to the linguistic model, the input $\mathbf{x}$ is a crisp variable (linguistic inputs are in principle possible, but would require the use of the extension principle (Zadeh, 1975) to compute the fuzzy value of $\mathbf{y}_i$). The functions $\mathbf{f}_i$ are typically of the same structure, only the parameters in each rule are different. Generally, $\mathbf{f}_i$ is a vector-valued function, but for the ease of notation we will consider a scalar f_i in the sequel. A simple and practically useful parameterization is the affine (linear in parameters) form, yielding the rules:

$$\mathcal{R}_i\text{:}\ \textbf{If } \mathbf{x} \textbf{ is } A_i \textbf{ then } y_i = \mathbf{a}_i^T\mathbf{x} + b_i, \qquad i = 1, 2, \ldots, K, \tag{26}$$

where $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset. This model is called an *affine TS model*. Note that if $\mathbf{a}_i = 0$ for each i, the singleton model (20) is obtained.

2.2.1 Inference mechanism

The inference formula of the TS model is a straightforward extension of the singleton model inference (21):

$$y = \frac{\sum_{i=1}^{K} \beta_i y_i}{\sum_{i=1}^{K} \beta_i} = \frac{\sum_{i=1}^{K} \beta_i (\mathbf{a}_i^T\mathbf{x} + b_i)}{\sum_{i=1}^{K} \beta_i} \,. \tag{27}$$

When the antecedent fuzzy sets define distinct but overlapping regions in the antecedent space and the parameters $\mathbf{a}_i$ and b_i correspond to a local linearization of a nonlinear function, the TS model can be regarded as a smoothed piece-wise approximation of that function, see Fig. 9.

2.2.2 TS model as a quasi-linear systems

The affine TS model can be regarded as a quasi-linear system (i.e., a linear system with input-dependent parameters). To see this, denote the normalized degree of fulfillment by

$$\gamma_i(\mathbf{x}) = \beta_i(\mathbf{x}) / \sum_{j=1}^{K} \beta_j(\mathbf{x}) \,. \tag{28}$$

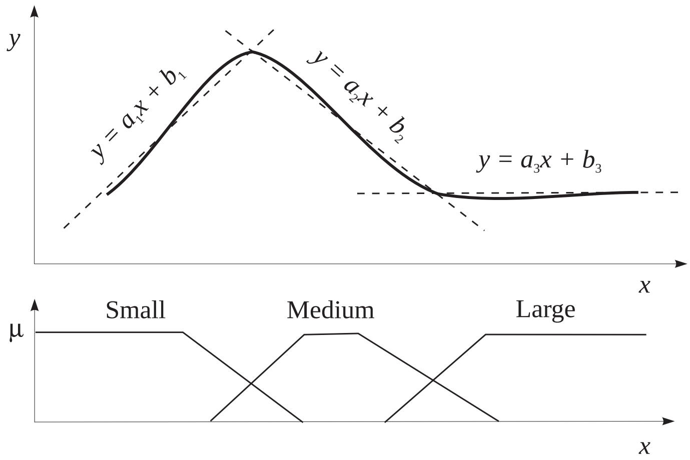


Figure 9. *Takagi–Sugeno fuzzy model as a smoothed piece-wise linear approximation of a nonlinear function.*

Here we write $\beta_i(\mathbf{x})$ explicitely as a function $\mathbf{x}$ to stress that the TS model is a quasi-linear model of the following form:

$$y = \left(\sum_{i=1}^{K} \gamma_i(\mathbf{x}) \mathbf{a}_i^T \right) \mathbf{x} + \sum_{i=1}^{K} \gamma_i(\mathbf{x}) b_i = \mathbf{a}^T(\mathbf{x})\mathbf{x} + b(\mathbf{x}) \, . \tag{29}$$

The 'parameters' $\mathbf{a}(\mathbf{x})$, $b(\mathbf{x})$ are convex linear combinations of the consequent parameters $\mathbf{a}_i$ and b_i, i.e.:

$$\mathbf{a}(\mathbf{x}) = \sum_{i=1}^{K} \gamma_i(\mathbf{x}) \mathbf{a}_i, \quad b(\mathbf{x}) = \sum_{i=1}^{K} \gamma_i(\mathbf{x}) b_i \, . \tag{30}$$

In this sense, a TS model can be regarded as a mapping from the antecedent (input) space to a convex region (polytope) in the space of the parameters of a quasi-linear system, as schematically depicted in Fig. 10.

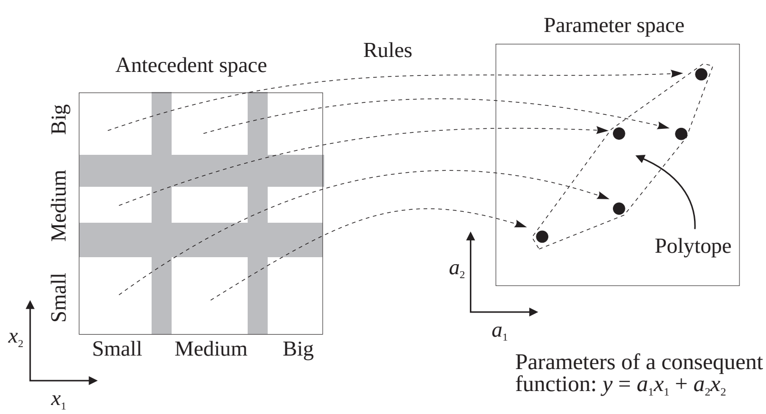


Figure 10. *A TS model with affine consequents can be regarded as a mapping from the antecedent space to the space of the consequent parameters.*

This property facilitates the analysis of TS models in a framework similar to that of linear systems. Methods have been developed to design controllers with desired closed loop characteristics (Filev, 1996) and to analyze their stability (Tanaka and Sugeno, 1992; Zhao, 1995; Tanaka, et al., 1996).

