FUZZY CONTROL

Fuzzy control is a control approach which is based on the concept of fuzzy sets and fuzzy logic invented by Lotfi Zadeh in 1965 (1). Fuzzy sets are noncrisp or nonsharp sets or numbers, and fuzzy logic is a logic which deals with implications or IF THEN statements using noncrisp truth values. Fuzzy control deals with IF THEN statements or IF THEN rules, respectively, but in the sense of control commands like "IF temperature is LOW THEN change current of heater by a POSITIVE HIGH value." In this rule LOW and POSITIVE HIGH are fuzzy terms which are not sharply described. With the help of rules like that, one can formulate the knowledge of an operator in a complex plant with the aim to introduce an automatic control of the plant or of parts of it. Another option is to build up an advisory system by means of a set of fuzzy rules that supports the human operator making decisions. Fuzzy control is not only useful when human operators come into play but also in existing automatic control loops. Here, the fuzzy controller is a nonlinear control element that is able to improve control performance and robustness of a plant. In automatic control it is often required to have a process model available for compensation of nonlinear system's behavior and a corresponding feed forward control. For complex systems or plants it is therefore of advantage to use fuzzy system plant models in order to simplify both the identification and the control task. The following article deals with common fuzzy control techniques seen both from the system's and the controller's point of view. A special part is attended to the nonlinear nature of fuzzy control. Aspects of heuristic and model based fuzzy control are dealt with and the main points of supervisory and adaptive control are discussed.

Fuzzy control in the form of set of IF-THEN fuzzy rules was initiated by E. H. Mamdani when he started an investigation of fuzzy set theory–based algorithms for the control of a simple dynamic plant (2). Østergaard reported a fuzzy control application of a heat exchanger (3), and in 1982 Holmblad and Østergaard presented a cement kiln fuzzy controller (4). However, mainly due to the attention that Japan's industry paid to the new control technology, it was not until the late 1980s that fuzzy control became more and more accepted in industry. The commonly used technique in industrial process control is the Proportional-Integral-Differential (PID) controller, and it is used in a variety of different control schemes (e.g., adaptive, gain scheduling, and supervisory control architectures).

Today, processes and plants under control are so complex that PID controllers are not sufficient even though augmented with additional adaptive, gain scheduling, and supervisory algorithms. Although there is a large number of methods and theories (5) to cope with sufficiently complex control problems in the automation, robotics, consumer and industrial electronics, car, aircraft, and ship-building industries, the restrictions for applying these methods are either too strong or too complicated to be applied in a practically efficient and inexpensive manner. Therefore, control engineers are in a need of simpler process and plant models and controller design methods far removed from the sophisticated mathematical models available and their underlying rigorous assumptions. These simpler design methods should provide good performance characteristics, and they should be robust enough with regard to disturbances, parameter uncertainties, and unmodeled structural properties of the process under control.

In connection with traditional control techniques, fuzzy control provides a variety of design methods that can cope with modern control problems. There are three main aspects of fuzzy controllers that go beyond the conventional controllers designed via traditional control methods:

1. The use of IF-THEN rules.
2. The universal approximation property.
3. The property of dealing with vague (fuzzy) values.

The first aspect concerns the human operator's knowledge and its heuristic experience for controlling a plant. This knowledge is formulated in terms of IF-THEN fuzzy rules. In the same way, the plant's behavior can also be expressed by a set of IF-THEN fuzzy rules. The major problem is to identify the fuzzy rules and the regarding parameters such that the operator's control actions and the systems's response are sufficiently well described (6–8). Identification of this type of fuzzy rules can be done in two ways:

1. Knowledge acquisition via the use of interviewing techniques from the area of knowledge-based expert systems. This type of identification has been applied successfully to the control of Single Input/Single Output

(SISO) plants and processes, but is difficult to apply and verify for Multi Input/Multi Output (MIMO) control problems.

2. Black box type of identification via the use of clustering, neural nets, and genetic algorithm–based techniques.

In the latter approach one distinguishes between structure identification and parameter identification.

Structure identification requires structural a priori knowledge about the system to be controlled (e.g., whether the system is assumed to be linear, and what the order of the system might be). If one has to identify a plant with only little structural knowledge, one has to use algorithms that learn from data. The result of structure identification is a set of fuzzy rules.

Parameter identification deals with a proper parametrization, scaling, and normalization of physical signals. Parameter identification is a comparetively simple task and can be done by classical methods (e.g., Linear Quadratic (LQ) methods and related techniques).

The second aspect, the universal approximation property, means that a fuzzy system with product-based rule firing, centroid defuzzification, and Gaussian membership functions can approximate any real continuous function on a compact set to arbitrary accuracy (9–11). However, in most cases the approximation of a finite state space by a finite number of fuzzy rules is required while using triangular or trapezoidal membership functions. In this case certain approximation errors must be accepted.

The approximation property is due to the overlap of the membership functions from the IF parts of the set of fuzzy rules. Because of this overlap, every rule is influenced by its neighboring rules. The result is that every point in state space is approximated by a subset of fuzzy rules.

The third aspect considers control tasks where the controller inputs are fuzzy values instead of being crisp variables. In contrast to classical controllers, fuzzy controllers (FC) can also deal with fuzzy values, and even the mixture of crisp and fuzzy values becomes possible. Fuzzy values are qualitative "numbers" obtained from different sources. One particular source is a qualitative statement of a human operator while controlling a plant, like "temperature is high." Another source may originate from a sensor that provides information about the intensity of a physical signal with in a certain interval. Here, the intensity or distribution of the signal with respect to this interval is expressed by a membership function.

This article is arranged as follows: The following section deals with fuzzy control techniques, including the design goal, the definition of a fuzzy region, and the most important FC techniques for systems and controllers. Then the article deals with the fuzzy controller as a nonlinear transfer element while the computational structure of a fuzzy controller, its transfer characteristics, and its nonlinearity are discussed. Different heuristic and model-based control strategies, such as the Mamdani controller, the sliding mode fuzzy controller, the cell mapping control strategy, and the Takagi Sugeno control strategy, are discussed, a short overview of supervisory control is provided; and finally the main aspects of adaptive fuzzy control are discussed.

FUZZY CONTROL TECHNIQUES

FC techniques can be divided into experiential (heuristic) and model-based techniques. The choice for a special FC technique depends on how the system to be controlled is described. Figure 1 shows the most important FC techniques dealing with systems and controllers.

The following subsection deals with the design goal of fuzzy control. In a subsequent subsection the fuzzy region is defined. Finally, the individual FC techniques for systems and controllers are outlined.

The Design Goal

The objective of the design in fuzzy control can be stated as follows:

1. *Stabilization.* In stabilization control problems, a fuzzy controller, called a stabilizer, or regulator, is to be designed so that the state vector of the closed-loop system will be stabilized around a point (operating point, or a setpoint) of the state space. The asymptotic stabilization control problem is to find a control law in terms of a set of fuzzy rules such that, starting anywhere in a region around the setpoint $\boldsymbol{x}_d$, the state vector $\boldsymbol{x}$ of the closed-loop system goes to the setpoint $\boldsymbol{x}_d$, as t goes to infinity.
2. *Tracking.* In tracking control problems, a fuzzy controller is to be designed so that the closed-loop system output follows a given time-varying trajectory. The asymptotic tracking problem is to find a control law in terms of a set of fuzzy rules such that starting from any initial state $\boldsymbol{x}^0$ in a region around $\boldsymbol{x}_d(t)$, the tracking error $\boldsymbol{x}(t) - \boldsymbol{x}_d(t)$ tends to $\boldsymbol{0}$ while the whole state vector remains bounded. Let us stress here that perfect tracking (i.e., when the initial states imply zero tracking error) is not possible. Therefore, the design objective of having asymptotic tracking cannot be achieved. In this case, one should aim at bounded-error tracking, with small tracking errors to be obtained for trajectories of particular interest.

From a theoretical point of view, there is a relationship between the stabilization and the tracking control problems. Stabilization can be regarded as a special case of tracking where the desired trajectory is a constant. On the other hand, if, for example, we have to design a tracker for the open-loop system

$$\ddot{y} + f(\dot{y}, y, u) = 0 \tag{1}$$

so that $e(t) = y(t) - y_d(t)$ tends to zero, the problem is equivalent to the asymptotic stabilization of the system,

$$\ddot{e} + f(\dot{e}, e, u, y_d, \dot{y}_d, \ddot{y}_d) = 0 \tag{2}$$

its state vector components being e and $\dot{e}$. Thus the tracker design problem can be solved if one designs a regulator for the latter nonautonomous open-loop system.

Performance. In linear control, the desired behavior of the closed-loop system can be systematically specified in exact quantitative terms. For example, the specifications of the desired behavior can be formulated in the time domain in terms of rise time and settling time, overshoot and undershoot, etc. Thus, for this type of control, one first postulates the quanti-

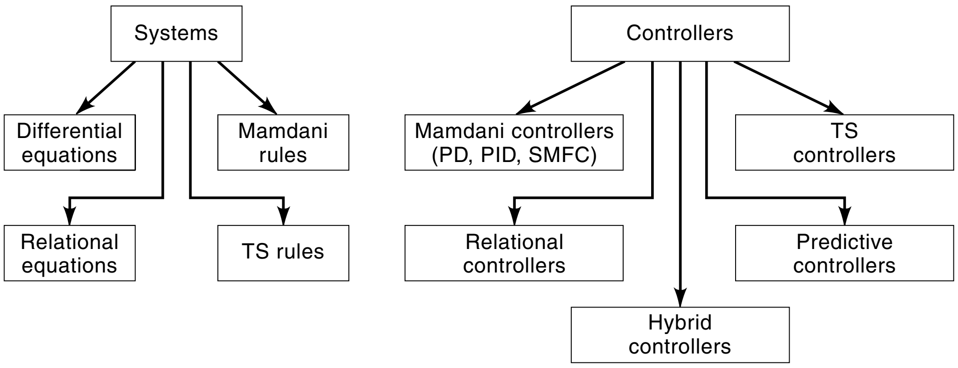

Figure 1. FC techniques. Collection of fuzzy control techniques for systems and controllers. Mamdani controllers can, e.g., be applied to systems described by differential equations.

tative specifications of the desired behavior of the closed-loop system and then designs a controller that meets these specifications (for example, by choosing the poles of the closed-loop system appropriately).

