Promising Fuzzy Modeling and Control Methodologies for Industrial Applications

R. Babuška[†], H.B. Verbruggen[†] and H. Hellendoorn[‡†]
[†] Delft University of Technology, Faculty of Information Technology and Systems
Control Engineering Laboratory, Mekelweg 4, P.O. Box 5031, 2600 GA Delft
The Netherlands, e-mail: R.Babuska@its.tudelft.nl
[‡] Siemens Nederland N.V., P.O. Box 16068, NL-2500 BB, The Hague
The Netherlands, e-mail: Hans.Hellendoorn@siemens.nl

ABSTRACT: Control problems in the industry are dominated by nonlinear and time-varying behavior, many sensors that measure all kinds of variables, many loops and interaction among the control loops. The extraction of (fuzzy) information out of raw data is very important and contains saving potential for industrial applications. Major types of rule-based fuzzy models are described which are based on pre-processed data available within the process. It is shown that these models can be used for different purposes because of their transparency and can be used in industrial applications which are partly described by first principle models and partly by experience built up by designers and operators. Fuzzy control can be based on human experience and can mimic the control actions of human operators. Fuzzy control can also be used in more sophisticated control schemes based on a (fuzzy) models of the process. Applications of fuzzy modeling and control are given.

KEYWORDS: Fuzzy modeling, fuzzy control, process industry, applications.

INTRODUCTION

Modern process control and production methods are confronted with a large number of requirements posed by increasing competition, environmental regulations, increase in energy and raw material costs and an increasing demand for high quality customer-tailored products. This leads to:

- An increasing demand for flexibility of the production process: operating the plant under varying throughput, different feed stock, varying product-mix and product grade. In other words: costumer-defined production in smaller amounts at the right due time, instead of producer-defined high-volume production.
- A strong demand for new production methods and new products. The production plants should be more compact to minimize energy consumption, the influence on the environment and the waste of materials. It is very important to introduce new products as fast as possible on the market (e.g. pharmaceutical industry), so the route from idea, first experiments, pilot plant and full-scale production should be as short as possible and less attractive alternatives should be excluded as soon as possible. This means the introduction of methods which are able to increase the complexity of the models of the process development in order to decide in an early stage how to proceed.
- The introduction of integrated information systems that are plant-wide and can handle various levels of automation in one concept: management, scheduling, planning, optimization, supervision, fault detection and fault diagnosis, control. This requires the ability to handle qualitative and quantitative information with different levels of precision and complexity and the use of different interfaces to the people responsible for each automation level.

The result of these requirements for process control on different levels of process automation can be summarized as follows:

- Fast and extensive changes in operating conditions and operation modes leading to more emphasis on the highly non-linear and time-varying behavior of the process.

- Many inner loops and utility feedbacks to decrease waste of material and energy consumption. This leads again to highly non-linear systems of high dimensions with much interaction between the control loops.
- The need for new models of the system which describe the non-linear time-varying behavior of the new production methods and production units in a more unified way and not only based on first principle descriptions or on a black box approach, but based on a combination of knowledge of system experts, measurements and operational experience.
- The need for dynamic and reactive responses on all levels of automation.

The control engineering community developed a complete framework for linear systems which resulted in a number of universal analysis and design methods for closed and open loop systems. Although many attempts were made during the last decades to develop analysis and design methods for non-linear systems, there is no universal method available at the moment for these systems.

This means that modern control practice is asking for methods which have still to be developed, and in many cases one relies on process models linearized around a number of operating points.

Can fuzzy modeling and control play a role in solving these problems to a certain extent? First of all fuzzy control was introduced to mimic the control actions of the human operator who was well experienced in controlling difficult to analyze and to describe control problems, for instance in the control of cement kilns (Holmblad and Østergaard, 1982).

In these applications a priori knowledge of the operators is translated into a knowledge based system and the final controller performs as well as the best operators. The main problems involved are the knowledge acquisition problem and the maintenance of the system when the plant is updated to produce other products, to handle different feedstocks, etc.

There is still a demand to apply these expert-based control methods for systems which are difficult to describe and to analyze and for systems for which extensive experience in operating the process is available from human operators and system designers. A recent example is the control of the weight and moisture of washing powders (Setnes, et al., 1997). It is expected that this kind of applications will continue to be introduced also in the near future either to advise the operators in control or fault diagnosis applications or as a real-time control alternative for direct control and supervision.

However, as will be shown in the next sections, fuzzy control can also be described as a non-linear mapping. In that case the controller acts as a non-linear controller which is a natural extension of the still extensively used classical PID-controller which is in 90% of all low-level control applications still the reliable workhorse.

Traditionally, in the process industry control methods has been used which are a combination of simple PID controllers and control actions performed by human operator. For simple processes the majority of the control tasks can be handled by intrinsic simple and low cost PID controllers. On a higher control level setpoints for the control loops are determined automatically by supervisory control algorithms or manually by operators.

Although operators are able to control complicated nonlinear and time varying systems after a long history of built up experience, PID controllers are not good at coping with non-linearity, operational constraints and interaction between process variables. Consequently it is common practice to regulate complex systems by human operators, although they suffer from inconsistent behavior, tiredness and apply different control actions depending on the experience of the people in a labor shift.

This paper will concentrate on the application of fuzzy techniques for modeling and control. In the next section an overview is given of data-driven construction of fuzzy models. It is shown that these models could be used as black box models but can also been made transparent and relatively simple to expose the main features of the system to the user or operator. A very important and promising direction is the use of semi-mechanistic models which allow the combination of known, analytically described parts of a process with unknown parts of the process found by a neural network approach or fuzzy techniques. In the latter case a better transparency is obtained. A neural network approach sometimes allows for faster computations which might be important in some real-time applications where a process model is part of the control strategy and not only used in the design stage.

Next an overview is given of some new developments in fuzzy control which might be of major importance in applying fuzzy techniques in the process industry.

FUZZY MODELING AND IDENTIFICATION

Development of mathematical models is essential for many disciplines of engineering and science. Models are used for different purposes such as simulations, design and analysis of systems, and for process control, monitoring and supervision. The traditional “mechanistic” approach to modeling is based on a thorough understanding of the nature and behavior of the actual system, and on a suitable mathematical treatment that leads to the development of a model. For incompletely understood processes, however, this approach may become laborious and inefficient. Large amount of process knowledge is qualitative and imprecise and as such cannot be readily transformed into traditional mathematical models based on differential and algebraic equations. Formal methods to incorporate such information in the development of models and controllers have been developed. They explore alternative representation schemes, using, for instance, natural language, rules, semantic networks, or qualitative models. Among these methods, techniques based on fuzzy sets and fuzzy logic represent a promising approach which has been developed considerably in recent years. (Pedrycz, 1993; Yager and Filev, 1994; Lin, 1994)

Fuzzy models can be seen as logical models which use “if–then” rules and logical operators to establish qualitative relationships among the variables in the model. Fuzzy sets serve as a smooth interface between qualitative variables involved in the rules and numerical domains of the inputs and outputs of the model. The rule-based nature of fuzzy models allows the use of information expressed in the form of natural language statements, and makes the models transparent to interpretation and analysis. At the same time, at the computational level, fuzzy models can be regarded as flexible mathematical structures, similar to neural networks or radial basis function networks, that can approximate a large class of nonlinear systems to a desired degree of accuracy. (Wang, 1992; Kosko, 1994; Zeng and Singh, 1995) This duality allows qualitative knowledge to be combined with quantitative data. Compared to other nonlinear approximation techniques, such as neural networks, fuzzy systems provide a more transparent representation of the nonlinear system under study, and can also be given a linguistic interpretation in the form of rules. In this way, process data can be translated in a model and analyzed in a manner very similar to what are people acquainted with.

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 1 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.