2.3 Modeling dynamic systems

Before discussing dynamic fuzzy models, let us recall that time-invariant dynamic systems are in general modelled by static functions, by using the concept of the system's *state*. Given the state of a system and given its input, we can determine what the next state will be. In the discrete-time setting we can write

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \tag{31}$$

where $\mathbf{x}(k)$ and $\mathbf{u}(k)$ are the state and the input at time k, respectively, and $\mathbf{f}$ is a static function, called the *state-transition function*. Fuzzy models of different types can be used to approximate the state-transition function. As the state of a process is often not measured, *input-output* modeling is usually applied. The most common is the NARX (Nonlinear AutoRegessive with eXogenous input) model:

$$y(k+1) = f(y(k), y(k-1), \ldots, y(k-n_y+1), u(k), u(k-1), \ldots, u(k-n_u+1)) \,. \tag{32}$$

Here $y(k), \ldots, y(k-n_y+1)$ and $u(k), \ldots, u(k-n_u+1)$ denote the past model outputs and inputs respectively and n_y, n_u are integers related to the model order (usually selected by the user). For example, a linguistic fuzzy model of a dynamic system may consist of rules of the following form:

$$\begin{aligned} R_i : \textbf{If } & y(k) \text{ is } A_{i1} \textbf{ and } y(k-1) \text{ is } A_{i2} \textbf{ and}, \ldots y(k-n+1) \text{ is } A_{in} \textbf{ and } u(k) \text{ is } B_{i1} \\ & \textbf{and } u(k-1) \text{ is } B_{i2} \textbf{ and}, \ldots, u(k-m+1) \text{ is } B_{im} \textbf{ then } y(k+1) \text{ is } C_i \,. \end{aligned} \tag{33}$$

In this sense, we can say that the dynamic behavior is taken care of by external dynamic filters added to the fuzzy system Fig. 11. In (33), the input dynamic filter is a simple generator of the lagged inputs and outputs, and no output filter is used.

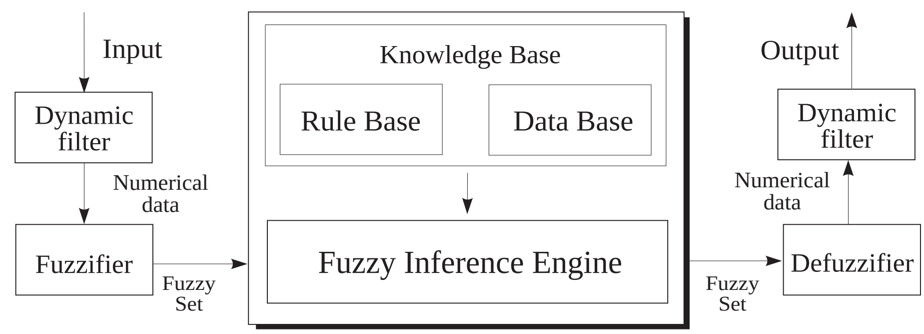


Figure 11. *A generic fuzzy system with fuzzification and defuzzification units and external dynamic filters.*

Since the fuzzy models can approximate any smooth function to any degree of accuracy (Wang, 1992), models of type (33) can approximate any observable and controllable modes of a large class of discrete-time nonlinear systems (Leonaritis and Billings, 1985).

3 Building Fuzzy Models

Two common sources of information for building fuzzy models are the prior knowledge and data (process measurements). The prior knowledge can be of a rather approximate nature (qualitative knowledge, heuristics), which usually originates from "experts", i.e., process designers, operators, etc. In this sense, fuzzy models can be regarded as simple *fuzzy expert systems* (Zimmermann, 1987).

For many processes, data are available as records of the process operation or special identification experiments can be designed to obtain the relevant data. Building fuzzy models from data involves methods based on fuzzy logic and approximate reasoning, but also ideas originating from the field of neural networks, data analysis and conventional systems identification. The acquisition or tuning of fuzzy models by means of data is usually termed *fuzzy identification*.

Two main approaches to the integration of knowledge and data in a fuzzy model can be distinguished:

1. The expert knowledge expressed in a verbal form is translated into a collection of if–then rules. In this way, a certain model structure is created. Parameters in this structure (membership functions, consequent singletons or parameters) can be fine-tuned using input-output data. The particular tuning algorithms exploit the fact that at the computational level, a fuzzy model can be seen as a layered structure (network), similar to artificial neural networks, to which standard learning algorithms can be applied. This approach is usually termed *neuro-fuzzy modeling* (Jang, 1993; Jang and Sun, 1993; Pedrycz, 1995).

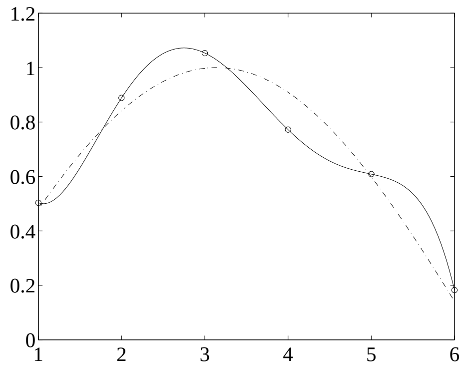
2. No prior knowledge about the system under study is initially used to formulate the rules, and a fuzzy model is constructed from data. It is expected that the extracted rules and membership functions can provide an a posteriori interpretation of the system's behavior. An expert can confront this information with his own knowledge, can modify the rules, or supply new ones, and can design additional experiments in order to obtain more informative data.

These techniques, of course, can be combined, depending on the particular application. In the sequel, we describe the main steps and choices in the knowledge-based construction of fuzzy models, and the main techniques to extract or fine-tune fuzzy models by means of data.

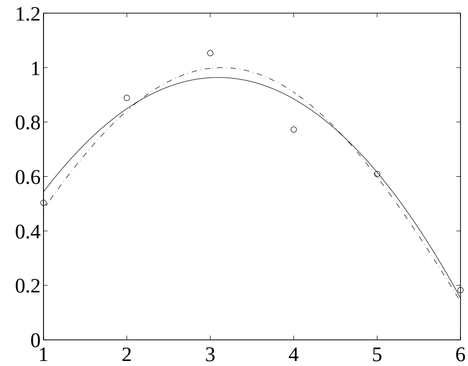
3.1 Structure and parameters

With regard to the design of fuzzy (and also other) models, two basic items are distinguished: the structure and the parameters of the model. The structure determines the flexibility of the model in approximation (unknown) mappings. The parameters are then tuned (estimated) to fit the data at hand. A model with a rich structure is able to approximate more complicated functions, but, at the same time, has worse *generalization* properties. Good generalization means that a model fitted to one data set will also perform well on another data set from the same process.