As observed in Ref. 12, such systematic specifications of the desired behavior of nonlinear closed-loop systems, except for those that can be approximated by linear systems, are not obvious at all because the response of a nonlinear system (open or closed loop) to one input vector does not reflect its response to another input vector. Furthermore, a frequency domain description of the behavior of the system is not possible either.

The consequence is that in specifying the desired behavior of a nonlinear closed-loop system, one employs some qualitative specifications of performance, including stability, accuracy and response speed, and robustness.

Stability. Stability must be guaranteed for the model used for design (the nominal model) either in a local or in a global sense. The regions of stability and convergence are also of interest.

One should, however, keep in mind that stability does not imply the ability to withstand persistent disturbances of even small magnitude. This is so since the stability of a nonlinear system is defined with respect to initial conditions, and only temporary disturbances may be translated as initial conditions. Thus stability of a nonlinear system is different from stability of a linear system. In the case of a linear system, stability always implies the ability to withstand bounded disturbances when, of course, the system stays in its linear range of operation. The effects of persistent disturbances on the behavior of a nonlinear system are addressed by the notion of robustness.

Accuracy and Response Speed. Accuracy and response speed must be considered for some desired trajectories in the region of operation. For some classes of systems, appropriate design methods can guarantee consistent tracking accuracy independent of the desired trajectory, as is the case in sliding mode control and related control methods.

Robustness. Robustness reflects the sensitivity of the closed-loop system to effects that are neglected in the nominal model used for design. These effects can be disturbances, measurement noise, unmodeled dynamics, etc. The closed-loop system should be insensitive to these neglected effects in the sense that they should not negatively affect its stability.

We want to stress here that the aforementioned specifications of desired behavior are in conflict with each other to some extent, and a good control system can be designed only based on tradeoffs in terms of robustness versus performance, cost versus performance, etc.

Fuzzy Regions

In fuzzy control, a crisp state vector $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ is a state vector the values of which are defined on the closed interval (the domain) X of reals. A crisp control input vector $\boldsymbol{u} = (u_1, \ldots, u_n)^T$ is a control input vector the values of which are defined on the closed interval (the domain) U of reals. The set of fuzzy values of a component x_i is called the term set of x_i denoted as $\boldsymbol{TX_i} = \{LX_{i1}, \ldots, LX_{im_i}\}$ (e.g., NB, NM, NS, Z, PS, PM, PB with N negative, P positive, S small, M medium, B big). LX_{ij} is defined by a membership function $\int_X \mu_{X_{ij}}(x)/x$. The term set of u_i is likewise denoted as $\boldsymbol{TU_i} = \{LU_{i1}, \ldots, LU_{ik_i}\}$. LU_{ij} is defined by a membership function $\int_U \mu_{U_{ij}}(u)/u$. An arbitrary fuzzy value from $\boldsymbol{TX_i}$ is denoted as LX_i that can be any one of $LX_{i1}, \ldots, LX_{im_i}$. An arbitrary fuzzy value from $\boldsymbol{TU_i}$ will be denoted as LU_i and can be any one of $LU_{i1}, \ldots, LU_{ik_i}$.

A fuzzy state vector $\boldsymbol{LX} = (LX_1, \ldots, LX_n)^T$ denotes a vector of fuzzy values. Each component $x_1, \ldots, x_n$ of the state vector $\boldsymbol{x}$ takes a corresponding fuzzy value $LX_1, \ldots, LX_n$, where $LX_i \in \boldsymbol{TX_i}$. A fuzzy region $\boldsymbol{LX^i} = (LX_1^i, \ldots, LX_n^i)^T$ is defined as a fuzzy state vector for which there exists a contiguous set of crisp state vectors $\{\boldsymbol{x^*}\}$, each crisp state vector satisfying the given fuzzy state vector $\boldsymbol{LX^i}$ to a certain degree different from 0. The fuzzy state space is defined as the set of all fuzzy regions LX^i.

Example Let $\boldsymbol{x} = (x_1, x_2)^T$, $\boldsymbol{TX_1} = \{LX_{11}, LX_{12}, LX_{13}\}$, and $\boldsymbol{TX_2} = \{LX_{21}, LX_{22}, LX_{23}\}$. Then the total number of different fuzzy state vectors is $\boldsymbol{M} = 9$ and the corresponding state vectors are

1. $\boldsymbol{LX^1} = (LX_{11}, LX_{21})^T$
2. $\boldsymbol{LX^2} = (LX_{11}, LX_{22})^T$
3. $\boldsymbol{LX^3} = (LX_{11}, LX_{23})^T$
4. $\boldsymbol{LX^4} = (LX_{12}, LX_{21})^T$
5. $\boldsymbol{LX^5} = (LX_{12}, LX_{22})^T$
6. $\boldsymbol{LX^6} = (LX_{12}, LX_{23})^T$
7. $\boldsymbol{LX^7} = (LX_{13}, LX_{21})^T$

8. $\boldsymbol{LX^8} = (LX_{13}, LX_{22})^T$
9. $\boldsymbol{LX^9} = (LX_{13}, LX_{23})^T$

FC Techniques for Systems and Controllers

In this subsection we deal with systems and controllers according to the scheme shown in Fig. 1. Given a model (heuristic or analytical) of the physical system to be controlled and the specifications of its desired behavior, design a feedback control law in the form of a set of fuzzy rules such that the closed-loop system exhibits the desired behavior.

The general control scheme is shown in Fig. 2. Here, we have the following notations:

$\boldsymbol{x}$ is the state vector (also controller input)
$\boldsymbol{x_d}$ is the desired state vector
$\boldsymbol{u}$ is the control input vector (also controller output)

where the vectors $\boldsymbol{x}$, $\boldsymbol{x_d}$, $\boldsymbol{u}$, are continuous functions of time. For simplicity, the output vector $\boldsymbol{y}$ is set to be equal to the state vector $\boldsymbol{x}$:

$$\boldsymbol{y} = \boldsymbol{x}$$

In the following we define two basic types of nonlinear control problems: namely, nonlinear regulation (stabilization) and nonlinear tracking (12). Then we will briefly discuss the specifications of desired behavior, such as performance, stability, and robustness, in the context of nonlinear control.

Stabilization and Tracking. In general, the tasks of a control system can be divided into two basic categories:

Heuristic System Models. When an analytical model of the plant is not available, the control design has to be carried out on the basis of qualitative modeling. This can be done either in terms of a set of Mamdani fuzzy rules or a fuzzy relation (6,13). A typical Mamdani rule of a continuous first-order system is

$$R_{Si}: \quad \text{IF} \quad x \,\text{is PS} \quad \text{AND} \quad u \,\text{is NB} \quad \text{THEN} \quad \dot{x} \,\text{is NS} \tag{3}$$

and for a discrete system

$$R_{Si}: \quad \text{IF} \quad x(k) \,\text{is PS} \quad \text{AND} \quad u(k) \,\text{is NB} \quad \text{THEN} \quad x(k+1) \,\text{is NS}$$

A typical fuzzy relational equation of a discrete first-order system is

$$X(k+1) = X(k) \circ U(k) \circ S \tag{4}$$

relating the state at time $k + 1$ to the state and control input at time k. $\boldsymbol{S}$ is the fuzzy relation. $\circ$ denotes the relational composition (e.g., max-min composition). A fuzzy relation is another representation of a Mamdani fuzzy system.

Analytical Systems Models. If an analytical model of the plant is available, then the system's behavior can be described by a set of differential equations or by a set of so-called Takagi Sugeno fuzzy rules (TS rules) (8). A typical differential equation of an open-loop system is

$$\dot{\boldsymbol{x}} = A \cdot \boldsymbol{x} + B \cdot \boldsymbol{u} \tag{5}$$

On the other hand, a TS fuzzy rule consists of a fuzzy antecedent part and a consequent part consisting of an analytical equation. A typical TS rule for a first-order system is

$$R_{Si}: \quad \text{IF} \quad x = LX^i \quad \text{THEN} \quad \dot{x} = A_i \cdot x + B_i \cdot u \tag{6}$$

where LX^i is the ith fuzzy region for x, and A_i and B_i are parameters corresponding to that region.

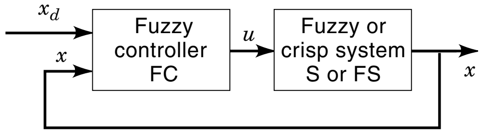

Figure 2. General control structure.

Fuzzy controllers can be classified as follows:

Mamdani Controller. A Mamdani controller works in the following way:

1. A crisp value is scaled into a normalized domain.
2. The normalized value is fuzzified with respect to the input fuzzy sets.
3. By means of a set of fuzzy rules, a fuzzy output value is provided.
4. The fuzzy output is defuzzified with the help of an appropriate defuzzification method (center of gravity, height method, etc.).
5. The defuzzified value is denormalized into a physical domain.

A typical Mamdani controller is

$$R_{Ci}: \quad \text{IF} \quad x = LX^i \quad \text{THEN} \quad u = LU^i \tag{7}$$

where LU^i is the corresponding fuzzy value for the control variable.