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 $f\colon X \to Y$ 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 $X \times Y$

Table 1: Crisp and fuzzy information in systems.

system description	input data	resulting output data	mathematical framework
crisp	crisp	crisp	functional analysis
crisp	fuzzy	fuzzy	extension principle (Zadeh, 1975)
fuzzy	crisp/ fuzzy	fuzzy	fuzzy relational calculus

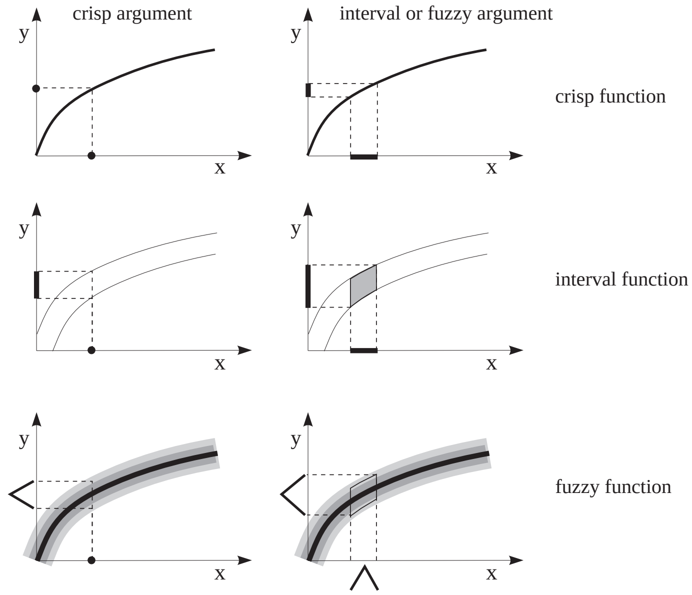


Figure 1: Evaluation of a crisp, interval and fuzzy function for crisp, interval and fuzzy arguments.

(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).

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.

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. (Yager and Filev, 1994; Pedrycz, 1990) 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 fuzzy 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; Zeng and Singh, 1995)

DATA EXTRACTION AND PREPROCESSING

We have to keep in mind that the gathering of data necessary to build a model is not always straight forward. In some case we should rely on past process data which may not complete or not representative (different operating point, time-varying behavior of the plant).

Obviously, it is quite likely that in a real process it is not possible to disturb the system to gather the requested data because this could cause difficulties in operation. Although to build a good model a representative data set for the expected operating range (in closed loop control) is crucial, this data set can not always be obtained. Data-mining techniques from data over a long period of time might be useful when the necessary data is available, spread across the entire operating region.

We have to keep in mind that it is very difficult to set general rules for getting the most valuable information from available historical data and that each application is different depending on the amount of data, the usefulness, the range in which the data reside, etc.

However, the following steps can generally be distinguished:

1. *Preprocessing of the data, also called reconciliation.* This is the removal of those data that must be due to failures (sensor drift, fall out of equipment, values outside the defined domain). One has to check ranges, expected means and standard deviations if the data. Furthermore, filtering plays an important role in smoothing data in order to keep valuable data. The remaining data is concerned with pre-processed measurements (e.g. flow, pressure, temperature), control information (e.g. local controllers, valve openings), machine states (e.g. on/off) and, quality data of products. Non-numerical data can be mapped on a possibly fuzzy scale.

   The following procedures might be useful to get the most valuable data from a large data set. The data must also be checked on sensor failures and runaways: substitute expected input data for faulty data using a (fuzzy) filtering algorithms, remove columns of constants from the data set, etc. Finally, normalize the data to prevent the fact that sensors defined on numerical larger domains get higher weights in future data processing. Some of the methods using non-normalized data are much more sensitive than others.

2. *Reducing the data.* In many cases the amount of data is overwhelming when gathered over a long period and when a large number of sensors is generating data which is not always relevant for the problems at hand. Correlation techniques can be used to determine the influence of inputs on outputs (only a linear correspondence is found). A ranking test can also be used for instance by applying a radial base function network. By substituting each variable by its mean value, the error of the RBF model with the mean value as compared to the actual value gives a ranking of the error values and consequently of the importance of the input variables involved. Another well-known method is Principal Component Analysis (PCA). Set up a square matrix of data, calculate the eigenvalues and eigenvectors and in a normalized data domain the size of the eigenvalues shows the importance of the attached eigenvector. The inputs which contribute most to these eigenvalues are the ones which should be kept in the system.

3. *Modeling the data*, depending on the aim the model is used for: control, forecasting, advise, etc. This will be the main subject of the remaining part of this section in which fuzzy identification and modeling is highlighted. Other possible approaches are: Radial Base Functions (RBF) Networks, Kohonen mappings and similar mappings, e.g. the Sammon mapping. These mappings show the structure of the data in a two- or three-dimensional figure, showing the behavior of the process visually, which is important, as sensors outputs as well as characteristic plant parameters tend to drift. It can easily be shown that RBF-networks can be mapped on a special class of fuzzy systems and vice versa (Jang and Sun, 1993).

4. *Rule extraction*; the behavior of the process is described in terms of causal rules and presented to the operator in a transparent way. There are several ways to do this. One is via RBF-networks. This network can be translated into a fuzzy system. However, the complexity of the rule base thus found is a disadvantage. Each rule has its own premise with their own fuzzy sets, which makes it hard to understand the meaning of the rules. A principal component approach (PCA) can be used to reduce the number of inputs, but again the resulting system can be hard to understand by the people from the plant. If one is able to determine a small number of really important inputs, it is better to use these inputs for RBF modeling and rule production. Another possibility is the use of fuzzy clustering methods. Again, the number of rules and the transparency of the rule base can be a disadvantage. However, in the sequel it will be shown that many interesting simplification and reduction methods exist which will diminish the complexity problem and reveal the necessary transparency of the model without too much deterioration of the validity of the model.

5. *Drift analysis* is easy when one knows the natural state of the plant. In this case, one has to describe this state and calculate the difference between the current state and the predefined state. In many cases, however, one does not know the standard state of a plant or a system, in this case one has to follow carefully the movements in the data. If there are jumps in several variables that are abnormal with respect to time, one should give a signal. This usually implies that one has to do a re-identification. This re-identification is dangerous, because one has to know which data to chose for this process. If one uses data that is too local with respect to time, one might be learning a system that has nothing in common with the system as a whole. If a new drift occurs, the system cannot be controlled appropriately. A solution is to save some standard data and use them as extra data to re-identify the system. This is dangerous too, because systems have natural drifts and might never recur to such a 'standard' state. Careful analysis is needed in this case.

RULE-BASED FUZZY MODELS

Different methods have been developed using fuzzy set theory to model systems, such as rule-based fuzzy systems, (Zadeh, 1973) fuzzy linear regression methods, (Tanaka, et al., 1982) fuzzy based on cell structures. (Smith, et al., 1994) This chapter deals with 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, three main types of rule-based fuzzy models are distinguished:

- *Linguistic fuzzy model*: both the antecedent and the consequent are fuzzy propositions.
- *Fuzzy relational model*: generalization of the linguistic model, the relation between the antecedent and consequent terms is fuzzy.
- *Takagi–Sugeno (TS) fuzzy model*: the antecedent is a fuzzy proposition, the consequent is a crisp function.

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

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:

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

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:[1] $\mathbf{x} \in X \subset \mathbb{R}^p$ and $\mathbf{y} \in Y \subset \mathbb{R}^q$. The membership functions of the antecedent and consequent fuzzy sets are the mappings: $\mu(\mathbf{x})\colon X \to [0,1]$ and $\mu(\mathbf{y})\colon Y \to [0,1]$, respectively. 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.

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.

RELATIONAL REPRESENTATION OF LINGUISTIC MODELS

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_i implies B_i". 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, $A_i \wedge B_i$, the interpretation of the if-then rules is "it is true that A_i and B_i simultaneously hold". This relationship is symmetrical and can be inverted. For simplicity, in this text we restrict ourselves to the Mamdani (conjunction) method. The relation R_i 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:

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

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 ($\circ$):

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

Since the relation R can be visualized as a "fuzzy" graph in $X \times Y$, the linguistic fuzzy model is sometimes called a *fuzzy graph*. The relational composition (4) can be regarded as a function evaluation on the fuzzy graph, see also Fig. 1.

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

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

avoided. To show this, suppose an input fuzzy value $\tilde{x} = A'$, for which the output value B' is given by the relational composition:

$$\mu_{B'}(\mathbf{y}) = \max_X \big(\mu_{A'}(\mathbf{x}) \wedge \mu_R(\mathbf{x}, \mathbf{y})\big), \quad y \in Y\,. \tag{5}$$

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

$$\mu_{B'}(\mathbf{y}) = \max_X \Big[\mu_{A'}(\mathbf{x}) \wedge \max_{1 \le i \le K} \big(\mu_{A_i}(\mathbf{x}) \wedge \mu_{B_i}(\mathbf{y})\big)\Big]\,. \tag{6}$$

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

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

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

$$\mu_{B'}(\mathbf{y}) = \max_{1 \le i \le K} \big(\beta_i \wedge \mu_{B_i}(\mathbf{y})\big), \quad \mathbf{y} \in Y\,. \tag{8}$$

This algorithm, called the *max-min* or *Mamdani inference*, is summarized in Algorithm 1 and visualized in Fig. 2.