Example 3.1 A well-known example of a general function approximator is a polynomial function: $y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$. In this case, the structure is the order n of the polynomial and the parameters are the constants a_0 to a_n. Higher-order polynomials will be able to approximate more complicated functions, but will have worse generalization properties. For instance, a 5-th order polynomial has six parameters and therefore will perfectly fit 6 data points (a unique analytical solution). However, if the data is corrupted by noise, a large error may occur for new data, as depicted in Fig. 12a. A less complex model, such as a second-order polynomial, will do much better in this case, see Fig. 12b. See the function `polynom.m`. □



(a) *Fifth-order polynomial.*



(b) *Second-order polynomial.*

Figure 12. *Approximation of a sinusoidal function (dashed-dotted line) by two models of a different complexity (solid line).*

In fuzzy models, structure selection involves the following choices:

- *Input and output variables.* With complex systems, it is not always clear which variables should be used as inputs to the model. In the case of dynamic systems, one also must estimate the order of the system. For the input-output NARX model (32) this means to define the number of input and output lags n_y and n_u, respectively. Prior knowledge, insight in the process behavior and the purpose of modeling are the typical sources of information for this choice. Sometimes, automatic data-driven selection can be used to compare different choices in terms of some performance criteria.
- *Structure of the rules*. This choice involves the model type (linguistic, singleton, Takagi-Sugeno) and the antecedent form (refer to Section 2.1.3). Important aspects are the purpose of modeling and the type available knowledge.
- *Number and type of membership functions for each variable.* This choice determines the level of detail (granularity) of the model. Again, the purpose of modeling and the detail of available knowledge, will influence this choice. Automated, data-driven methods can be used to add or remove membership functions from the model.
- *Type of the inference mechanism, connective operators, defuzzification method.* These choices are restricted by the type of fuzzy model (Mamdani, TS). Within these restrictions, however, some freedom remains, e.g., as to the choice of the conjunction operators, etc. To facilitate data-driven optimization of fuzzy models (learning), differentiable operators (product, sum) are often preferred to the standard min and max operators.

After the structure is fixed, the performance of a fuzzy model can be fine-tuned by adjusting its parameters. Tunable parameters of linguistic models are the parameters of antecedent and consequent membership functions (determine their shape and position) and the rules (determine the mapping between the antecedent and consequent fuzzy regions). Takagi-Sugeno models have parameters in antecedent membership functions and in the consequent functions ($\mathbf{a}$ and b for the affine TS model).

3.2 Knowledge-based design

To design a (linguistic) fuzzy model based on available expert knowledge, the following steps can be followed:

1. Select the input and output variables, the structure of the rules and the inference and defuzzification methods.
2. Decide on the number of linguistic terms for each variable and define the corresponding membership functions.
3. Formulate the available knowledge in terms of fuzzy if-then rules.
4. Validate the model (for instance by using data). If the model does not meet the expected performance, iterate on the above design steps.

It should be noted that the success of this method heavily depends on the problem at hand, and the extent and quality of the available knowledge. For some problems, the knowledge-based design may lead fast to useful models, while for others it may be a very time-consuming and inefficient procedure (especially manual fine-tuning of the model parameters). Therefore, it is useful to combine the knowledge based design with a data-driven tuning of the model parameters. The following sections review several methods for the adjustment of fuzzy model parameters by means of data.

3.3 Data-driven acquisition/tuning of fuzzy models

In this section, we assume that a set of N input-output data pairs $\{(\mathbf{x}_i, y_i) | i = 1, 2, \ldots, N\}$ is available. Recall that $\mathbf{x}_i \in \mathbb{R}^p$ are input vectors and y_i are output scalars. Denote $\mathbf{X} \in \mathbb{R}^{N\times p}$ a matrix having the vectors $\mathbf{x}_k^T$ in its rows, and $\mathbf{y} \in \mathbb{R}^N$ a vector containing the outputs y_k:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T, \quad \mathbf{y} = [y_1, \ldots, y_N]^T. \tag{34}$$

3.3.1 Least-squares estimation of consequents

Note that the defuzzification formulas of the singleton and TS models, equations (21) and (27), respectively, are linear in the consequent parameters, $\mathbf{a}_i$, b_i. Hence, these parameters can be estimated from the available data by least-squares techniques. Denote $\mathbf{\Gamma}_i \in \mathbb{R}^{N\times N}$ the diagonal matrix having the normalized membership degree $\gamma_i(\mathbf{x}_k)$ of (28) as its kth diagonal element. By appending a unitary column to $\mathbf{X}$, the extended matrix $\mathbf{X}_e = [\mathbf{X}, \mathbf{1}]$ is created. Further, denote $\mathbf{X}'$ the matrix in $\mathbb{R}^{N\times KN}$ composed of the products of matrices $\mathbf{\Gamma}_i$ and $\mathbf{X}_e$

$$\mathbf{X}' = [\mathbf{\Gamma}_1\mathbf{X}_e,\ \mathbf{\Gamma}_2\mathbf{X}_e,\ \ldots,\ \mathbf{\Gamma}_K\mathbf{X}_e]\ . \tag{35}$$

The consequent parameters $\mathbf{a}_i$ and b_i are lumped into a single parameter vector $\theta \in \mathbb{R}^{K(p+1)}$:

$$\theta = \left[\mathbf{a}_1^T, b_1,\ \mathbf{a}_2^T, b_2,\ \ldots,\ \mathbf{a}_K^T, b_K\right]^T . \tag{36}$$

Given the data $\mathbf{X}$, $\mathbf{y}$, eq. (27) now can be written in a matrix form, $\mathbf{y} = \mathbf{X}'\theta + \epsilon$. From linear algebra (Strang, 1976) we know that this set of equations can be solved for the parameter θ by:

$$\theta = \left[(\mathbf{X}')^T\mathbf{X}'\right]^{-1}(\mathbf{X}')^T\mathbf{y}\,. \tag{37}$$

This is an optimal least-squares solution which gives the minimal prediction error, and as such is suitable for prediction models. At the same time, however, it may bias the estimates of the consequent parameters as parameters of local models. If an accurate estimate of local model parameters is desired, a weighted least-squares approach applied per rule may be used:

$$[\mathbf{a}_i^T, b_i]^T = \left[\mathbf{X}_e^T\boldsymbol{\Gamma}_i\mathbf{X}_e\right]^{-1}\mathbf{X}_e^T\boldsymbol{\Gamma}_i\mathbf{y}\,. \tag{38}$$

In this case, the consequrent parameters of individual rules are estimated independently of each other, and therefore are not "biased" by the interactions of the rules. By omitting $\mathbf{a}_i$ for all $1 \le i \le K$, equations (37) and (38) directly apply to the singleton model (20).