Relational Controller. According to the description of the system in terms of a relational equation, a typical discrete fuzzy relational equation for a controller is

$$U(k) = X(k) \circ C \tag{8}$$

where X is the fuzzy state, U is the fuzzy control variable, and C is the fuzzy relation. A relational controller is another representation of a Mamdani Controller.

Takagi Sugeno Controller. A typical TS controller is

$$R_{Ci}: \quad \text{IF} \quad x = LX^i \quad \text{THEN} \quad u = K_i \cdot x \tag{9}$$

where LX^i is the ith fuzzy region for x, and K_i is the gain corresponding to that region.

Predictive Controller. A special way of predictive fuzzy control was introduced by Yasunobu (14) for automatic train operation. It includes control rules for the time k to predict the behavior of the system for the next timestep $k + 1$. By means of a performance index $J(k)$, which appears for a specific control action $u(k)$, different features like velocity, riding comfort, energy saving, and accuracy of a stop gap are evaluated. By means of going through the whole range of possible control actions $u(k)$, one obtains a range of corresponding performance indices $J(k)$ from which the control action $u(k + 1)$ with the highest performance index $J(k)$ is applied to the plant. A typical predictive control rule is

IF the performance index $J(k) = LJ^i$ is obtained
AND a control value $u(k)$ is chosen to be LU^i
THEN the control value to be applied to the plant for the next timestep $k + 1$
is chosen to be $u(k + 1) = LU^i$.

A formal description is

$$\text{IF} \quad J(k) = LJ^i \quad \text{AND} \quad u(k) = LU^i \quad \text{THEN} \quad u(k+1) = LU^i \tag{10}$$

A further relationship to model predictive control (15) can be found in Refs. 16 and 17.

Hybrid Controller. A hybrid controller is represented by a mixture of fuzzy controller and conventional controller. Fuzzy hybrid controllers are, e.g., applied for tuning conventional controllers and in adaptation schemes. Another application is the use of a nonlinear fuzzy mapping in nonlinear control tasks. A typical hybrid controller appears if the control law consists of a Mamdani controller C_{fuzz} and an analytical feedforward term C_{comp} that compensates (e.g., statical or dynamical forces in a mechanical system):

$$u = C_{\text{fuzz}}(\boldsymbol{x}, \boldsymbol{x_d}) + C_{\text{comp}}(\boldsymbol{x}) \tag{11}$$

where $\boldsymbol{x_d}$ is the desired vector. Further information can be found in Ref. 18.

THE FUZZY CONTROLLER AS A NONLINEAR TRANSFER ELEMENT

A fuzzy logic controller defines a control law in the form of a static nonlinear transfer element (TE) due to the nonlinear nature of the computations performed by a fuzzy controller. However, the control law of a fuzzy controller is not represented in an analytic form, but by a set of fuzzy rules. The antecedent of a fuzzy rule (IF part) describes a fuzzy region in the state space. Thus one effectively partitions an otherwise continuous state space by covering it with a finite number of fuzzy regions and, consequently, fuzzy rules. The consequent of a fuzzy rule (THEN part) specifies a control law applicable within the fuzzy region from the IF part of the same fuzzy rule. During control with a fuzzy controller, a point in the state space is affected to a different extent by the control laws associated with all the fuzzy regions to which this particular point in the state space belongs. By using the operations of aggregation and defuzzification, a specific control law for this particular point is determined. As the point moves in the state space, the control law changes smoothly. This implies that a fuzzy controller yields a smooth nonlinear control law despite the quantization of the state space in a finite number of fuzzy regions.

One goal of this section is to describe computation with a fuzzy controller and its formal description as a static nonlinear transfer element and thus provide the background knowledge needed for understanding control with a fuzzy controller. Furthermore, we show the relationship between conventional and rule-based transfer elements, thus establishing the compatibility between these two conceptually different, in terms of representation, types of transfer elements.

The Computational Structure of a Fuzzy Controller

A control law represented in the form of a fuzzy controller directly depends on the measurements of signals and is thus a static control law. This means that the fuzzy rule-based representation of a fuzzy controller does not include any dynamics, which makes a fuzzy controller a static transfer element, like a state controller. Furthermore, a fuzzy controller is, in general, a nonlinear static transfer element that is due to those computational steps of its computational structure that have nonlinear properties. In what follows we will describe the computational structure of a fuzzy controller by presenting the computational steps that it involves.

The computational structure of a fuzzy controller consists of a number of computational steps and is illustrated in Fig. 3:

1. Input scaling (normalization)
2. Fuzzification of controller-input variables
3. Inference (rule firing)
4. Defuzzification of controller-output variables
5. Output scaling (denormalization)

The state variables $x_1, x_2, \ldots, x_n$ (or $e, \dot{e}, \ldots, e^{(n-1)}$) that appear in the IF part of the fuzzy rules of a fuzzy controller are also called controller inputs. The control input variables $u_1, u_2, \ldots, u_m$ that appear in the THEN part of the fuzzy rules of a fuzzy controller are also called controller outputs. We will now consider each of the computational steps for the case of a multiple-input/single-output (MISO) fuzzy controller. The generalization to the case of multiple-input/multiple-output fuzzy controller, where there are m controller outputs $u_1, u_2, \ldots, u_m$ instead of a single controller output u, can easily be done.

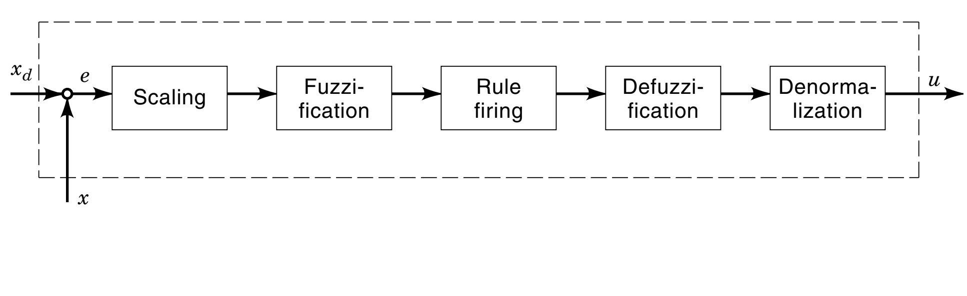

Figure 3. The computational structure of a fuzzy controller. The arrangement of the blocks correspond to the sequence of computation.

Input Scaling. There are two principal cases in the context of input scaling:

1. The membership functions defining the fuzzy values of the controller inputs and controller outputs are defined off-line on their actual physical domains. In this case the controller inputs and controller outputs are processed only using fuzzification, rule firing, and defuzzification. For example, this is the case of a Takagi-Sugeno fuzzy controller.
2. The membership functions defining the fuzzy values of controller inputs and controller outputs are defined off-line, on a common normalized domain. This means that the actual, crisp physical values of the controller inputs and controller outputs are mapped onto the same predetermined normalized domain. This mapping, called normalization, is done by appropriate normalization factors. Input scaling is then the multiplication of a physical, crisp controller input, with a normalization factor so that it is mapped onto the normalized domain. Output scaling is the multiplication of a normalized controller output with a denormalization factor so that it is mapped back onto the physical domain of the controller outputs.

The advantage of the second case is that fuzzification, rule firing, and defuzzification can be designed independent of the actual physical domains of the controller inputs and controller outputs.

To illustrate the notion of input scaling, let us consider, for example, the state vector $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)^T = (e, \dot{e}, \ldots, e^{(n-1)})^T$, where for each i, $e_i = x_i - x_{d_i}$. This vector of physical controller inputs is normalized with the help of a matrix N_e containing predetermined normalization factors for each component of e. The normalization is done as

$$e_N = N_e \cdot e \tag{12}$$

with

$$N_e = \begin{pmatrix} N_{e_1} & 0 & \ldots & 0 \\ 0 & N_{e_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & N_{e_k} \end{pmatrix} \tag{13}$$

where N_{e_i} are real numbers and the normalized domain for e is, say, $E_N = [-a, +a]$.

Example Let $e = (e_1, e_2)^T = (e, \dot{e})^T$ with

$$e = x - x_d \quad \text{and} \quad \dot{e} = \dot{x}_d - \dot{x}_d \tag{14}$$

Then input scaling of e into e_N and $\dot{e}$ into $\dot{e}_N$ yields

$$e_N = N_e \cdot e \quad \text{and} \quad \dot{e}_N = N_{\dot{e}} \cdot \dot{e} \tag{15}$$

where N_e and $N_{\dot{e}}$ are the normalization factors for e and $\dot{e}$, respectively.

In the context of a phase plane representation of the dynamic behavior of the controller inputs, the input scaling affects the angle of a line that divides the phase plane into two semiplanes (see Fig. 4).

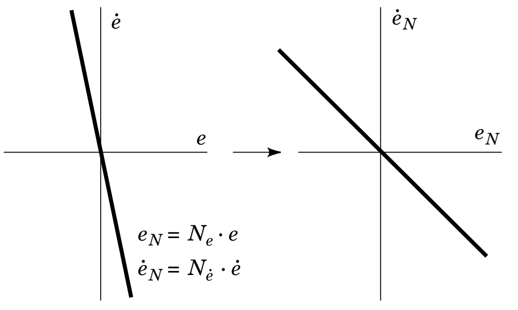

Figure 4. Normalization of the phase plane. Different normalization factors N_e and $N_{\dot{e}}$ correspond to different slopes of the line $N_e \cdot e + N_{\dot{e}} \cdot \dot{e} = 0$.

Furthermore, we can see how the supports of the membership functions defining the fuzzy values of e and $\dot{e}$ change because of the input scaling of these controller inputs (see Fig. 5).