Algorithm 1 *Mamdani (max-min) inference*

---

1. Compute the degree of fulfillment by:

$$\beta_i = \max_X \left[\mu_{A'}(\mathbf{x}) \wedge \mu_{A_i}(\mathbf{x})\right], \quad i = 1, 2, \ldots, K\,.$$

2. Derive the output fuzzy sets B'_i:

$$\mu_{B'_i}(\mathbf{y}) = \beta_i \wedge \mu_{B_i}(\mathbf{y}), \quad \mathbf{y} \in Y, \quad i = 1, 2, \ldots, K\,.$$

3. Aggregate the output fuzzy sets B'_i into a single fuzzy set B':

$$\mu_{B'}(\mathbf{y}) = \max_{1 \le i \le K} \mu_{B'_i}(\mathbf{y}), \quad \mathbf{y} \in Y\,.$$

Note that for a singleton fuzzy set[2] A', Step 1 of the above algorithm simplifies to $\beta_i = \mu_{A_i}(\mathbf{x}')$.

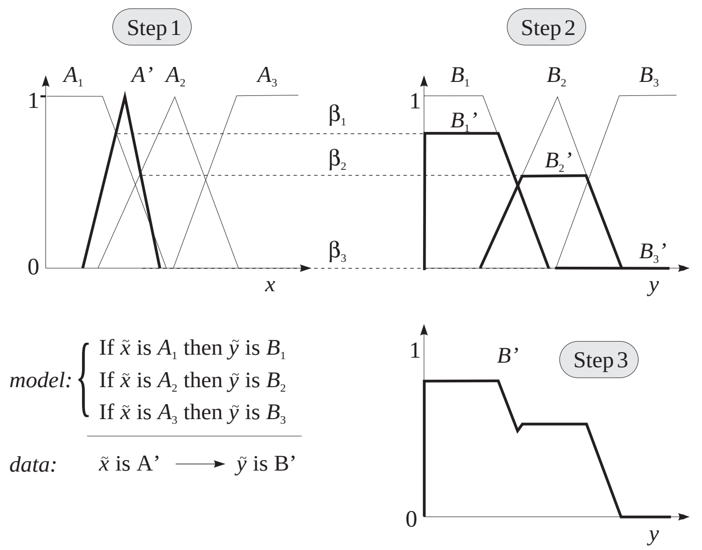


Figure 2: A schematic representation of the Mamdani inference algorithm.

[2]A singleton fuzzy set is a fuzzy representation of a crisp number. Suppose a crisp number $\mathbf{x}'$, the corresponding fuzzy singleton A' is defined by: $\mu_{A'}(\mathbf{x}') = 1$ and $\mu_{A'}(\mathbf{x}) = 0, \forall \mathbf{x} \in X,\ \mathbf{x} \neq \mathbf{x}'$.

The Mamdani inference method does not require the discretization of the domains and can work with analytically defined membership functions. It also can make use of learning algorithms, as discussed in Section .

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, often 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:

$$\mathcal{R}_i\text{: } \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } \ldots \textbf{ and } x_p \text{ is } A_{ip} \textbf{ then } y \text{ is } B_i, \quad i = 1, 2, \ldots, K\,. \tag{9}$$

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 1) is given by:

$$\beta_i = \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \cdots \wedge \mu_{A_{ip}}(x_p), \quad i = 1, 2, \ldots, K\,. \tag{10}$$

Other conjunction operators, such as the product, can be used. A set of rules in the conjunctive antecedent form divide the input domain into a lattice of fuzzy hyperboxes, parallel with the axes. Each of the hyperboxes is a Cartesian product-space intersection of the corresponding univariate fuzzy sets. This is shown in Fig. 3a. The number of rules in the conjunctive form, needed to cover the entire domain, is given by:

$$K = \Pi_{i=1}^{p} N_i, \tag{11}$$

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

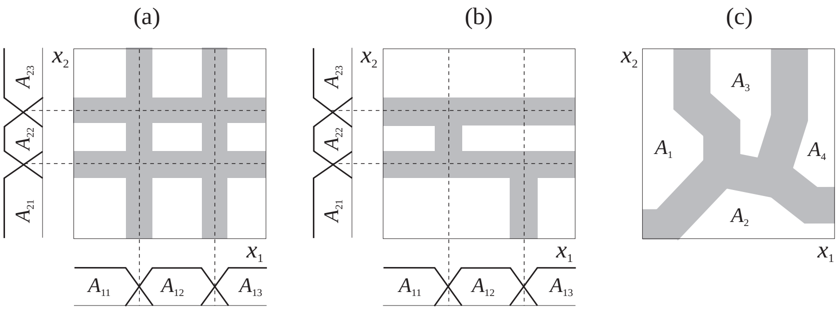


Figure 3: Different partitions of the antecedent space. Gray areas denote the overlapping regions of the fuzzy sets.

By combining conjunctions, disjunctions and negations, various partitions of the antecedent space can be obtained. The boundaries remain restricted to the rectangular grid defined by the fuzzy sets of the individual variables, see Fig. 3b. As an example consider the rule antecedent covering the lower left corner of the antecedent space in this figure:

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

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

$$\beta = [1 - \mu_{A_{13}}(x_1)] \wedge \mu_{A_{21}}(x_2)\,. \tag{12}$$

The antecedent form with multivariate membership functions (1) is the most general one, as there is no restriction as to the shape of the fuzzy regions. The boundaries between these regions can be arbitrarily curved and opaque to the axes, as depicted in Fig. 3c. 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. 3c still can be projected onto x_1 and x_2 to obtain an approximate linguistic interpretation of the regions described.

Another way of reducing the complexity is to decompose a multivariable fuzzy system into several subsystems with fewer inputs per rule base. These 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. 4. 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 these two 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, see eq. (11).

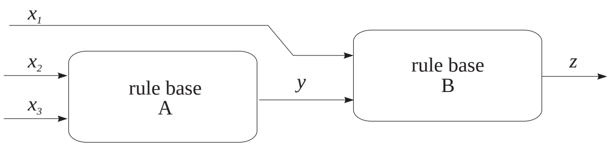


Figure 4: Cascade connection of two rule bases.

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':

$$y' = \text{cog}(B') = \frac{\sum_{j=1}^{F} \mu_{B'}(y_j) y_j}{\sum_{j=1}^{F} \mu_{B'}(y_j)}, \tag{13}$$

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.

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:

$$R_i\text{:} \quad \textbf{If } \tilde{\mathbf{x}} \text{ is } A_i \textbf{ then } y = b_i, \quad i = 1, 2, \ldots, K\,. \tag{14}$$

This model is called the *singleton model*. A simplified version of (13), the *fuzzy-mean* defuzzification method is usually used with this model:

$$y = \frac{\sum_{i=1}^{K} \beta_i \, b_i}{\sum_{i=1}^{K} \beta_i}\,. \tag{15}$$

The singleton fuzzy model belongs to a general class of general function approximators, called the basis functions expansion (Friedman, 1991) taking the form:

$$y = \sum_{i=1}^{K} \phi_i(\mathbf{x}) b_i\,. \tag{16}$$

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. Multi-linear interpolation between the rule consequents is obtained if:

1. the antecedent membership functions are trapezoidal, pairwise overlapping and the membership degrees sum up to one for each domain element, and

2. the product operator is used to represent the logical **and** connective in the rule antecedents.

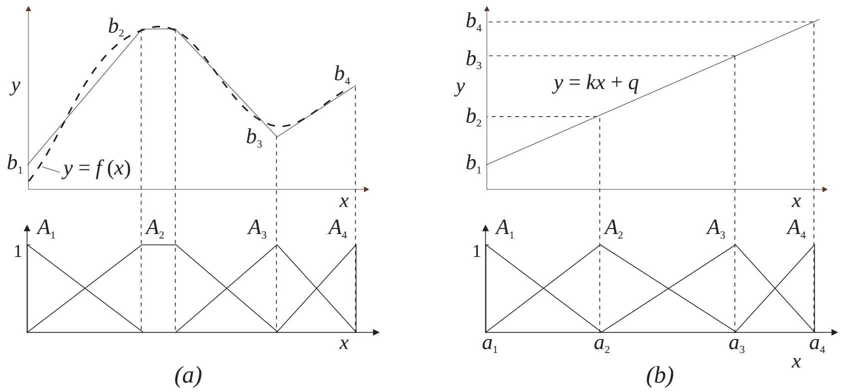


Figure 5: Singleton model with triangular or trapezoidal membership functions results in a piecewise linear input-output mapping (a), of which a linear mapping is a special case (b).

The input-output mapping of the singleton model is then piecewise multi-linear, as shown in Fig. 5a.