3.3.2 Template-based modeling

With this approach, the domains of the antecedent variables are simply partitioned into a specified number of equally spaced and shaped membership functions. The rule base is then established to cover all the combinations of the antecedent terms. The consequent parameters are estimated by the least-squares method.

Example 3.2 Consider a nonlinear dynamic system described by a first-order difference equation:

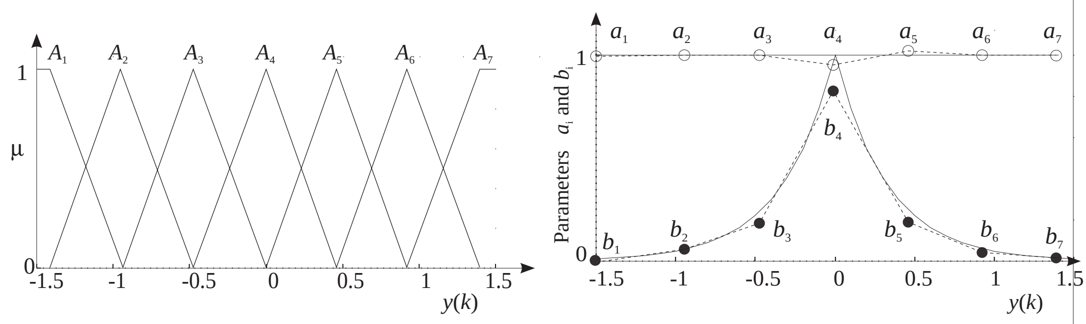
$$y(k+1) = y(k) + u(k)e^{-3|y(k)|}\,. \tag{39}$$

We use a stepwise inputs signal to generate with this equation a set of 300 input–output data pairs (see Fig. 14a). Suppose that it is known that the system is first order and that the nonlinearity of the system is only caused by y, the following TS rule structure can be chosen:

$$\textbf{If } y(k) \textbf{ is } A_i \textbf{ then } y(k+1) = a_i y(k) + b_i u(k), \tag{40}$$

Assuming that no further prior knowledge is available, seven equally spaced triangular membership functions, A_1 to A_7, are defined in the domain of $y(k)$, as shown in Fig. 13a.

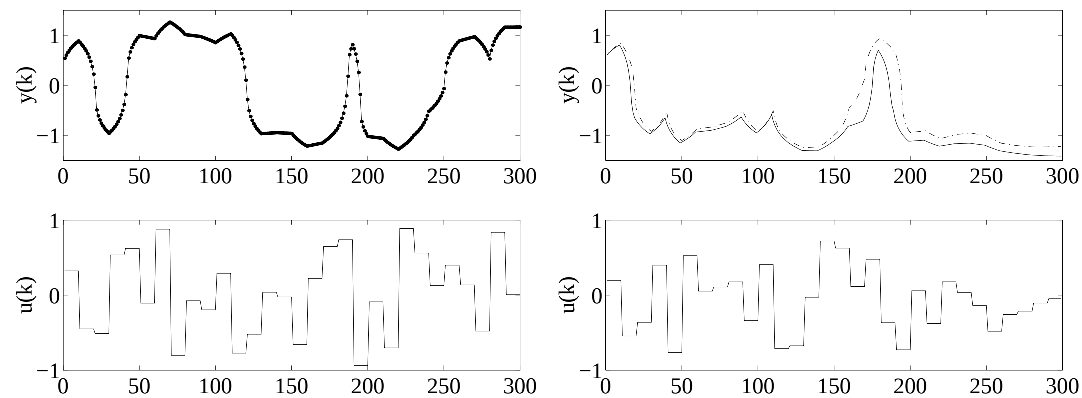
The consequent parameters were estimated by the least-squares method as described in Section 3.3.1. Figure 13b gives a plot of the parameters a_i, b_i against the cores of the antecedent fuzzy sets A_i. Also plotted is the linear interpolation between the parameters (dashed line) and the true system nonlinearity (solid line). The interpolation between a_i and b_i is linear, since the membership functions are piece-wise linear (triangular). One can observe that the dependence of the consequent parameters on the antecedent variable approximates quite accurately the system's nonlinearity, which gives the model a certain transparency. The values of the parameters $\mathbf{a}^T = [1.00, 1.00, 1.00, 0.97, 1.01, 1.00, 1.00]$ and $\mathbf{b}^T = [0.01, 0.05, 0.20, 0.81, 0.20, 0.05, 0.01]^T$ indicate the strong input nonlinearity and the linear dynamics as in (39). Validation of the model in simulation using a different data set is given in Fig. 14b. This example is implemented in the MATLAB function `phdemo.m`. □



(a) Membership functions.

(b) Estimated consequent parameters.

Figure 13. *(a) Equidistant triangular membership functions designed for the output $y(k)$; (b) comparison of the true system nonlinearity (solid line) and its approximation in terms of the estimated consequent parameters (dashed line).*



(a) Identification data.

(b) Validation.