In the next three subsections on fuzzification, rule firing, and defuzzification, we consider only the case when the fuzzy values of the controller inputs and controller outputs are defined on normalized domains (e.g., E_N and U_N), and in this case we will omit the lower index N from the notation of normalized domains and fuzzy and crisp values. In the subsection on denormalization we will use the lower index N to distinguish between normalized and nonnormalized fuzzy and crisp values.

Fuzzification. During fuzzification a crisp controller input $\boldsymbol{x}^*$ is assigned a degree of membership to the fuzzy region from the IF part of a fuzzy rule. Let $LE_1^i, \ldots, LE_n^i$ be some fuzzy values taken by the controller inputs $e_1, \ldots, e_n$ in the IF part of the $\boldsymbol{i}$th fuzzy rule R_C^i of a fuzzy controller; that is, these fuzzy values define the fuzzy region $\boldsymbol{LE^i} = (LE_1^i, \ldots, LE_n^i)^T$.

Each of the preceding fuzzy values, LE_k^i is defined by a membership function on the same (normalized) domain of error E. Thus the fuzzy value LE_k^i is given by the membership function $\int_E \mu_{LE_k^i}(e_k)/e_k$.

Let us consider now a particular normalized crisp controller input

$$\boldsymbol{e}^* = (e_1^*, \ldots, e_n^*)^T \tag{16}$$

from the normalized domain E. Each e_k^* is a normalized crisp

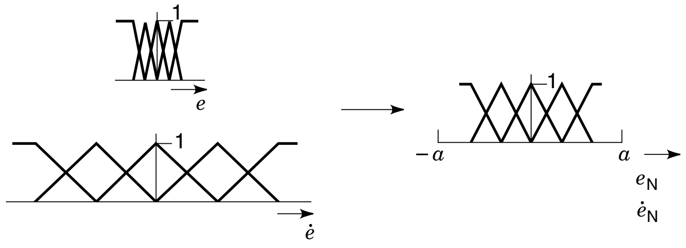

Figure 5. Change of the supports of the membership functions due to input scaling. Scaling normalizes different supports for e and $\dot{e}$ to a common support for e_N and $\ddot{e}_N$.

value obtained after the input scaling of the current physical controller input. The *fuzzification* of the crisp normalized controller input then consists of finding the membership degree of e_k^* in $\int_E \mu_{LE_k^i}(e_k)/e_k$. This is done for every element of $\boldsymbol{e}^*$.

Example Consider the fuzzy rule R_C^i given as

$$R_C^{\boldsymbol{i}}: \text{ IF } \boldsymbol{e} = (\mathrm{PS}_e, \mathrm{NM}_{\dot{e}}) \text{ THEN } u = \mathrm{PM}_u \quad (17)$$

where PS_e is the fuzzy value POSITIVE SMALL of the controller input e, $\mathrm{NM}_{\dot{e}}$ is the fuzzy value NEGATIVE MEDIUM of the second controller input $\dot{e}$, and PM_u is the fuzzy value NEGATIVE MEDIUM of the single controller output u. The membership functions representing these two fuzzy values are given in Fig. 6.

In this example we have $\boldsymbol{e} = (e, \dot{e})^T$ and thus the IF part of the preceding rule represents the fuzzy region $\boldsymbol{LE^i} = (\mathrm{PS}_e, \mathrm{NM}_{\dot{e}})^T$. Furthermore, let $e^* = a_1$ and $\dot{e}^* = a_2$ be the current normalized values of the physical controller inputs e^* and $\dot{e}^*$, respectively, as depicted in Fig. 6. Then from Fig. 6 we obtain the degrees of membership $\mu_{\mathrm{PS}_e}(a_1) = 0.3$ and $\mu_{\mathrm{NM}_{\dot{e}}}(a_2) = 0.65$.

Rule Firing. For a multi-input/single-output fuzzy controller, the ***i***th fuzzy rule of the set of fuzzy rules has the form

$$R_C^{\boldsymbol{i}}: \text{ IF } \boldsymbol{e} = \boldsymbol{LE^i} \text{ THEN } u = LU^{\boldsymbol{i}} \quad (18)$$

where the fuzzy region $\boldsymbol{LE^i}$ from the IF part of the preceding fuzzy rule is given as $\boldsymbol{LE^i} = (LE_1^i, LE_2^i, \ldots, LE_n^i)^T$. Also, LE_k^i denotes the fuzzy value of the kth normalized controller input e_k that belongs to the term set of e_k given as $\boldsymbol{TE_k} = \{LE_{k1}, LU_{k2}, \ldots, LU_{kn}\}$. Furthermore, LU^i denotes an arbitrary fuzzy value taken by the normalized controller output u, and this fuzzy value belongs to the term set $\boldsymbol{TU}$ of u; that is, $\boldsymbol{TU} = \{LU_1, LU_2, \ldots, LU_n\}$.

Let the membership functions defining the fuzzy values from $\boldsymbol{LE^i}$ and LU_i be denoted by $\int_E \mu_{LE_k^i}(e_k)/e_k$ ($k = 1, 2, \ldots, n$) and $\int_U \mu_{LU^i}^i(u)/u$, respectively. The membership function $\int_U \mu_{LU^i}(u)/u$ is defined on the normalized domain U, and the membership functions $\int_E \mu_{LE_k^i}(e_k)/e_k$ are defined on the normalized domain E.

Given a controller input vector $\boldsymbol{e}^*$ consisting of the normalized crisp values $e_1^*, \ldots, e_n^*$, first the degree of satisfaction $\mu^i(\boldsymbol{e}^*)$ of the fuzzy region $\boldsymbol{LE^i}$ is computed as

$$\mu^{\boldsymbol{i}}(\boldsymbol{e}^*) = \min\left(\mu_{LE_1^i}(e_1^*), \mu_{LE_2^i}(e_2^*), \ldots, \mu_{LE_n^i}(e_n^*)\right) \quad (19)$$

Second, given the degree of satisfaction $\mu^i(\boldsymbol{e}^*)$ of the fuzzy region $\boldsymbol{LE^i}$, the normalized controller output of the ***i***th fuzzy rule is computed as

$$CLU^{\boldsymbol{i}} = \int_U \mu_{CLU^{\boldsymbol{i}}}(u)/u = \min\left(\mu^{\boldsymbol{i}}(\boldsymbol{e}^*), \int_U \mu_{LU^{\boldsymbol{i}}}(u)/u\right) \quad (20)$$

Thus the controller output of the ***i***th fuzzy rule is modified by the degree of satisfaction $\mu^i(\boldsymbol{e}^*)$ of the fuzzy region $\boldsymbol{LE^i}$ and hence defined as the fuzzy subset $CLU^i = \int_U \mu_{CLU^i}(u)/u$ of $\int_U \mu_{LU^i}(u)/u$. That is,

$$\forall u{:}\mu_{CLU^{\boldsymbol{i}}}(u) = \begin{cases} \mu_{LU^{\boldsymbol{i}}}(u) & \text{if } \mu_{LU^{\boldsymbol{i}}}(u) \le \mu^{\boldsymbol{i}}, \\ \mu_{LU^{\boldsymbol{i}}}(u) = \mu^{\boldsymbol{i}}(\boldsymbol{e}^*) & \text{otherwise} \end{cases} \quad (21)$$

The fuzzy set $CLU^i = \int_U \mu_{CLU^i}(u)/u$ is called the clipped controller output. It represents the modified version of the controller output $\int_U \mu_{LU^i}(u)/u$ from the ***i***th fuzzy rule given certain crisp controller input $e_1^*, \ldots, e_n^*$.

In the final stage of rule firing, the clipped controller outputs of all fuzzy rules are combined in a global controller output via aggregation:

$$\forall u{:}\mu_{CU}(u) = \max(\mu_{CLU^1}, \ldots, \mu_{CLU^{\boldsymbol{M}}}) \quad (22)$$

where $CU = \int_U \mu_{CU}(u)/u$ is the fuzzy set defining the fuzzy value of the global controller output. The type of rule firing described here is called max-min composition. Another type of composition can be found in Ref. 40.

Defuzzification. The result of rule firing is a fuzzy set CU with a membership function $\int_U \mu_{CU}(u)/u$, as defined in Eq. (22). The purpose of defuzzification is to obtain a scalar value u from μ_{CU}. The scalar value u is called a defuzzified controller output. This is done by the center of gravity method as follows.

In the continuous case we have

$$u = \frac{\int_U \mu_{CU}(u) \cdot u\, du}{\int_U \mu_{CU}(u)\, du} \quad (23)$$

and for the discrete case

$$u = \frac{\sum_U \mu_{CU}(u) \cdot u\, du}{\sum_U \mu_{CU}(u)} \quad (24)$$

Example Consider the normalized domain $U = \{1, 2, \ldots, 8\}$ and let the fuzzy set CU be given as

$$CU = \{0.5/3, 0.8/4, 1/5, 0.5/6, 0.2/7\} \quad (25)$$

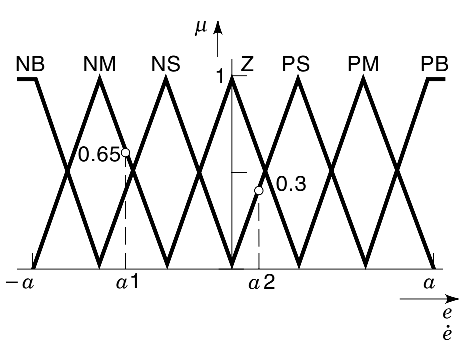

Figure 6. Fuzzification of crisp values e^* and $\dot{e}^*$. Fuzzification of $e^* = a1$ with respect to a fuzzy set NM is obtained by finding the crosspoint between $a1$ and the corresponding membership function NM.