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

$$y = \mathbf{k}^T \mathbf{x} + q = \sum_{j=1}^{p} k_j x_j + q\,. \tag{17}$$

In this case, the antecedent membership functions must be triangular, see Fig. 5b. 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. The consequent singletons can be computed by evaluating the desired mapping (17) for the cores a_{ij} of the antecedent fuzzy sets A_{ij}:

$$b_i = \sum_{j=1}^{p} k_j a_{ij} + q\,. \tag{18}$$

FUZZY RELATIONAL MODEL

A fuzzy relational model (Pedrycz, 1993; Pedrycz, 1984) can be seen as an extension of the linguistic model, where the mapping between the input and output fuzzy sets is represented by a fuzzy relation.

Denote $\mathcal{A}_j$ and B the sets of linguistic terms defined for the antecedent and consequent variables, respectively:

$$\begin{aligned} \mathcal{A}_j &= \{A_{jk} | k = 1, 2, \ldots, N_j\}, \quad j = 1, 2, \ldots, p, \\ \mathcal{B} &= \{B_k | k = 1, 2, \ldots, M\}\,. \end{aligned}$$

The rule base (9) can be represented as a crisp relation R between the linguistic terms in the antecedents and in the consequent:

$$R: \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n \times \mathcal{B} \to \{0, 1\}\,. \tag{19}$$

Further, by denoting $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ the Cartesian space of the antecedent linguistic terms, eq. (19) can be simplified to $R: \mathcal{A} \times \mathcal{B} \to \{0, 1\}$. A fuzzy relational model is obtained by generalizing R to a fuzzy relation:

$$R: \mathcal{A} \times \mathcal{B} \to [0, 1]\,. \tag{20}$$

In this model, each rule contains all the possible consequent terms, each with a different weighting factor, given by the respective elements of the fuzzy relation. With this weighting, one can more easily fine-tune the model, e.g., to fit some data. A single-input, single-output fuzzy relational model is depicted in Fig. 6.

It should be stressed that the relation R in (20) is different from the relation (3) encoding linguistic if–then rules. The latter relation is a multidimensional membership function defined in the product space of the input and output domains. Each element of this relation represents the degree of association between the individual *crisp* elements in the antecedent

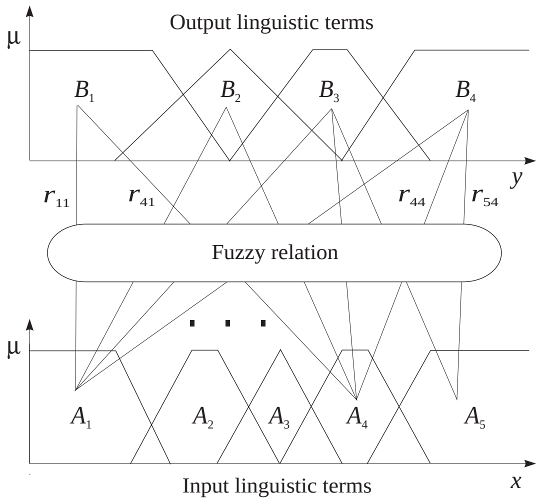


Figure 6: Fuzzy relational model. Reproduced from, (Babuška and Verbruggen, 1996) ©1996 Elsevier Science Ltd.

and consequent domains. In fuzzy relational models, however, the fuzzy relation represents associations between the individual linguistic terms.

The main advantage of the relational model is that the input–output mapping can be fine-tuned without changing the consequent fuzzy sets. In the linguistic model, the outcomes of the individual rules are restricted to the grid given by the centroids of the output fuzzy sets, which is not the case in the relational model.

The inference in the relational model proceeds in the following three steps:

Algorithm 2 *Inference in fuzzy relational model*

1. Compute the degree of fulfillment by:

$$\beta_i = \mu_{A_{i1}}(x_1) \wedge \cdots \wedge \mu_{A_{ip}}(x_p), \; i = 1, 2, \ldots, K\,.$$

2. Apply relational composition: $\boldsymbol{\omega} = \boldsymbol{\beta} \circ R$, given by:

$$\omega_j = \max_{1 \le i \le K} \big(\beta_i \wedge r_{ij}\big), \quad j = 1, 2, \ldots, M\,.$$

3. Defuzzify the consequent fuzzy set by:

$$y = \sum_{l=1}^{M} \omega_l \, b_l \Big/ \sum_{l=1}^{M} \omega_l\,,$$

where $b_l = \mathrm{cog}(\mathcal{B}_l)$.

TAKAGI–SUGENO MODEL

The linguistic model, introduced in 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 modeling and mathematical regression, 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:

$$\mathcal{R}_i\colon \textbf{If } \mathbf{x} \text{ is } A_i \textbf{ then } \mathbf{y}_i = \mathbf{f}_i(\mathbf{x}), \qquad i = 1, 2, \ldots, K\,. \tag{21}$$

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:

$$\mathcal{R}_i\colon \ \textbf{If } \mathbf{x} \text{ is } A_i \textbf{ then } y_i = \mathbf{a}_i^T \mathbf{x} + b_i, \qquad i = 1, 2, \ldots, K, \tag{22}$$

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 (14) is obtained.

INFERENCE MECHANISM

The inference formula of the TS model is a straightforward extension of the singleton model inference (15):

$$y = \frac{\sum_{i=1}^K \beta_i y_i}{\sum_{i=1}^K \beta_i} = \frac{\sum_{i=1}^K \beta_i (\mathbf{a}_i^T \mathbf{x} + b_i)}{\sum_{i=1}^K \beta_i}\,. \tag{23}$$

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 linear approximation of that function, see Fig. 7.

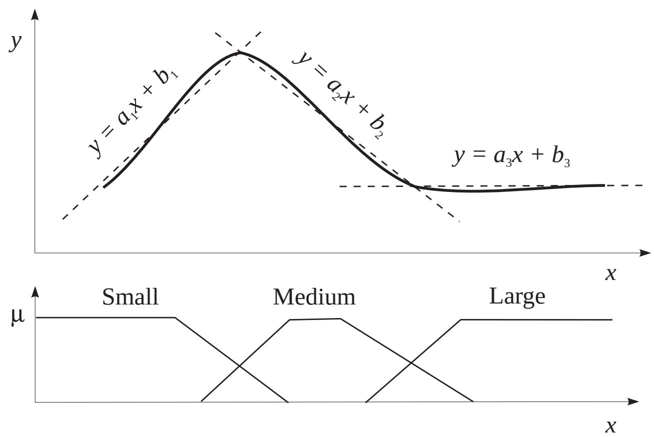


Figure 7: Takagi–Sugeno fuzzy model as a smoothed piece-wise linear approximation of a nonlinear function.

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

$$\gamma_i(\mathbf{x}) = \beta_i(\mathbf{x}) / \sum_{j=1}^K \beta_j(\mathbf{x})\,. \tag{24}$$

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

$$y = \left(\sum_{i=1}^K \gamma_i(\mathbf{x}) \mathbf{a}_i^T \right) \mathbf{x} + \sum_{i=1}^K \gamma_i(\mathbf{x}) b_i = \mathbf{a}^T(\mathbf{x})\mathbf{x} + b(\mathbf{x})\,. \tag{25}$$

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.:

$$\mathbf{a}(\mathbf{x}) = \sum_{i=1}^K \gamma_i(\mathbf{x}) \mathbf{a}_i, \quad b(\mathbf{x}) = \sum_{i=1}^K \gamma_i(\mathbf{x}) b_i\,. \tag{26}$$

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. 8.

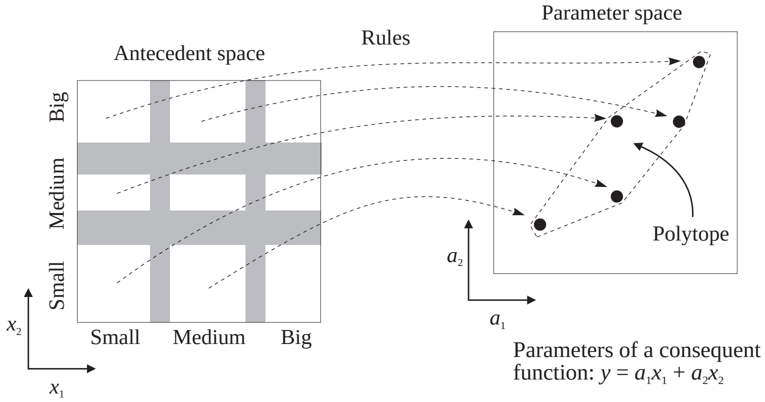


Figure 8: A TS model with affine consequents can be regarded as a mapping from the antecedent space to the space of the consequent parameters.