Figure 14. *Identification data set (a), and performance of the model on a validation data set (b). Solid line: process, dashed-dotted line: model.*

The transparent local structure of the TS model facilitates the combination of local models obtained by parameter estimation and linearization of known mechanistic (white-box) models. If measurements are available only in certain regions of the process' operating domain, parameters for the remaining regions can be obtained by linearizing a (locally valid) mechanistic model of the process. Suppose that this model is given by $y = f(\mathbf{x})$. Linearization around the center $\mathbf{c}_i$ of the ith rule's antecedent membership function yields the following parameters of the affine TS model (26):

$$\mathbf{a}_i = \left.\frac{df}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{c}_i}, \quad b_i = f(\mathbf{c}_i)\,. \tag{41}$$

A drawback of the template-based approach is that the number of rules in the model may grow very fast. If no knowledge is available as to which variables cause the nonlinearity of the system, all the antecedent variables are usually partitioned uniformly, which leads to an exponential increase of the number of rules.

The complexity of the system's behavior is typically not uniform, which means that certain regions can be well approximated by a single model, while other regions require rather fine partitioning. In order to obtain an efficient representation with as few rules as possible, the membership functions must be placed such that they capture the non-uniform behavior of the system. This often requires that system measurements are also used to form the membership functions, as discussed in the following sections.

3.3.3 Neuro-fuzzy modeling

In Section 3.3.1 we have seen that parameters that are linearly related to the output can be (optimally) estimated by least-squares methods. In order to optimize also the parameters which are related to the output in a nonlinear way, training algorithms known from the area of neural networks can be employed. These techniques exploit the fact that, at the computational level, a fuzzy model can be seen as a layered structure (network), similar to artificial neural networks. Hence, this approach is usually referred to as neuro-fuzzy modeling (Jang, 1993; Jang and Sun, 1993; Brown and Harris, 1994). Figure 15 gives an example of a singleton fuzzy model with two rules represented as a network. The rules are:

If x_1 **is** A_{11} **and** x_2 **is** A_{21} **then** $y = b_1$.
If x_1 **is** A_{12} **and** x_2 **is** A_{22} **then** $y = b_2$.

The nodes in the first layer compute the membership degree of the inputs in the antecedent fuzzy sets. The product nodes Π in the second layer represent the antecedent conjunction operator. The normalization node N and the summation node Σ realize the fuzzy-mean operator (27).

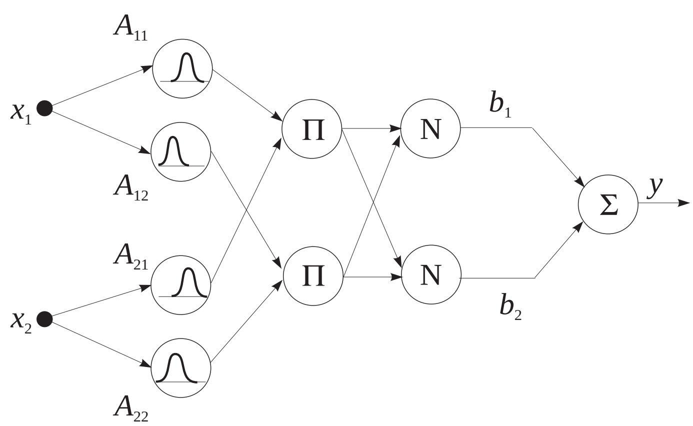


Figure 15. *An example of a singleton fuzzy model with two rules represented as a (neuro-fuzzy) network.*

By using smooth antecedent membership functions, such as Gaussian functions:

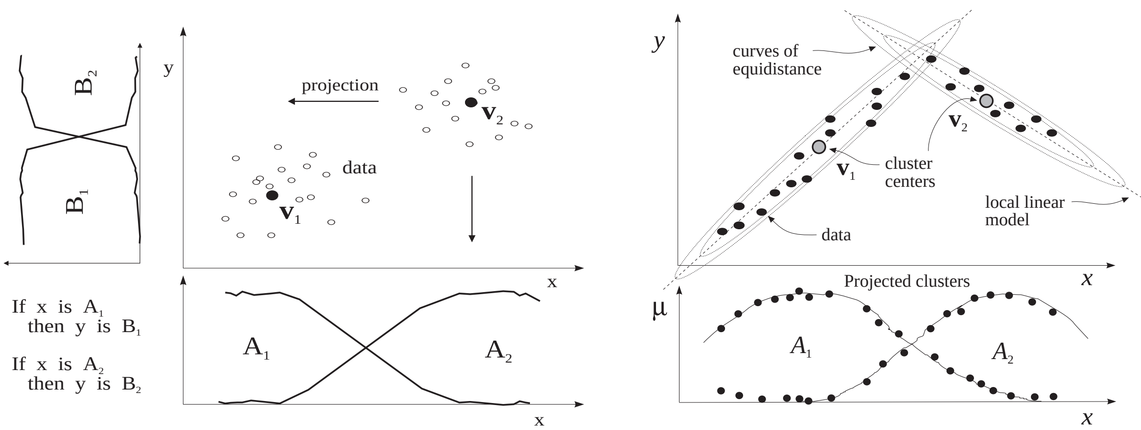
$$\mu_{A_{ij}}(x_j; c_{ij}, \sigma_{ij}) = \exp\left(-(\frac{x_j - c_{ij}}{2\sigma_{ij}})^2\right), \tag{42}$$

the c_{ij} and σ_{ij} parameters can be adjusted by gradient-descent learning algorithms, such as back-propagation (Wang, 1992). This allows for a fine-tuning of the fuzzy model to the available data in order to optimize its prediction accuracy.

3.3.4 Fuzzy clustering

Identification methods based on fuzzy clustering originate from data analysis and pattern recognition, where the concept of graded membership is used to represent the degree to which a given object, represented as a vector of features, is similar to some prototypical object. The degree of similarity can be calculated using a suitable distance measure. Based on the similarity, feature vectors can be clustered such that the vectors within a cluster are as similar (close) as possible, and vectors from different clusters are as dissimilar as possible.

This idea of *fuzzy clustering* is depicted in Fig. 16a, where the data is clustered into two groups with prototypes $\mathbf{v}_1$ and $\mathbf{v}_2$, using the Euclidean distance measure. The partitioning of the data is expressed in the *fuzzy partition matrix* whose elements μ_{ij} are degrees of membership of the data points $[x_i, y_i]$ in a fuzzy cluster with prototypes v_j.