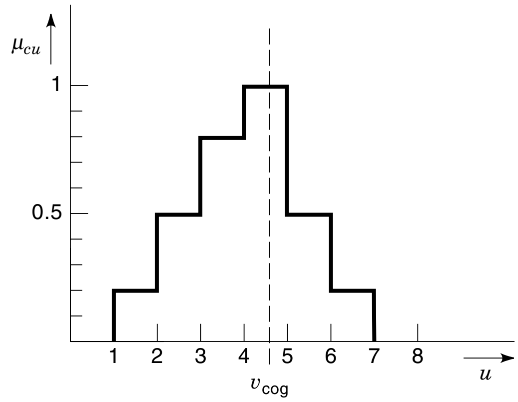

Figure 7. Defuzzification of a fuzzy controller output. Defuzzification of a fuzzy set μ_{CU} is obtained by computing the u-coordinate of the center of gravity of the membership function.

Then the defuzzified controller output u is computed as (see also Fig. 7)

$$u = \frac{0.5 \cdot 3 + 0.8 \cdot 4 + 1 \cdot 5 + 0.5 \cdot 6 + 0.2 \cdot 7}{0.5 + 0.8 + 1 + 0.5 + 0.2} = 4.7 \quad (26)$$

Denormalization. In the denormalization procedure the defuzzified normalized controller output u_N is denormalized with the help of an off-line predetermined scalar denormalization factor N_u^{-1}, which is the inverse of the normalization factor N_u. Let the normalization of the controller output be performed as

$$u_N = N_u \cdot u \quad (27)$$

Then the denormalization of u_N is

$$u = N_u^{-1} \cdot u_N \quad (28)$$

The choice of N_u essentially determines, together with the rest of the scaling factors, the stability of the system to be controlled.

In the case of Takagi Sugeno fuzzy controllers, the preceding computational steps are performed on the actual physical domains of the controller inputs and outputs. Thus the computational steps of normalization and denormalization are not involved in the computational structure of a Takagi Sugeno fuzzy controller, which, in turn, eliminates the need for input and output scaling factors.

The Transfer Characteristics

The way to obtain a specific input output transfer characteristics shows the following example (SISO):

1. Suppose there is a set of rules like

$$R1: \text{IF}\, x = \text{NB THEN}\, y = \text{PB}$$
$$R2: \text{IF}\, x = \text{NS THEN}\, y = \text{PS}$$
$$R3: \text{IF}\, x = \text{Z THEN}\, y = \text{Z}$$
$$R4: \text{IF}\, x = \text{PS THEN}\, y = \text{NS}$$
$$R5: \text{IF}\, x = \text{PB THEN}\, y = \text{NB}$$

where

x = input; y = output;
N = negative; P = positive; Z = zero; S = small; B = big

2. Shape and location of the corresponding membership functions are chosen so that they always overlap at the degree of membership $\mu_X = 0.5$ (see Fig. 8).
3. For the specific crisp controller input x_{in} one obtains the degrees of membership $\mu_{X_{NS}}(x_{in}) > 0$ and $\mu_{X_Z}(x_{in}) > 0$, where the remaining degrees of membership $\mu_{X_{NB}}(x_{in})$, $\mu_{X_{PS}}(x_{in})$, and $\mu_{X_{PB}}(x_{in})$ are equal to zero. Hence, only rules R2 and R3 fire. The controller output set is computed by cutting the output set $\mu_{Y_{PS}}$ at the level of $\mu_{X_{NS}}(x_{in})$ and μ_{Y_Z} at $\mu_{X_Z}(x_{in})$. The resulting output membership function μ_Y takes every rule into account, performing the union of the resulting output membership function $\mu_{Y_{Ri}}$ of each rule Ri (i = 1, . . ., 5) which means the maximum operation between them.
4. The crisp controller output $\bar{y}$ is obtained by calculating the center of gravity of the output set LY:

$$\bar{y} = \frac{\int_{-A}^{+A} \mu_{Y_{Ri}}(y) \cdot y\, dy}{\int_{-A}^{+A} \mu_{Y_{Ri}}(y)\, dy} \quad (29)$$

The cut operation (min operation), the max operation over all resulting fuzzy subsets LY_{Ri}, and the center of gravity are nonlinear operations that cause a nonlinear operating line between x and $\bar{y}$. This seems to make a systematic design of a desired transfer function with the help of membership functions difficult.

However, in the x domain there are operating points A1, A2, A3, A4, and A5 at which only one of the five

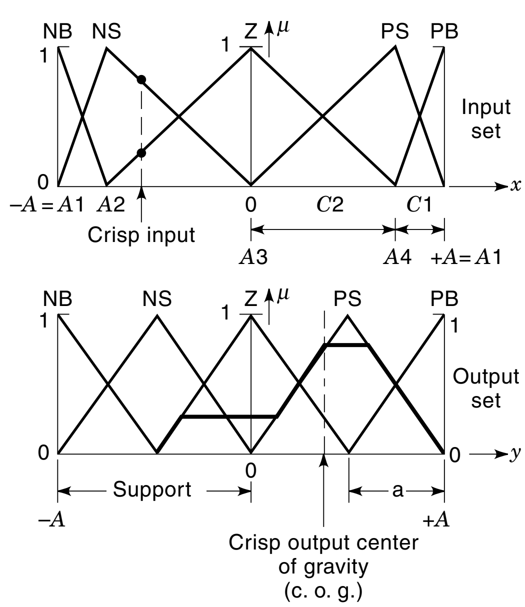

Figure 8. Membership functions for input x and output y. The output membership function is obtained by clipping the output membership functions at the corresponding degrees of membership of the input.

rules fires. At these operating points the center of gravity can be calculated more easily than for the intermediate points. The operating points A1, A2, A3, A4, and A5 form points in the x-y domain (see Fig. 9). The values of the transfer characteristic between the operating points may show a slight nonlinear behavior, but from a linear approximation (interpolation) between two operating points one obtains the relation between the supports of the input and output membership functions, on the one hand, and slopes required of the transfer characteristic, on the other hand.

The Nonlinearity of the Fuzzy Controller

In this subsection we will describe the sources of nonlinearity of the transfer characteristic of a fuzzy controller by relating them to particular computational steps.

System theory distinguishes between two basic types of systems: linear and nonlinear. A system is linear if and only if it has both the additivity property and the scaling property; otherwise it is a nonlinear system.

Additivity Property (Superposition Property). Let it be the case that

$$y_1 = f(x) \quad \text{and} \quad y_2 = f(z) \tag{30}$$

Then for the additivity property to hold, it is required that

$$y_1 + y_2 = f(x + z) \tag{31}$$

Hence, we obtain

$$f(x) + f(z) = f(x + z) \tag{32}$$

Scaling Property (Homogeneity Property). Let it be the case that

$$y = f(x) \tag{33}$$

Then for the scaling property to hold, it is required that

$$\alpha \cdot y = f(\alpha \cdot x) \quad \text{and} \quad \alpha \cdot f(x) = f(\alpha \cdot x) \tag{34}$$

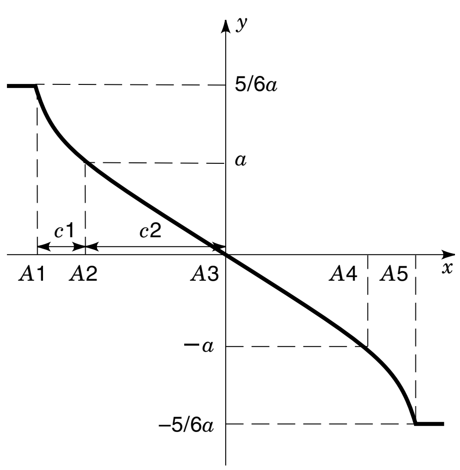

Figure 9. Transfer characteristic of a fuzzy controller. The transfer characteristic is a static input/output mapping of a fuzzy controller.

Because of fuzzification and defuzzification, a fuzzy controller is in fact a crisp transfer element. This crisp TE has a nonlinear transfer characteristic because of the nonlinear character of fuzzification (when performed on nonlinear membership functions), rule firing, and defuzzification. The argument for this is that if one computational step within the computational structure of the TE is nonlinear, then the whole TE is nonlinear as well. Using the additivity and scaling properties of a linear system, we will now establish the linearity, or nonlinearity, of each computational step in the computational structure of a fuzzy controller with respect to these two properties.

In what follows, without any loss of generality, we will use a single SISO fuzzy rule such as

$$R_C\text{:} \quad \text{IF} \quad e = LE \quad \text{THEN} \quad u = LU \tag{35}$$

where LE and LU are the fuzzy values taken by the normalized, single controller input e and the normalized, single controller output u, respectively. These two fuzzy values are determined by the membership functions $\int_E \mu_{LE}(e)/e$ and $\int_U \mu_{LU}(u)/u$ defined on the normalized domains E and U. Here again we only consider normalized domains, fuzzy and crisp values, and thus the lower index N will be omitted from the notation unless there is a need to distinguish between normalized and actual crisp and fuzzy values used within the same expression.

Furthermore, let e_1^* and e_2^* be two normalized crisp controller inputs and u_1^* and u_2^* be the defuzzified controller outputs corresponding to these normalized controller inputs.