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 and to analyze their stability. (Tanaka and Sugeno, 1992; Zhao, 1995; Tanaka, et al., 1996)

MODELING DYNAMIC SYSTEMS

Before discussing dynamic fuzzy models, let us recall that time-invariant dynamic systems are in general modeled 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

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \tag{27}$$

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 often applied. The most common is the NARX (Nonlinear AutoRegessive with eXogenous input) model:

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)) \,. \tag{28}$$

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

$$\begin{aligned} R_i\colon \quad & \textbf{If } y(k) \text{ is } A_{i1} \textbf{ and } y(k-1) \text{ is } A_{i2} \textbf{ and} \dots y(k-n+1) \text{ is } A_{in} \\ & \textbf{and } u(k) \text{ is } B_{i1} \textbf{ and } u(k-1) \text{ is } B_{i2} \textbf{ and} \dots u(k-m+1) \text{ is } B_{im} \\ & \textbf{then } y(k+1) \text{ is } c_i \,. \end{aligned} \tag{29}$$

In this sense, we can say that the dynamic behavior is taken care of by external dynamic filters added to the fuzzy system, see Fig. 9. In (29), the input dynamic filter is a simple generator of the lagged inputs and outputs, and no output filter is used.

Since fuzzy models can approximate any smooth function to any degree of accuracy, (Wang, 1992) models of type (29) can approximate any observable and controllable modes of a large class of discrete-time nonlinear systems. (Leonaritis and Billings, 1985)

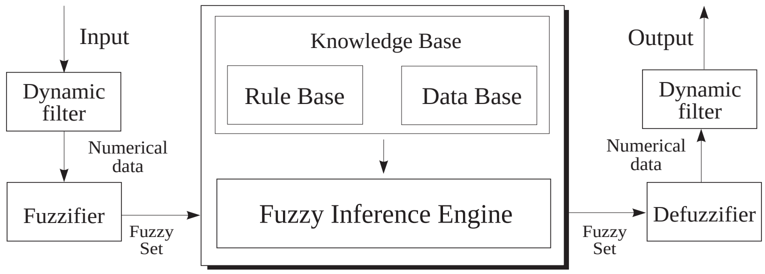


Figure 9: A generic fuzzy system with fuzzification and defuzzification units and external dynamic filters. Reproduced from, (Babuška and Verbruggen, 1996) ©1996 Elsevier Science Ltd.

BUILDING FUZZY MODELS

Two common sources of information for building fuzzy models are the *prior knowledge* and *data* (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, et al., 1997)
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. This approach can be termed *rule extraction*. Fuzzy clustering is one of the techniques that are often applied.(Yoshinari, et al., 1993; Nakamori and Ryoke, 1994; Babuška and Verbruggen, 1997a)

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.

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 the approximation of (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. 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 (28) 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, relational, Takagi-Sugeno) and the antecedent form (refer to Section). Important aspects are the purpose of modeling and the type of 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). In fuzzy relational model, this mapping is encoded in the fuzzy relation. Takagi-Sugeno models have parameters in antecedent membership functions and in the consequent functions ($\mathbf{a}$ and b for the affine TS model).

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 (typically 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.

DATA-DRIVEN ACQUISITION AND 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:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T, \quad \mathbf{y} = [y_1, \ldots, y_N]^T. \tag{30}$$

LEAST-SQUARES ESTIMATION OF CONSEQUENTS

Note that the defuzzification formulas of the singleton and TS models are linear in the consequent parameters, $\mathbf{a}_i$, b_i (see equations (15) and (23), respectively). 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 (24) 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 $\mathbf{X}_e$

$$\mathbf{X}' = [\mathbf{\Gamma}_1\mathbf{X}_e,\ \mathbf{\Gamma}_2\mathbf{X}_e,\ \ldots,\ \mathbf{\Gamma}_K\mathbf{X}_e]\ . \tag{31}$$

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

$$\theta = \left[\mathbf{a}_1^T, b_1, \mathbf{a}_2^T, b_2, \ldots, \mathbf{a}_K^T, b_K\right]^T\ . \tag{32}$$

Given the data $\mathbf{X}, \mathbf{y}$, eq. (23) now can be written in a matrix form, $\mathbf{y} = \mathbf{X}'\theta + \epsilon$. It is well known that this set of equations can be solved for the parameter θ by:

$$\theta = \left[(\mathbf{X}')^T\mathbf{X}'\right]^{-1}(\mathbf{X}')^T\mathbf{y}\ . \tag{33}$$

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:

$$[\mathbf{a}_i^T, b_i]^T = \left[\mathbf{X}_e^T\mathbf{\Gamma}_i\mathbf{X}_e\right]^{-1}\mathbf{X}_e^T\mathbf{\Gamma}_i\mathbf{y}\ . \tag{34}$$

In this case, the consequent 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 $1 \le i \le K$, and by setting $\mathbf{X}_e = \mathbf{1}$, equations (33) and (34) directly apply to the singleton model (14).

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

$$y(k+1) = y(k) + u(k)e^{-3|y(k)|}\ . \tag{35}$$

We use a stepwise input signal to generate with this equation a set of 300 input–output data pairs (see Fig. 11a). 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:

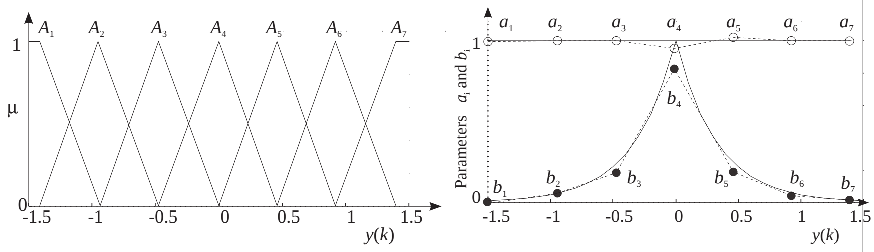
$$\textbf{If } y(k) \text{ is } A_i \textbf{ then } y(k+1) = a_i y(k) + b_i u(k), \tag{36}$$

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. 10a.

The consequent parameters can be estimated by the least-squares method. Figure 10b shows 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. Their values, $\mathbf{a}^T = [1.00, 1.00, 1.00, 0.97, 1.01, 1.00, 1.00]$ 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 of (35). Validation of the model in simulation using a different data set is given in Fig. 11b. □

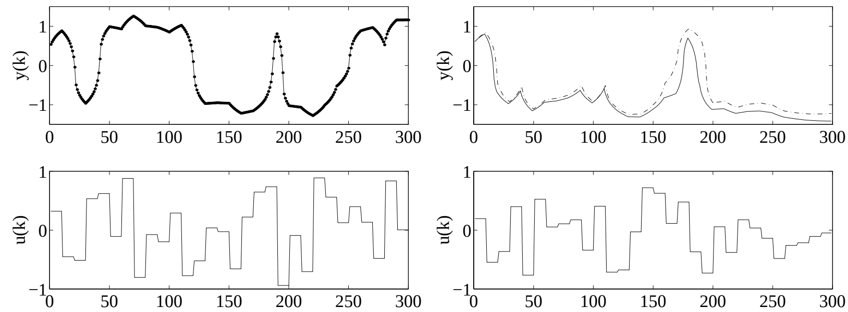
The transparent local structure of the TS model facilitates the combination of local models obtained by parameter estimation and linearization of known mechanistic (white-box) models. If measurements are available only in certain regions of the process' operating domain, parameters for the remaining 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 (22):

$$\mathbf{a}_i = \left.\frac{df}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{c}_i},\quad b_i = f(\mathbf{c}_i)\ . \tag{37}$$



(a) Membership functions. *(b) Estimated consequent parameters.*

Figure 10: (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). Reproduced from, (Babuška and Verbruggen, 1997b) ©1997 Taylor & Francis Ltd.



(a) Identification data. *(b) Validation.*

Figure 11: Identification data set (a), and performance of the model on a validation data set (b). Solid line: process, dashed-dotted line: model. Reproduced from, (Babuška and Verbruggen, 1997b) ©1997 Taylor & Francis Ltd.

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. However, the complexity of the system's behavior is typically not uniform. Some operating 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.

NEURO-FUZZY MODELING

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 and nonlinear optimization 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 and Sun, 1993; Brown and Harris, 1994; Jang, et al., 1997; Jang, 1993) Figure 12 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 (23).

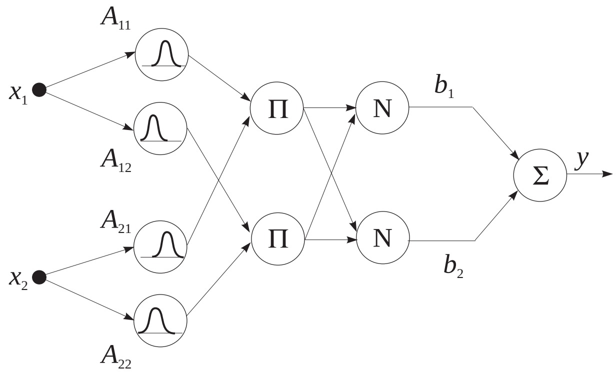


Figure 12: An example of a singleton fuzzy model with two rules represented as a (neuro-fuzzy) network.