(a) Rule-based interpretation of fuzzy clusters. *(b) Hyperellipsoidal fuzzy clusters.*

Figure 16. *Identification by fuzzy clustering.*

Fuzzy if-then rules can be extracted by projecting the clusters onto the axes. Figure 16a shows a data set with two apparent clusters and two associated fuzzy rules. The concept of similarity of data to a given prototype leaves enough space for the choice of an appropriate distance measure and of the character of the prototype itself. For example, the prototypes can be defined as linear subspaces (Bezdek, 1981), or the clusters can be ellipsoids with adaptively determined shape (Gustafson and Kessel, 1979), see Fig. 16b. From such clusters, the antecedent membership functions and the consequent parameters of the Takagi–Sugeno model can be extracted (Babuška and Verbruggen, 1995):

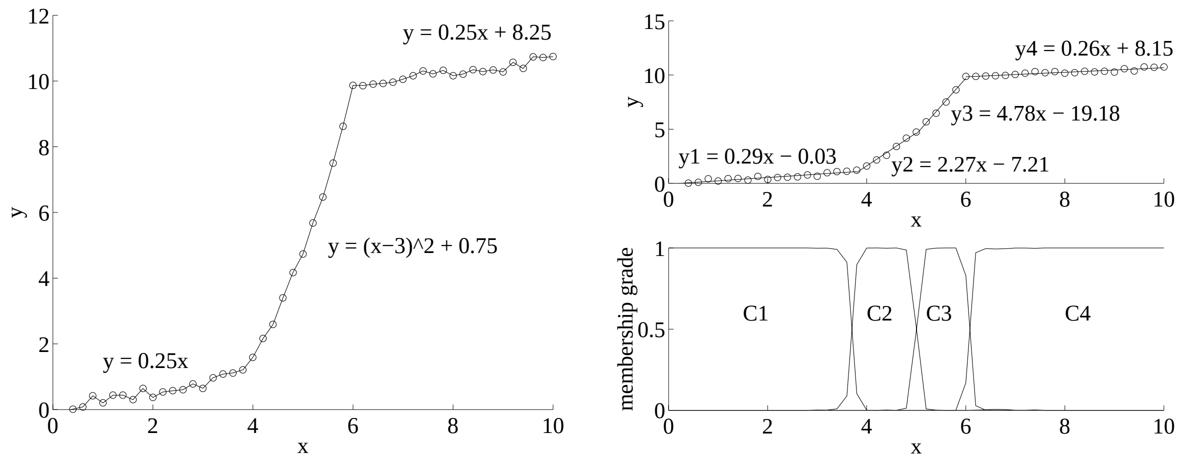
$$\textbf{If } x \textbf{ is } A_1 \textbf{ then } y = a_1 x + b_1 ,$$
$$\textbf{If } x \textbf{ is } A_2 \textbf{ then } y = a_2 x + b_2 .$$

Each obtained cluster is represented by one rule in the Takagi–Sugeno model. The membership functions for fuzzy sets A_1 and A_2 are generated by point-wise projection of the partition matrix onto the antecedent variables. These point-wise defined fuzzy sets are then approximated by a suitable parametric function. The consequent parameters for each rule are obtained as least-squares estimates (37) or (38).

Example 3.3 Consider a nonlinear function $y = f(x)$ defined piece-wise by:

$$\begin{aligned} y &= 0.25x, && \text{for } x \le 3 \\ y &= (x-3)^2 + 0.75, && \text{for } 3 < x \le 6 \\ y &= 0.25x + 8.25, && \text{for } x > 6 \end{aligned} \tag{43}$$

Figure 17a shows a plot of this function evaluated in 50 samples uniformly distributed over $x \in [0, 10]$. Zero-mean, uniformly distributed noise with amplitude 0.1 was added to y.



(a) A nonlinear function (43).

(b) Cluster prototypes (top) and the corresponding fuzzy sets (bottom).

Figure 17. Approximation of a static nonlinear function using a Sugeno–Takagi fuzzy model.

The data $\{(x_i, y_i) | i = 1, 2, \ldots, 50\}$ was clustered into four hyperellipsoidal clusters. The upper plot of Fig. 17b shows the local linear models obtained through clustering, the bottom plot shows the corresponding fuzzy partition. In terms of the TS rules, the fuzzy model is expressed as:

$\mathcal{R}_1$: **If** x is in C_1 **then** $y = 0.29x - 0.03$ $\mathcal{R}_3$: **If** x is in C_3 **then** $y = 4.78x - 19.18$
$\mathcal{R}_2$: **If** x is in C_2 **then** $y = 2.27x - 7.21$ $\mathcal{R}_4$: **If** x is in C_4 **then** $y = 0.26x + 8.15$

Note that the consequents of $\mathcal{R}_1$ and $\mathcal{R}_4$ correspond almost exactly to the first and third equation (43). Consequents of $\mathcal{R}_2$ and $\mathcal{R}_3$ are approximate tangents to the parabola defined by the second equation of (43) in the respective cluster centers. See the function `clustdem.m`. □

4 Overview of Applications

Fuzzy modeling and identification methodologies have been successfully used in a number of real-world applications. The Takagi–Sugeno model has often been employed in the modeling and identification of nonlinear technical processes from data. Examples are the modeling of a multilayer incinerator (Sugeno and Kang, 1986), a converter in a steel-making process (Takagi and Sugeno, 1985), or a glass-melting furnace (Zhao, et al., 1994).

Biotechnology and ecology are typical examples of areas where conventional modeling techniques do not give satisfactory results. Fuzzy modeling has been used in a number of applications, such as Penicillin–G conversion (Babuška, et al., 1996), prediction of river water flow (Sugeno and Tanaka, 1991), enzymatic soil removal in washing processes (Kaymak, 1994), or modeling of algae growth in lakes (Setnes, et al., 1997).

Fuzzy models can be used in the design of automatic controllers, for instance in train operation (Terano, et al., 1994), combustion control (Sugeno and Kang, 1986), or pressure control (Babuška, et al., 1996). Fuzzy models can also serve as decision support systems to assist operators (den Hartog, et al., 1997), or can be used to clone the operators based on traces of their behavior (Sugeno and Yasukawa, 1993).