Input Scaling and Output Scaling. Input scaling is linear because it simply multiplies each physical controller input e_1^* and e_2^* with a predetermined scalar N_e (normalization factor) to obtain their normalized counterparts e_{1N}^* and e_{2N}^*. Thus we have

$$N_e \cdot e_1^* + N_e \cdot e_2^* = N_e \cdot (e_1^* + e_2^*) \tag{36}$$

Furthermore, for a given scalar α we have

$$\alpha \cdot N_e \cdot e_1^* = N_e \cdot (\alpha \cdot e_1^*) \tag{37}$$

Thus input scaling has the properties of additivity and scaling and is thus a linear computational step. The same is valid for output scaling since it uses N_e^{-1} instead of N_e.

Fuzzification. Let the membership function $\int_E \mu_{LE}(e)/e$ defining the normalized fuzzy value LE be, in general, a nonlinear function (e.g., a triangular membership function). The fuzzification of e_1^* and e_2^* results in finding $\mu_{LE}(e_1^*)$ and $\mu_{LE}(e_2^*)$. Linearity requires

$$\mu_{LE}(e_1^*) + \mu_{LE}(e_2^*) = \mu_{LE}(e_1^* + e_2^*) \tag{38}$$

The preceding equality cannot be fulfilled because the membership function $\int_E \mu_{LE}(e)/e$ is, in general, nonlinear. Thus, fuzzification in the case of nonlinear membership functions is a nonlinear computational step.

Rule Firing. Let the membership function $\int_U \mu_{LU}(u)/u$ defining the normalized fuzzy value LU be, in general, a nonlinear function. Then the result of rule firing given the normalized crisp controller input e_1^* will be

$$\forall u{:}\mu'_{CLU}(u) = \min(\mu_{LU}(e_1^*), \mu_{LU}(u)) \tag{39}$$

Similarly, for the normalized crisp controller input e_2^* we obtain

$$\forall u{:}\mu''_{CLU}(u) = \min(\mu_{LE}(e_2^*)\mu_{LU}(u)) \tag{40}$$

Linearity requires

$$\forall u{:}\mu'_{CLU}(u) + \mu''_{CLU}(u) = \min(\mu_{LE}(e_1^* + e_2^*), \mu_{LU}(u)) \tag{41}$$

but the preceding equality does not hold because

- $\int_U \mu_{LU}(u)/u$ is a nonlinear membership function.
- $\int_U \mu'_{CLU}(u)/u$ and $\int_U \mu''_{CLU}(u)/u$ are nonlinear membership functions (usually defined as only piecewise linear functions).
- the min-operator is nonlinear.

Thus rule firing is a nonlinear computational step within the computational structure of a fuzzy controller.

Defuzzification. Let defuzzification be performed with the center of gravity method. Furthermore, let u_1 and u_2 be the normalized defuzzified controller outputs obtained after defuzzification. That is,

$$u_1 = \frac{\int_U \mu'_{CLU}(u)\cdot u\,du}{\int_U \mu'_{CLU}(u)\,du}, \tag{42}$$

$$u_2 = \frac{\int_U \mu''_{CLU}(u)\cdot u\,du}{\int_U \mu''_{CLU}(u)\,du} \tag{43}$$

Linearity requires, however,

$$u_1 + u_2 = \frac{\int_U (\mu'_{CLU}(u) + \mu''_{CLU}(u))\cdot u\,du}{\int_U (\mu'_{CLU}(u) + \mu''_{CLU}(u))\,du} \tag{44}$$

However, the preceding equality cannot be fulfilled since instead of it we have

$$u_1 + u_2 = \frac{\int_U \mu'_{CLU}(u)\cdot u\,du}{\int_U \mu'_{CLU}(u)\,du} + \frac{\int_U (\mu''_{CLU}(u)\cdot u\,du}{\int_U \mu''_{CLU}(u)\,du} \tag{45}$$

This shows that the nonlinearity of the computational step of defuzzification comes from the normalization of the products $\int_U \mu'_{CLU}(u)\cdot u du$ and $\int_U \mu''_{CLU}(u)\cdot u du$.

From all of the foregoing it is readily seen that a fuzzy controller is a nonlinear TE, its sources of nonlinearity being the nonlinearity of membership functions, rule firing, and defuzzification.

However, in the case of a Takagi Sugeno FC-1, each single fuzzy rule is a linear TE for all controller inputs (state vectors) that belong to the center of the fuzzy region specified by the IF part of this rule. At the same time, for controller inputs outside the center of a fuzzy region, this same fuzzy rule is a nonlinear TE. Because of the latter, the set of all fuzzy rules of a Takagi Sugeno FC-1 defines a nonlinear TE. In the case of a Takagi Sugeno gain scheduler, we have that each fuzzy rule defines a linear TE everywhere in a given fuzzy region.

HEURISTIC CONTROL AND MODEL-BASED CONTROL

Fuzzy control can be classified into the main directions *heuristic fuzzy control* and *model-based fuzzy control.* Heuristic control deals with plants that are unsufficiently described from the mathematical point of view, while model-based fuzzy control deals with plants for which a mathematical model is available. In this section we will describe the following control strategies:

Mamdani control (MC)
Sliding mode fuzzy control (SMFC)
Cell mapping control (CM)
Takagi Sugeno control (TS1)
Takagi Sugeno control (TS2) with Lyapunov linearization

The Mamdani Controller

This type of fuzzy controller obtains its control strategy from expert knowledge. Since a model of the plant is not available, a simulation of the closed loop cannot be performed. Therefore, the control design is based on trial-and-error strategies, which makes the implementation of the fuzzy controller critical. The crucial point is that the behavior of the plant to be controlled is only reflected through the operator rules. However, from the control point of view this is not a satisfactory situation. Thus, one seeks methods to build qualitative models in terms of fuzzy rules.

In the context of heuristic control, the so-called Mamdani control rules are used where both the antecedent and the consequent include fuzzy values. A typical control rule (operator rule) is

$$R_{Ci}\text{:}\quad \text{IF}\quad \boldsymbol{x} = \boldsymbol{LX^i}\quad \text{THEN}\quad \boldsymbol{u} = \boldsymbol{LU^i} \tag{46}$$

For a system with two state variables and one control variable, we have, for example,

$$R_{Ci}\text{:}\quad \text{IF}\quad x = \text{PS}\quad \text{AND}\quad \dot{x} = \text{NB}\quad \text{THEN}\quad u = \text{PM} \tag{47}$$

which can be rewritten into

$$R_{Ci}\text{:}\quad \text{IF}\quad (x, \dot{x})^T = (\text{PS}, \text{NB})^T\quad \text{THEN}\quad u = \text{PM} \tag{48}$$

with

$$\boldsymbol{x} = (x, \dot{x})^T$$
$$\boldsymbol{LX^i} = (\text{PS}, \text{NB})^T$$
$$\boldsymbol{u} = u$$
$$\boldsymbol{LU^i} = \text{PM}$$

Even if there is only a little knowledge about the system to be controlled, one has to have some ideas about the behavior of the system state vector $\boldsymbol{x}$, its change with time $\dot{\boldsymbol{x}}$, and the control variable $\boldsymbol{u}$. This kind of knowledge is structural and can be formulated in terms of fuzzy rules. A typical fuzzy rule for a system is

$$R_{Si}: \quad \text{IF} \quad \boldsymbol{x} = \boldsymbol{LX^i} \quad \text{AND} \quad \boldsymbol{u} = \boldsymbol{LU^i} \quad \text{THEN} \quad \dot{\boldsymbol{x}} = \boldsymbol{L\dot{X}^i} \tag{49}$$

For the preceding system with two states and one control variable we have, for example,

$$\begin{aligned} R_{Si}: \quad & IF \quad (x, \dot{x})^T = (\text{PS}, \text{NB})^T \quad AND \quad u = \text{PM} \\ & THEN \quad (\dot{x}, \ddot{x})^T = (\text{NM}, \text{PM})^T \end{aligned} \tag{50}$$

with

$$\begin{aligned} \boldsymbol{x} &= (x, \dot{x})^T \\ \boldsymbol{LX^i} &= (\text{PS}, \text{NB})^T \\ \dot{\boldsymbol{x}} &= (\dot{x}, \ddot{x})^T \\ \boldsymbol{L\dot{X}^i} &= (\text{NM}, \text{PM})^T \\ \boldsymbol{u} &= u \\ \boldsymbol{LU^i} &= \text{PM} \end{aligned}$$

Once the qualitative system structure is known, one has to find the corresponding quantitative knowledge. Quantitative knowledge means the following: In general, both control rules and system rules work with normalized domains. The task is to map inputs and outputs of both the controller and the system to normalized domains. For the system, this task is identical with the identification of the system parameters. For the controller, this task is identical with the controller design (namely, to find the proper control gains).

Sliding Mode Fuzzy Controller

A typical Mamdani controller is the sliding mode fuzzy controller (SMFC) (19–21). Fuzzy controllers for a large class of second-order nonlinear systems are designed by using the phase plane determined by error e and change of error $\dot{e}$ (22–25). The fuzzy rules of these fuzzy controllers determine a fuzzy value for the input u for each pair of fuzzy values of error and change of error (that is, for each fuzzy state vector). The usual heuristic approach to the design of these fuzzy rules is the partitioning of the phase plane into two semiplanes by means of a sliding (switching) line. This means that the fuzzy controller has a so-called diagonal form (see Fig. 10). Another possibility is, instead of using a sliding line, to use a sliding curve like a time optimal trajectory (26).

A typical fuzzy rule for the fuzzy controller in a diagonal form is

$$\text{IF} \quad e = \text{PS} \quad \text{AND} \quad \dot{e} = \text{NB} \quad \text{THEN} \quad u = \text{PS} \tag{51}$$

where PS stands for the fuzzy value of error POSITIVE SMALL, NB stands for the fuzzy value change of error NEGATIVE BIG, and PS stands for the fuzzy value POSITIVE SMALL of the input.