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

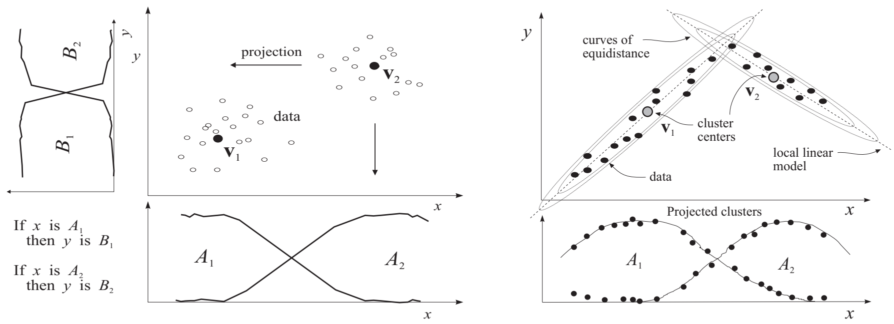
$$\mu_{A_{ij}}(x_j; c_{ij}, \sigma_{ij}) = \exp\left(-(\frac{x_j - c_{ij}}{2\sigma_{ij}})^2\right), \tag{38}$$

the c_{ij} and σ_{ij} parameters can be adjusted by gradient-descent learning algorithms, such as back-propagation. (Jang, et al., 1997) This allows for a fine-tuning of the fuzzy model to the available data in order to optimize its prediction accuracy.

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 is depicted in Fig. 13a, 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 13: Identification by fuzzy clustering. Reproduced from, (Babuška and Verbruggen, 1997b) ©1997 Taylor & Francis Ltd.

Fuzzy if-then rules can be extracted by projecting the clusters onto the axes. Figure 13a 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 elliptical shape, (Gustafson and Kessel, 1979) see Fig. 13b. From such clusters, the antecedent membership functions and the consequent parameters of the Takagi–Sugeno model can be extracted: (Babuška and Verbruggen, 1995)

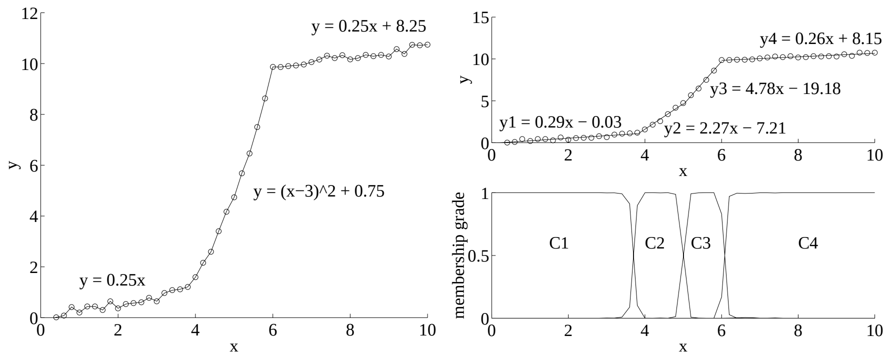
$$\textbf{If } x \text{ is } A_1 \textbf{ then } y = a_1 x + b_1\,,$$
$$\textbf{If } x \text{ is } A_2 \textbf{ then } y = a_2 x + b_2\,.$$

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 (33) or (34).

Example 2 Consider a nonlinear function $y = f(x)$ defined piece-wise by:

$$\begin{array}{lcll} y & = & 0.25x, & \text{for } x \leq 3 \\ y & = & (x-3)^2 + 0.75, & \text{for } 3 < x \leq 6 \\ y & = & 0.25x + 8.25, & \text{for } x > 6 \end{array} \tag{39}$$

Figure 14a 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 (39).

(b) Cluster prototypes and the corresponding fuzzy sets.

Figure 14: Approximation of a static nonlinear function using a Sugeno–Takagi fuzzy model.

The data set $\{(x_i, y_i) | i = 1, 2, \ldots, 50\}$ was clustered into four hyperellipsoidal clusters. The upper plot of Fig. 14b 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:

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

Note that the consequents of $\mathcal{R}_1$ and $\mathcal{R}_4$ correspond almost exactly to the first and third equation (39). Consequents of $\mathcal{R}_2$ and $\mathcal{R}_3$ are approximate tangents to the parabola defined by the second equation of (39) in the respective cluster centers. □

ADVANCED FUZZY CONTROL IN THE PROCESS INDUSTRY

In this section of the chapter an outline is given of some promising control methods based on fuzzy techniques which are expected to be potential solutions for problems arising in process control strategies in modern process industry. After an introduction, short descriptions are given of Fuzzy Inverse Control and Fuzzy Internal Model Control.

Next, a more extensive description of Fuzzy Model-based Predictive Control is given which is closely related to the approach accepted in the process industry for linear systems with constraints. Next, decision making in control and especially the application to model-based predictive control, is shortly introduced. Finally, problems arising when fuzzy logic is used for Multi-Input Multi-Output (MIMO) systems are shortly introduced. Multivariable control plays an important role in the process industry, because many control variables have to be controlled simultaneously and moreover, many interactions exist between manipulated variables and the different control variables. Much research has still to be done in this area, but some promising directions are already in progress.

ADVANCED CONTROL SCHEMES BASED ON SIMPLE CONTROLLERS

A control strategy which is becoming quite popular in Process Control is the model-based approach. In the remainder of this section the explicit use of a model of the system as part of the control algorithm is addressed.

FUZZY INVERSE CONTROL

The most simple approach to design a controller is a completely open loop control strategy, in which the controller is the inverse of the process. In case we have a non-linear process model it is clear that a non-linear inverse controller will be found and that a perfect controlled system will be obtained. However, we have to keep in mind that the system can exhibit considerable delay times which should be taken into account and also other dynamic and static characteristics of the process should be well-known.

Besides, the remaining process model will never be an exact copy of the real process and there will be always disturbances acting on the process which will not be taken into account. However, many of these problems can be overcome using the control configuration called Internal Model Control.

When a fuzzy non-linear model is obtained from a (partly) unknown and highly complicated process, we have still the problem of inverting the fuzzy model.

Moreover, you have to keep in mind that these inversions should be computationally fast for its use in an in-line real-time control structure. This is simple for singleton fuzzy models. This type of models belongs to a general class of function approximations. Another type of model which can be inverted exactly is the Takagi-Sugeno type with affine inputs $u(k)$. However, constraining the model to an affine one reduces usually the accuracy of the model.

We should also keep in mind that inversion based approaches can only be applied to stable systems with a minimum phase character.

We normally speak about partial inversion and not about global inversion. This means in the case of a fuzzy model of a SISO system in which additional inputs are produced to get a dynamic model, that only one of the input variables of the model becomes the output of the inverted model and the output of the model becomes one of the inputs of the inverted model. The original model of the plant consists of n inputs of the fuzzy model $x_1, x_2, \ldots, x_n$ (previous inputs and outputs) and only one output y.

FUZZY INTERNAL MODEL CONTROL

To overcome some of the problems introduced by the open loop approach presented in the previous subsection (*Fuzzy Inverse Control*), a feedback approach was introduced by Garcia and Morari (1982)). This approach called *Internal Model Control* (IMC) consists of three parts:

- a model to predict the effect of the control action on the system
- a controller based on the inverse of the process model
- a filter to increase robustness to model mismatch and disturbances.

When the model and the controller is a fuzzy system, this approach is called: *Fuzzy IMC*. A general scheme is depicted in Fig. 15.

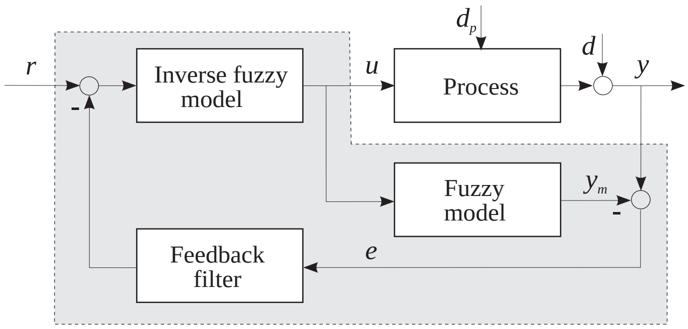


Figure 15: Internal model control scheme.