5 Summary and Concluding Remarks

Fuzzy modeling is a framework in which different modeling and identification methods are combined, providing, on the one hand, a transparent interface with the designer or the operator and, on the other hand, a flexible tool for nonlinear system modeling and control, comparable with other nonlinear black-box techniques. The rule-based character of fuzzy models allows for a model interpretation in a way that is similar to the one humans use. Conventional methods for statistical validation based on numerical data can be complemented by the human expertise, that often involves heuristic knowledge and intuition.

Fuzzy models can be used for various aims: analysis, design, control, monitoring, supervision, etc. Approaches have been presented to switch from one model representation to another one, which is more apt for a certain interpretation, allowing a multifaceted use of a model based on one set of data.

Rather than as a fully automated identification technique, fuzzy modeling should be seen as an interactive method, facilitating the active participation of the user in a computer-assisted modeling session. However, this is also the case in using other, more established methods. Modeling of complex systems will always remain an interactive approach. Intuition and experience of the team will always play a major role in this process. Special software tools will also need to be developed for this purpose.

Exercises

1. Explain the terms “white-box” (mechanistic) modeling and “black-box” modeling.
2. What is a fuzzy system? Give two examples of fuzzy systems.
3. Give an example of a linguistic if-then rule. What are the linguistic variables and linguistic terms (constants) in your example?
4. What do you understand under “relational representation of a rule base”? How is the fuzzy relation constructed? How is this relation used to derive an output fuzzy set, given the input fuzzy set?
5. Carry out the relational composition of A' with R from Example 2.2. Implement it in a MATLAB function.

6. What is fuzzy inference? Give the formulas for the Mamdani (max-min) inference algorithm. Give the graphical representation for an example with two antecedent variables and one consequent variable, using the conjunctive form of rules.

7. Given are two rules: " **If** $\tilde{x}$ **is** A_1 **then** $\tilde{y}$ **is** B_1" and " **If** $\tilde{x}$ **is** A_2 **then** $\tilde{y}$ **is** B_2", where $A_1 \cap A_2 \neq \emptyset$. Suppose a fuzzy input $\tilde{x} = A_1$. Is the output $\tilde{y} = B_1$? Explain your answer.

8. Give the formula for the center-of-gravity defuzzification.

9. Explain what the singleton fuzzy model is. Give the inference/defuzzification formula for this model.

10. Define the affine Takagi–Sugeno fuzzy model.

11. Give an example of a SISO (single-input, single-output) first-order NARX (Nonlinear AutoRegessive with eXogenous input) model with linguistic fuzzy rules.

12. What are the main steps of knowledge-based design of fuzzy models?

13. Which mathematical method can be used to estimate optimal consequent parameters of a Takagi–Sugeno fuzzy model?

14. What is a neuro-fuzzy network? Give a simple example.

A List of Symbols

A, B, ...	fuzzy sets
K	number of rules in a rule base
R	fuzzy relation
$\mathcal{R}$	fuzzy if-then rule
$\mathbf{X}$	matrix containing input data (regressors)
$\mathbf{0}$	matrix of appropriate dimensions with all entries equal to zero
$\mathbf{1}$	matrix of appropriate dimensions with all entries equal to one
$\mathbf{a}$, b	consequent parameters in a TS model
q	number of outputs of a (static) fuzzy model
p	number of inputs of a (static) fuzzy model
$\tilde{x}, \tilde{y}, \ldots$	linguistic variables (have fuzzy sets as their values)
$\mathbf{x}$	input vector
$\mathbf{y}$	output vector
$u(k)$, $y(k)$	input and output of a dynamic system at time k, respectively
$\mathbf{y}$	vector containing regressand data
β	degree of fulfillment of a rule
γ	normalized degree of fulfillment
μ, $\mu(\cdot)$	membership degree, membership function
X, Y	domains (universes) of variables x and y
$\mathbb{R}$	set of real numbers

Operators:

$\mathbf{X}^T$	transpose of matrix $\mathbf{X}$
$\mathrm{cog}(A)$	center of gravity defuzzification of fuzzy set A
$\mathrm{core}(A)$	core of fuzzy set A
$\mathrm{supp}(A)$	support of fuzzy set A
$\cap$	(fuzzy) set intersection (conjunction)
$\cup$	(fuzzy) set union (disjunction)
$\wedge$	minimum, (fuzzy) conjunction, logical AND
$\vee$	maximum, (fuzzy) disjunction, logical OR
$\bar{A}$	complement (negation) of A
$\circ$	max-min composition

Abbreviations:

AI	artificial intelligence
COG	center of gravity
MIMO	multiple–input, multiple–output
MISO	multiple–input, single–output
(N)ARX	(nonlinear) autoregressive with exogenous inputs
SISO	single–input, single–output

References

Babuška, R., H.A.B. te Braake, A.J. Krijgsman and H.B. Verbruggen (1996). Comparison of intelligent control schemes for real-time pressure control. *Control Engineering Practice 4*(11), 1585–1592.

Babuška, R., H.J.L. van Can and H.B. Verbruggen (1996). Fuzzy modeling of enzymatic Penicillin–G conversion. In *Preprints 13th IFAC World Congress*, Volume N, San Francisco, USA, pp. 479–484.

Babuška, R. and H.B. Verbruggen (1995). Identification of composite linear models via fuzzy clustering. In *Proceedings European Control Conference*, Rome, Italy, pp. 1207–1212.

Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function*. Plenum Press, New York.

Brown, M. and C. Harris (1994). *Neurofuzzy Adaptive Modelling and Control*. New York: Prentice Hall.

Buchanan, B.G. and E.H. Shortliffe (1984). *Rule-Based Expert Systems*. Readings (MA), USA: Addison–Wesley.

Driankov, D., H. Hellendoorn and M. Reinfrank (1993). *An Introduction to Fuzzy Control*. Springer, Berlin.

Dubois, D. and H. Prade (1980). *Fuzzy sets and systems: theory and applications*, Volume 144 of *Mathematics in science and engineering*. Academic Press.

Filev, D.P. (1996). Model based fuzzy control. In *Proceedings Fourth European Congress on Intelligent Techniques and Soft Computing EUFIT'96*, Aachen, Germany.