Each semiplane is used to define only negative or positive fuzzy values of the input u. The magnitude of a specific positive/negative fuzzy value of u is determined on the basis of the distance $|s|$ between its corresponding state vector $\boldsymbol{e}$ and the sliding line $s = \lambda \cdot e + \dot{e} = 0$. This is normally done in such a way that the absolute value of the required input u increases/decreases with the increasing/decreasing distance between the state vector $\boldsymbol{e}$ and the sliding line $s = 0$.

$\dot{e}$ \ e	NB	NM	NS	Z	PS	PM	PB
PB	Z	NS	NS	NM	NM	NB	NB
PM	PS	Z	NS	NS	NM	NM	NB
PS	PS	PS	Z	NS	NS	NM	NM
Z	PM	PS	PS	Z	NS	NS	NM
NS	PM	PM	PS	PS	Z	NS	NS
NM	PB	PM	PM	PS	PS	Z	NS
NB	PB	PB	PM	PM	PS	PS	Z

P — postitve
N — negative
Z — zero
S — small
M — medium
B — big

Figure 10. A fuzzy controller in a diagonal form. Diagonal form means that the same fuzzy attributes appear along a diagonal.

It is easily observed that this design method is very similar to sliding mode control (SMC) with a boundary layer (BL), which is a robust control method (12,27). Sliding mode control is applied especially to control of nonlinear systems in the presence of model uncertainties, parameter fluctuations, and disturbances. The similarity between the diagonal form fuzzy controller and SMC enables us to redefine a diagonal form fuzzy controller in terms of an SMC with BL and then to verify its stability, robustness, and performance properties in a manner corresponding to the analysis of an SMC with BL. In the following, the diagonal fuzzy controller is therefore called sliding mode fuzzy control (SMFC).

However, one is tempted to ask here, What does one gain by introducing the SMFC type of controller? The answer is that SMC with BL is a special case of SMFC. SMC with BL provides a linear transfer characteristic with lower and upper bounds, while the transfer characteristic of an SMFC is not necessarily a straight line between these bounds, but a curve that can be adjusted to reflect given performance requirements. For example, normally a fast rise time and as little overshoot as possible are the required performance characteristics for the closed-loop system. These can be achieved by making the controller gains much larger for state space regions far from the sliding line than its gains in state space regions close to the sliding line (see Fig. 11).

In this connection it has to be emphasized that an SMFC is a state-dependent filter. The slope of its transfer characteristic decides the convergence rate to the sliding line and, at the same time, the bandwidth of the unmodeled disturbances that can be coped with. This means that far from the sliding line higher frequencies are allowed to pass through than in the neighborhood of it. The other function of this state-dependent filter is given by the sliding line itself. That is, the velocity with which the origin is approached is determined by the slope λ of the sliding line $s = 0$.

Because of the special form of the rule base of a diagonal form fuzzy controller, each fuzzy rule can be redefined in terms of the fuzzy value of the distance $|s|$ between the state vector $\boldsymbol{e}$ and the sliding line and the fuzzy value of the input

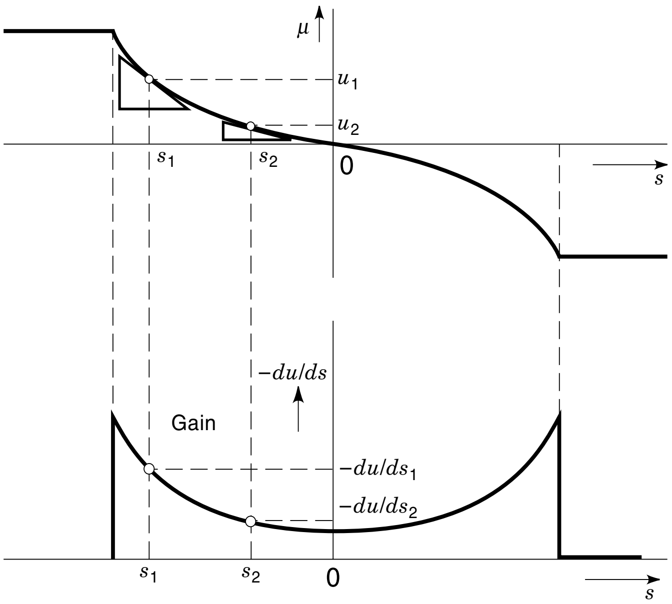

Figure 11. The adjustable transfer characteristic of an SMFC. The nonlinear input/output mapping of the SMFC provides a nonlinear gain for different input/output pairs.

u corresponding to this distance. This helps to reduce the number of fuzzy rules, especially in the case of higher-order systems. Namely, if the number of state variables is 2 and each state variable has m fuzzy values, the number of fuzzy rules of the diagonal form fuzzy controller is $M = m^2$. For the same case, the number of fuzzy rules of an SMFC is only m. This is so because the fuzzy rules of the SMFC only describe the relationship between the distance to the sliding line and the input u corresponding to this distance, rather than the relationship between all possible fuzzy state vectors and the input u corresponding to each fuzzy state vector.

Moreover, the fuzzy rules of an SMFC can be reformulated to include the distance d between the state vector $\boldsymbol{e}$ and the vector normal to the sliding line and passing through the origin (see Fig. 12). This gives an additional opportunity to affect

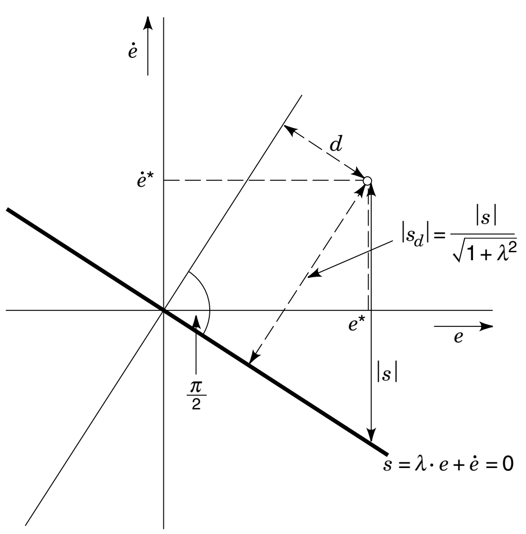

Figure 12. The s and d parameters of an SMFC. s is the distance between the state and the line $s = 0$. d is the distance between the state and the line perpendicular to $s = 0$.

the rate with which the origin is approached. A fuzzy rule including this distance is of the form

$$\text{IF}\quad s = \text{PS}\quad \text{AND}\quad d = \text{S}\quad \text{THEN}\quad u = \text{NS} \tag{52}$$

Despite of the advantages of an SMFC, it poses a number of problems the solutions of which can improve its performance and robustness. One such problem is the addition of an integrator to an SMFC in order to eliminate remaining errors in the presence of disturbances and model uncertainties. There are several ways to accomplish this. One option, for example, is to treat the integration term in the same manner as the other parameters of the IF part of the SMFC's fuzzy rules. This and other available options will be described later in this article.

Another problem is the so-called scaling of the SMFC parameters so that the domains on which their fuzzy values are defined are properly determined and optimized with respect to performance. This problem arises in the context of SMFC since the real physical domains of the SMFC parameters are normalized (i.e., their measured values are mapped on their respective normalized domains by the use of normalization factors). Thus a normalized input u is the result of the computation with SMFC. The normalized u is then consequently denormalized (i.e., mapped back on its physical domain) by the use of a denormalization factor.

The determination of the proper scaling factors, via which the normalization and denormalization of the SMFC parameters is performed, is not only part of the design, but is also important in the context of adaptation and on-line tuning of the SMFC. The behavior of the closed-loop system ultimately depends on the normalized transfer characteristic (control surface) of the SMFC. This control surface is mainly determined by the shape and location of the membership functions defining the fuzzy values of the SMFC's parameters. In this context one need pay attention to the following:

1. The denormalization factor for u influences most stability and oscillations. Because of its impact on stability, the determination of this factor has the highest priority in the design.
2. Normalization factors influence most of the SMFC sensitivity with respect to the proper choice of the operating regions for s and d. Therefore, normalization factors are second in priority in the design.
3. The proper shape and location of the membership functions and, with this, the transfer characteristics of the SMFC can influence positively the behavior of the closed-loop system in different fuzzy regions of the fuzzy state space provided that the operating regions of s and u are properly chosen through well-adjusted normalization factors. Therefore, this aspect is third in priority.

A third problem is the design of SMFC for MIMO systems. The design for SISO systems can still be utilized, though some new aspects and restrictions come into play when this design is extended to the case of MIMO systems. First, we assume that the MIMO system has as many input variables u_i as it has output variables y_i. Second, we assume that the so-called matching condition holds (12). This condition constrains the so-called parametric uncertainties. These are, for example, imprecision on the mass or inertia of a mechanical

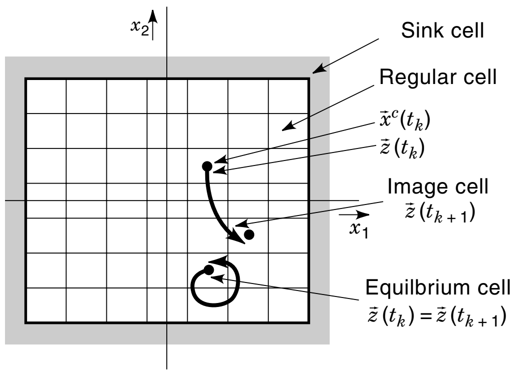

Figure 13. Cell mapping principle. The state space is partitioned into a finite set of cells. Cell mapping deals with the transition behavior between cells.

system and inaccuracies on friction functions. Nonparametric uncertainties include unmodeled dynamics and neglected time delays.