It is clear that with a perfect model describing the process and no disturbances acting on the process the feedback signal becomes zero and open loop control is obtained. If a disturbance d is acting on the process, the feedback signal is equal to d and is not affected by the control action, but simply subtracted from the reference. So the effects of output disturbances are completely canceled.

It can be proved that under relatively weak circumstances (steady state of the control is inverse of the process) control without steady state errors can be obtained. A filter is introduced to increase the robustness of the system to model mismatch described by d_p and measurement noise.

It can reduce the loop gain to stabilize the system and smooth out noisy or rapidly changing signals, reducing the transient response of the IMC scheme. For non-linear systems the filter must be designed for the part of the system where the dynamics is fast.

However, depending on the model mismatch and the disturbances contaminating the system, it is not possible to design the filter in a predetermined way, when the process exhibits a non-linear behavior, which is a major design problem.

FUZZY MODEL-BASED PREDICTIVE CONTROL

The concept of Model-Based Predictive Control (MBPC) was introduced about two decades ago. Although the concept is quite general a broad range of different methods are based on this concept depending on how the main ingredients of these methods are translated to specific solutions.

The basic concepts in all predictive control methods are the following:

- Use of a model to predict the process outputs at future discrete time instants over a certain prediction horizon (prediction model).
- Computation of a sequence of future control actions over a certain control horizon by minimizing a certain objective function, which requires that the predicted process outputs are as close as possible to the desired reference trajectories, under given operation constraints (optimization process).
- Receding horizon strategy, so that at each sampling instant the optimization process is repeated with the new measurements which has become available, and the first control action in the calculated control sequence is applied to the process (real-time receding horizon control).
- Sometimes an additional concept is introduced to compensate for model-plant mismatch and influences of disturbances.

Because of the explicit use of a process model and an optimization approach, MBPC can be applied to complex systems, e.g. multivariable, non-minimum-phase, open-loop unstable, non-linear, or processes with a long delay time. Moreover, the method can efficiently deal with constraints on input and output variables of the process.

MBPC has been well accepted in industry due to the generality of the method and there exists already a large number of industrial applications (Richalet, 1993).

Extensive software packages are available which fit well in industrial instrumentation and process control systems. However, these packages are based on linear MIMO process models and on classical quadratic objective functions, including hard constraints. Extensive use is made of quadratic programming method in the optimization process.

What could be the role of fuzzy techniques in MBPC?
A fuzzy model can be used to describe the non-linear behavior of the process. This model can be used to predict the process behavior in the future for a given control strategy.

Moreover, also the objective functions and constraints can be defined as fuzzy goals and constraints and the control problem is then translated to a fuzzy decision making problem. This is beneficial when some of the goals or constraints cannot be described analytically or should be fulfilled partially depending on other requirements. Sometimes a relaxation of some of the goals and constraints is possible.

Finally, a fuzzy model can be used in an IMC structure contained in the MBPC strategy to reduce process-model mismatch and the influence of disturbances.

Control and prediction horizons

The future plant outputs for an a priori prediction horizon H_p are predicted using a model of the process. The predicted output values $\hat{y}(k+j), j = 1...H_p$ depend on the state of the process at time k (e.g. given by past inputs and outputs) and on the future control signals $u(k+j)$ over a certain a priori determined control horizon H_c, assuming that $H_c \leq H_p$ and $u(k+j)$ remains constant for $j = H_c, ..., H_{p-1}$, see Fig. 16.

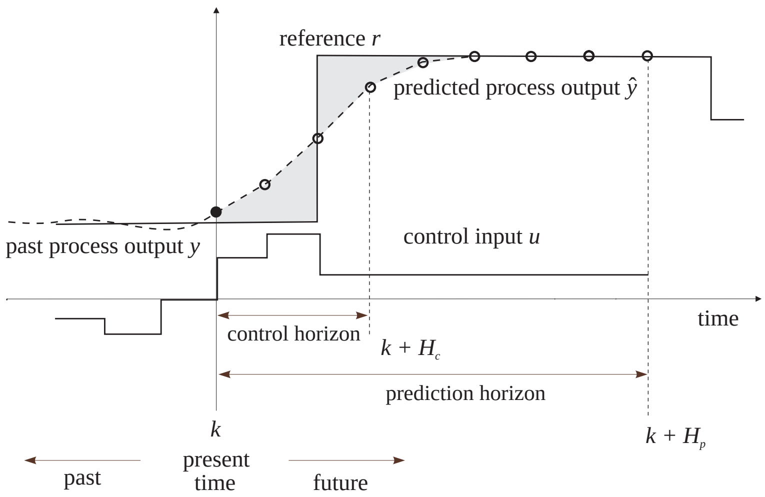


Figure 16: The basic principle of model-based predictive control.

H_c is usually chosen to be equal to the order or the model or lower which is allowed when fuzzy objective functions are utilized. A low value of H_c reduces considerably the computational costs in the optimization procedure.

H_p is usually related to the response time of the process (step response). For non-linear systems this is not clear because the time response can change considerably depending on the operating conditions. In a Takagi-Sugeno model this can be easily evaluated.

Objective function and reference trajectory

The sequence of future control signals is obtained by the optimization of an objective function which describes the goals of the control strategy. Constraints can be added as hard constraints separately to the objective function or should be included as soft constraints in the objective function.

In classical MBPC the objective function is usually expressed by the following quadratic function:

$$J(u) = \sum_{i=1}^{H_p} \alpha_i \left(w(k+i) - \hat{y}(k+i)\right)^2 + \sum_{i=1}^{H_c} \beta_i \Delta u(k+i-1)^2 \tag{40}$$

The first term accounts for minimizing the variance of the process output from the reference w, while the second term represents a penalty on the control effort. Sometimes u is used instead of Δu. The values α_i and β_i define the weighting of the different terms involved and indicate for instance that differences in the process output should be weighted more severely when time evolves. When the process is linear and no constraints are involved, an analytical solution to the control problem can be obtained. Otherwise numerical (usually iterative) optimization methods must be used.

In most cases the desired reference is known a priori and the system can react already before the change has actually been introduced. It is advisable to smooth the desired reference change to avoid sudden changes in the control actions at the cost of slower responses. This is obtained by introducing a reference generator.

Receding horizon principle and compensation of model-plant mismatch and disturbances

MBPC is an open loop control strategy, but by using the receding horizon principle a certain feedback is introduced. It is, however, advantageous to use some additional measures to decrease the influence of disturbances and plant-model mismatch. This can be obtained by using an IMC scheme together with the MBPC strategy.

The optimizer delivers the control signal according to the receding horizon principle. The model is used for prediction and the objective function introduced by the user is optimized by the optimizer given the predicted process output and the reference values delivered by the reference generator.

Note that the MBPC can be regarded as a generalization of IMC and that for $H_c = H_p = 1$, $\alpha_i = 1$ and $\beta_i = 0$ and requiring $y(k+1) = r(k+1)$ an inverse controller is obtained and the MBPC scheme becomes equivalent to the IMC scheme.

It is clear that a fuzzy model can be included in this scheme for predicting the output in the IMC scheme as well as for providing the future outputs for the optimizer (MBPC strategy).

We have to keep in mind that using fuzzy models for predicting the output of a non-linear process means that a non-convex optimization problem has to be solved. This is a time-consuming procedure which requires significant computer power and most of the optimization methods do not guarantee a global optimum.

This hampers generally the application of MBPC for non-linear systems. However, this is a general problem not specifically related to a fuzzy approach.

When the number of linguistic terms is, however, small and we allow a restricted number of magnitudes of the control signal, it is possible to use standard techniques known from operations research such as dynamic programming and the branch-and-bound method. These techniques have been applied successfully in fuzzy MBPC (Sousa, et al., 1997).

FUZZY DECISION MAKING IN CONTROL

A different approach to fuzzy MBPC is to use fuzzy sets for defining fuzzy goals and constraints.

In most design procedures one assumes that goals and constraints are deterministic and well-defined. Although goals and constraints are expressed in an analytical way in an objective function, the outcome of the optimization often results in changes of the parameters of the objective function when the designer is confronted with the results obtained, due to the high cost or the high energy consumption related to the solution. In many cases some of the goals or constraints can be relaxed, partly because the designer wants initially to be on the safe side when designing the control system.

Moreover, next to strict requirements and constraints, in many cases some desirable objectives play also an important role, but they do not have the same importance as the strict goals and constraints. They play a role in decision making when a number of alternatives should be compared which all satisfy the goals and constraints. In many cases a kind of tolerance is allowed with a preference for a certain value. It is clear that the concept of fuzzy sets fits well to this approach.