Friedman, J.H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics 19*(1), 1–141.

Gustafson, D.E. and W.C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, San Diego, CA, USA, pp. 761–766.

Hartog, den, M.H., R. Babuška, H.J.R. Deketh, M. Alvarez Grima, P.N.W. Verhoef and H.B. Verbruggen (1997). Knowledge-based fuzzy model for performance prediction of a rock-cutting trencher. *International Journal of Approximate Reasoning 16*(1), 43–66.

Jang, J.-S.R. (1993). ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man & Cybernetics 23*(3), 665–685.

Jang, J.-S.R. and C.-T. Sun (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks 4*(1), 156–159.

Kaymak, U. (1994). Application of fuzzy methodologies to a washing process. Chartered designer thesis, Delft University of Technology, Control Lab., Faculty of El. Eng., Delft.

Klir, G.J. and B. Yuan (1995). *Fuzzy sets and fuzzy logic; theory and applications*. Prentice Hall.

Kosko, B. (1994). Fuzzy systems as universal approximators. *IEEE Trans. Computers 43*, 1329–1333.

Leonaritis, I.J. and S.A. Billings (1985). Input-output parametric models for non-linear systems. *International Journal of Control 41*, 303–344.

Ljung, L. (1987). *System Identification, Theory for the User*. New Jersey: Prentice-Hall.

Mamdani, E.H. (1977). Application of fuzzy logic to approximate reasoning using linguistic systems. *Fuzzy Sets and Systems 26*, 1182–1191.

Patterson, D.W. (1990). *Introduction to Artificial Intelligence and Expert Systems*. Englewoord Cliffs (NJ), USA: Prentice Hall.

Pedrycz, W. (1990). Relevancy of fuzzy models. *Information Sciences 52*, 285–302.

Pedrycz, W. (1993). *Fuzzy Control and Fuzzy Systems (second, extended,edition)*. John Willey and Sons, New York.

Pedrycz, W. (1995). *Fuzzy Sets Engineering*. Boca Raton, Fl.: CRC Press.

Setnes, M., R. Babuška, H.B. Verbruggen, M.D. Sánchez and H.F.P. van den Boogaard (1997). Fuzzy modeling and similarity analysis applied to ecological data. In *Proceedings FUZZ-IEEE'97*, Barcelona, Spain, pp. 415–420.

Smets, P., E.H. Mamdani, D. Dubois and H. Prade (Eds.) (1988). *Non-Standard Logics for Automated Reasoning*. Academic Press, London.

Strang, G. (1976). *Linear Algebra and Its Applications*. New York, U.S.A.: Academic Press.

Sugeno, M. and G.T. Kang (1986). Fuzzy modelling and control of multilayer incinerator. *Fuzzy Sets and Systems 18*, 329.

Sugeno, M. and K. Tanaka (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems 42*, 315–334.

Sugeno, M. and T. Yasukawa (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Trans. Fuzzy Systems 1*, 7–31.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Systems, Man and Cybernetics 15*(1), 116–132.

Tanaka, K., T. Ikeda and H.O. Wang (1996). Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stability, H_∞ control theory and linear matrix inequalities. *IEEE Transactions on Fuzzy Systems 4*(1), 1–13.

Tanaka, K. and M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems 45*(2), 135–156.

Terano, T., K. Asai and M. Sugeno (1994). *Applied Fuzzy Systems*. Boston: Academic Press, Inc.

Wang, L.-X. (1992). Fuzzy systems are universal approximators. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, San Diego, USA, pp. 1163–1170.

Wang, L.-X. (1994). *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*. New Jersey: Prentice Hall.

Yager, R.R. and D.P. Filev (1994). *Essentials of Fuzzy Modeling and Control*. New York: John Wiley.

Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man, and Cybernetics 1*, 28–44.

Zadeh, L.A. (1975). Calculus of fuzzy restrictions. In L.A. Zadeh, K.-S. Fu, K. Tanaka and M. Shimura (Eds.), *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, pp. 1–39. New York, USA: Academic Press.

Zeng, X.J. and M.G. Singh (1995). Approximation theory of fuzzy systems – MIMO case. *IEEE Trans. Fuzzy Systems 3*(2), 219–235.

Zhao, J. (1995). *Fuzzy logic in modeling and control*. PhD dissertation, CESAME, Louvain la Neuve, Belgium.

Zhao, J., V. Wertz and R. Gorez (1994). A fuzzy clustering method for the identification of fuzzy models for dynamical systems. In *9th IEEE International Symposium on Intelligent Control*, Columbus, Ohio, USA.

Zimmermann, H.-J. (1987). *Fuzzy Sets, Decision Making and Expert Systems*. Boston: Kluwer Academic Publishers.

Subject index

A
antecedent, 5
applications, 24

B
black-box model, 1

C
conjunctive form, 11
consequent, 5

D
decomposition, 12
defuzzification, 12
    center-of-gravity method, 12
    fuzzy-mean method, 13
degree of fulfillment, 9, 11
dynamic fuzzy system, 16, 20

F
first-principle model, 1
fuzzy
    clustering, 23
    expert system, 17
    graph, 8
    identification, 17
    mean, 13
    modeling, 2
    number, 2
    relation, 6
    system, 2

G
general function approximation, 4
generalization, 17
granularity, 18
gray-box model, 1

H
hybrid model, 1

I
identification, 4
inference
    Mamdani, 8
    Takagi–Sugeno, 14
intelligent modeling, control, 1

L
least-squares method, 19
linguistic
    model, 5
    term, 5
    variable, 5
linguistic model, 5

M
Mamdani
    inference, 8, 9
    model, *see* linguistic model
max-min inference, *see* Mamdani inference
mechanistic model, 1
multivariable fuzzy system, 10

N
NARX model, 16
neural network, 2, 4
neuro-fuzzy
    modeling, 17, 22
    network, 22

P
physical model, 1

R
rule-based model, 5

S
semi-mechanistic model, 1
singleton model, 12

T
Takagi–Sugeno
    inference, 14
    model, 5, 14, 20
template-based modeling, 20

W
white-box model, 1