Moreover, when a system is non-linear, the advantages of having a quadratic cost function disappear. It is not longer possible to derive an analytical solution for the MBPC problem.

In fuzzy decision making, fuzzy goals and constraints are defined. It has been shown by Bellman and Zadeh (1970) that fuzzy goals and fuzzy constraints can be treated equally in fuzzy decision making.

Multiple goals and constraints can be defined in the same set or in different sets. All criteria can be defined in the same multidimensional set, which allows the use of aggregation operators between the different fuzzy criteria. Different accumulation operators have been considered between goals, between constraints and between goals and constraints. The predictive control problem is a multistage decision making problem which requires the definition of fuzzy criteria for each time step. The two main problems which emerge are the proper choice of the fuzzy goals and fuzzy constraints and the choice of the accumulation operators to combine these criteria.

Some standard membership functions can be selected for the requirements often met, such as a fast response without overshoot. Also hard constraints can be included requiring at the definition of a membership function that the support is restricted to the allowable region of the variable and the transient between the kernel and support is quite steep.

It is clear from this figure that only very few parameters are to be chosen, the K, H and S values, which have a physical meaning and are well suited to be chosen by the designer of the control system. It has been shown that the choice of a t-norm operator is most straight forward in satisfying all the goals and constraints and is compulsory in order not to violate hard constraints.

The solution of the optimization problem is of vital importance in any type of predictive control, especially because the solution should be obtained in real-time and the problem is non-convex.

It is in most cases sufficient to find a reasonably good solution as close as possible to the optimal one within the sampling period required for the dynamic system. After discretization of the possible control actions the branch-and-bound method provides a reasonable solution for the discrete optimization problem. The method has been applied to a heating, ventilating and air-condition problem in a real-time environment (Sousa, et al., 1997). Also genetic algorithms have been applied successfully for this kind of optimization problems. It is our expectation that the use of fuzzy goals and constraints methods will become increasingly popular in process control as long as the optimization problem can be kept sufficiently small.

MIMO ASPECTS OF FUZZY CONTROL

Especially in the process industry we are confronted with MIMO systems with many interactions which means that it is often difficult for human operators to express their control strategies when several control inputs should be manipulated simultaneously.

A MIMO system can generally be described by a fuzzy inference system, with rules described by:

R_i: If x_1 is A_{1i} and x_2 is A_{2i} ... and x_p is A_{pi} then y_1 is B_{1i} ... and y_q is B_{qi} for $i = 1, 2...k$

Due to the function approximation approach followed, it is, however, difficult to partition the input space because the non-linearities of the different outputs have different dynamic variations.

By decomposing the MIMO system in q MISO systems the number of membership functions and rules are reduced considerably.

We still have, however, the problem that the complexity exponentially grows with the number of inputs; even for a simple SISO system the number of rules can already grow extensively when the order of the system is relatively high. In that case for a model with input vector

$$x(k) = [y(k), y(k-1)...y(k-n), u(k), u(k-1)...u(k-n)] \tag{41}$$

the number of rules k becomes

$$k = \prod_{i=1}^{2n} N_i \tag{42}$$

where N_i is the number of linguistic terms of the i^{th} antecedent variable. For $N_i = 5$, for $i = 1...2n$ and $n = 2$ the number of rules becomes already: $5^4 = 625$.

For a MIMO system this number increases dramatically. For a system with m inputs and l outputs the number of rules becomes, when all single systems are of n^{th} order,

$$k = \prod_{i=1}^{(m+l)n} N_i \tag{43}$$

Thus, even for a 2×2 MIMO system of the second order with $N_i = 3$ the number of rules is already: $3^8 = 6561$.

It is clear from these examples that we have to look for a compact description, preferably a state space model, and that a decomposition of a MIMO system could considerably reduce the number of linguistic rules.

Several approaches have been proposed in order to reduce the number of rules, either directly by reducing the number of state variables and linguistic variables or indirectly by changing the structure (aggregation, temporal or spatial decomposition).

To the aggregation type of approach belongs the numeric fusion of state variables. Two states, e.g., error signal and error derivative signal, are first added with a certain weighting and the resulting signal is fed to the rule base. The fusing has to be very well justified.

Removing less important rules is also a straight forward way to reduce the number of rules. This applies to rules which are either physically not realisable or have small firing probability. The reduction of membership functions and related to this also the reduction of rules can also be obtained using similarity measures (Setnes, et al., 1998). The representation of a system as a chained hierarchical structure is based on the usage of a multilayer control structure where the first layer consists of the ones related to the most important state variables. The output of this MISO system is fed to the second layer and is combined with the second most important state variables, and so on. The number of rules in this case is a linear instead of an exponential function of the number of state variables. The reduction would be maximal if each layer contains only two state variables as input. When we compare the number of rules of this approach for the simple example of the SISO system with $N_i = 5$ we get $3 \times 5^2 = 75$ rules instead of 625 rules.

Another possibility is to divide the system into a number of different local domains each described by their own subrule bases. The consequent part of a meta rule base specifies the different local domains. As a result, only one subrule base may be active at a time and the number of rules is much smaller.

A very efficient method to define controllers for a MIMO system is to decouple the system first before applying a control structure. When the system can be fully decoupled a number of SISO controllers can be designed, each responsible for the control of one output variable. Research is going on to develop a unified approach for decoupling a fuzzy MIMO system, where also partly decoupling is possible. The result of this research will be very useful to design fuzzy controllers for MIMO systems in the near future.

REFERENCES

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

Babuška, R. and H.B. Verbruggen (1996). An overview of fuzzy modeling for control. *Control Engineering Practice 4*(11), 1593–1606.

Babuška, R. and H.B. Verbruggen (1997a). Constructing fuzzy models by product space clustering. H. Hellendoorn and D. Driankov (Eds.), *Fuzzy Model Identification: Selected Approaches*, pp. 53–90. Berlin, Germany: Springer.

Babuška, R. and H.B. Verbruggen (1997b). Fuzzy set methods for local modeling and identification. R. Murray-Smith and T. A. Johansen (Eds.), *Multiple Model Approaches to Nonlinear Modeling and Control*, pp. 75–100. London, UK: Taylor & Francis.

Bellman, R.E. and L.A. Zadeh (1970). Decision making in a fuzzy environment. *Management Science 17*(4), 141–164.

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.

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

Garcia, C.E. and M. Morari (1982). Internal model control. 1. a unifying review and some new results. *Ind. Eng. Chem. Process Des. Dev. 21*, 308–323.

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

Holmblad, L.P. and J.-J. Østergaard (1982). Control of a cement kiln by fuzzy logic. M.M. Gupta and E. Sanchez (Eds.), *Fuzzy Information and Decision Processes*, pp. 389–399. North-Holland.

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.

Jang, J.-S.R., C.-T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing; a Computational Approach to Learning and Machine Intelligence*. Upper Sadle River: 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.

Lin, C.T. (1994). *Neural Fuzzy Control Systems with Structure and Parameter Learning*. Singapore: World Scientific.

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.

Nakamori, Y. and M. Ryoke (1994). Identification of fuzzy prediction models through hyperellipsoidal clustering. *IEEE Trans. Systems, Man and Cybernetics 24*(8), 1153–73.

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

Pedrycz, W. (1984). An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems 13*, 153–167.

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

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

Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica 29*, 1251–1274.

Setnes, M., R. Babuška and H.B. Verbruggen (1998). Complexity reduction in fuzzy modeling. *IMACS Journal Mathematics and Computers in Simulation 46*, 507–516.

Setnes, M., R. Fleer, P. Bruijn and H.R. van Nauta Lemke (1997). Implementing fuzzy control in the manufacturing of washing powders. *Proceedings EUFIT'97*, Aachen, Germany.

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

Smith, S.M., B.R. Nokleby and D.J. Comer (1994). A computational approach to fuzzy logic controller design and analysis using cell state space methods. Kandel and Langholz (Eds.), *Fuzzy Control Systems*, pp. 398–427. Boca-Raton, Fl.: CRC Press.

Sousa, J.M., R. Babuška and H.B. Verbruggen (1997). Fuzzy predictive control applied to an air-conditioning system. *Control Engineering Practice 5*(10), 1395–1406.

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, H., S. Uejima and K. Asai (1982). Linear regression analysis with fuzzy model. *IEEE Trans. Systems, Man & Cybernetics 12*(6), 903–907.

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.

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

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

Yoshinari, Y., W. Pedrycz and K. Hirota (1993). Construction of fuzzy models through clustering techniques. *Fuzzy Sets and Systems 54*, 157–165.

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

Zadeh, L.A. (1975). Calculus of fuzzy restrictions. 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.

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