Let $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + B \cdot \boldsymbol{u}$, $\boldsymbol{y} = C \cdot \boldsymbol{x}$ be the nonlinear open-loop system to be controlled, where $\boldsymbol{f}$ is a nonlinear vector function of the state vector $\boldsymbol{x}$, $\boldsymbol{u}$ is the input vector, B is the input matrix, $\boldsymbol{y}$ is the output vector, and C is the output matrix.

Then the matching condition requires that the parametric uncertainties have to be within the range of the input matrix B.

CELL MAPPING

Cell mapping originates from a computational technique introduced by C. S. Hsu (28) that evaluates the global behavior and the stability of nonlinear systems. It is assumed that the computational (analytical) model of the system is available. Cell mapping was first applied to fuzzy systems by Chen and Tsao (29).

The benefits of using cell mapping for fuzzy controlled systems are as follows:

- Supporting of self-learning FC strategies
- Creating of methodologies for the design of time optimal fuzzy controllers

The basic idea of Hsu is as follows: Let a nonlinear system be described by the point mapping

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{f}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)) \tag{53}$$

where t_k represent the discrete timesteps over which the point mapping occurs. It has to be emphasized that these timesteps need not to be uniform in duration. If one wants to create a map of the state space taking into account all possible states $\boldsymbol{x}$ and control vectors $\boldsymbol{u}$, one obtains an infinite number of mappings even for finite domains for $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively. To simplify this mapping, the (finite) state space is divided into a finite number of cells (see Fig. 13). Cells are formed by partitioning the domain of interest of each axis x_i of the state space into intervals of size s_i that are denoted by an integer valued index z_i. Then a cell is an n-tuple (a vector) of intervals $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^T$. The remainder of the state space outside the finite state space of interest is lumped together into one so-called sink cell. The state of the system of Eq. (53) while in the cell $\boldsymbol{z}$ is represented by the center point $\boldsymbol{x}^c$. Now a cell mapping $\boldsymbol{C}$ is defined by

$$\boldsymbol{z}(t_{k+1}) = \boldsymbol{C}(\boldsymbol{z}(t_k)) \tag{54}$$

which is derived from the point mapping of Eq. (53) by computing the image of a point $\boldsymbol{x}(t_k)$ and then determining the cell in which the image point is located. It is clear that not all points $\boldsymbol{x}(t_k)$ in cell $\boldsymbol{z}(t_k)$ have the same image cell $\boldsymbol{z}(t_{k+1})$. Therefore, only the image cell of the *center* $\boldsymbol{x}^c(t_k)$ is considered. A cell that maps to itself is called an equilibrium cell. All cells in the finite state space are called regular cells.

The motivation for cell mapping is to obtain an appropriate sequence of control actions $\boldsymbol{u}(t_k)$ that drive the system of Eq. (53) to an equilibrium while minimizing a predefined cost function. Therefore, every cell is characterized by the following:

- The group number $G(\boldsymbol{z})$ that denotes cells $\boldsymbol{z}$ belonging to the same periodic domain or domain of attraction
- The step number $S(\boldsymbol{z})$ that indicates the number of transitions needed to transmit from cell $\boldsymbol{z}$ to a periodic cell
- The periodicity number $P(\boldsymbol{z})$ that indicates the number of cells contributing to the periodic motion

This characterization is introduced in order to find periodic motions and domains of attractions by a grouping algorithm.

Applied to fuzzy control, it is evident that each cell describing the system's behavior belongs to a corresponding fuzzy system rule. Furthermore, each cell describing a particular control action belongs to a corresponding fuzzy control rule.

Smith and Comer developed a fuzzy cell mapping algorithm the aim of which is to calibrate (tune) a fuzzy controller on the basis of the cell state space concept (30). Each cell is associated with a control action and a duration, which map the cell to a minimum cost trajectory (e.g., minimum time). With a given cost function and a plant simulation model, the cell state space algorithm generates a table of desired control actions. The mapping from cell to cell is carried out by a fuzzy controller, which smoothes out the control actions while the transitions between the cells. The cell-to-cell mapping technique has been used to fine-tune a Takagi Sugeno controller (see Fig. 14) (31).

Kang and Vachtsevanos developed a phase portrait assignment algorithm that is related to cell-to-cell mapping (32). In

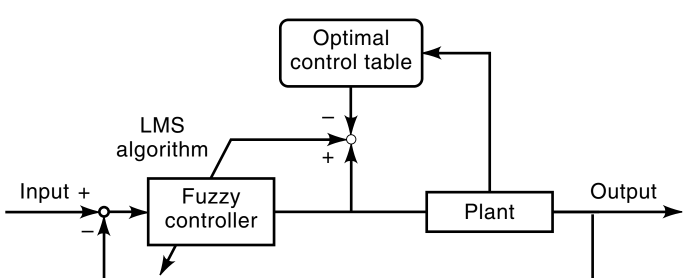

Figure 14. Cell mapping by Smith and Comer [Redrawn from Papa et al. 1995 (31)]. Cell mapping is used to fine-tune a Takagi Sugeno controller.

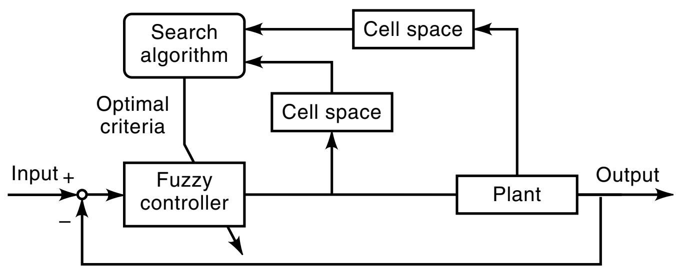

Figure 15. Cell mapping by Kang and Vachtsevanos [Redrawn from Papa et al. 1995 (31)]. Cell mapping is used for construction of an optimal rule base from data.

this approach states and control variables are partitioned into different cell spaces. The $\boldsymbol{x}$-cell space is recorded by applying a constant input to the system being simulated. Then, by means of a searching algorithm, the rule base of the funny controller is constructed such that asymptotic stability is guaranteed. This is performed by determining the optimal control actions regardless of from which cell the algorithm starts its search (see Fig. 15) (31).

Hu, Tai, and Shenoi apply genetic algorithms to improve the searching algorithm using cell maps (33). The aim of this method is to tune a Takagi Sugeno controller.

TS Model-Based Control

Model-based fuzzy controller design starts from the mathematical knowledge of the system to be controlled (8,34). In this connection one is tempted to ask why one should use FC in this particular case while conventional control techniques work well. The reasons that apply FC in analytical known systems are as follows:

1. FC is a user-friendly and transparent control method because of its rule-based structure.
2. FC provides a nonlinear control strategy that is related to traditional nonlinear control techniques.
3. The nonlinear transfer characteristics of a fuzzy controller can be tuned by changing the shape and location of the membership functions so that adaptation procedures can be applied.
4. The approximation property of FC allows the design of a complicated control law with the help of only few rules.
5. Gain scheduling techniques can be transfered to FC. In this connection FC is used as an approximator between linear control laws.

The description of the system starts from a fuzzy model of the system that uses both the fuzzy state space and a crisp description of the system.

Let the principle of a Takagi Sugeno system be explained by the following example.

Example Consider a TS system consisting of two rules with x_1 and x_2 as system inputs and y as the system output.

$$R_1: \text{ if } x_1 \text{ is BIG and } x_2 \text{ is MEDIUM then } y_1 = x_1 - 3 \cdot x_2.$$
$$R_2: \text{ if } x_1 \text{ is SMALL and } x_2 \text{ is BIG then } y_2 = 4 + 2 \cdot x_1.$$

Let the inputs measured be $x_1^* = 4$ and $x_2^* = 60$. From Fig. 16 we then obtain

$$\mu_{X_{\text{BIG}}}(x_1^*) = 0.3 \quad \mu_{X_{\text{BIG}}}(x_2^*) = 0.35$$

and

$$\mu_{X_{\text{SMALL}}}(x_1^*) = 0.7 \quad \mu_{X_{\text{MED}}}(x_2^*) = 0.75$$

For the degree of satisfaction of R_1 and R_2, respectively, we obtain

$$\min(0.3, 0.75) = 0.3 \quad \min(0.7, 0.35) = 0.35$$

Furthermore, for the consequents of rules R_1 and R_2 we have

$$y_1 = 4 - 3 \cdot 60 = -176 \quad y_2 = 4 + 2 \cdot 4 = 12$$

So the two pairs corresponding to each rule are (0.3, −176) and (0.35, 12). Thus, by taking the weighted normalized sum we get

$$y = \frac{0.3 \cdot (-176) + 0.35 \cdot 12}{0.3 + 0.35} = -74.77$$

This can be extended to differential equations in the following way: Let a fuzzy region $\boldsymbol{LX^i}$ be described by the rule

$$R_{Si}: \text{ IF } \boldsymbol{x} = \boldsymbol{LX^i} \text{ THEN } \dot{\boldsymbol{x}} = A(\boldsymbol{x^i}) \cdot \boldsymbol{x} + B(\boldsymbol{x^i}) \cdot \boldsymbol{u} \tag{55}$$

This rule means that IF state vector $\boldsymbol{x}$ is in fuzzy region $\boldsymbol{LX^i}$ THEN the system obeys the local differential equation $\dot{\boldsymbol{x}} = A(\boldsymbol{x^i}) \cdot \boldsymbol{x} + B(\boldsymbol{x^i}) \cdot \boldsymbol{u}$. A summation of all contributing system rules provides the global behavior of the system. In Eq. (55) $A(\boldsymbol{x^i})$ and $B(\boldsymbol{x^i})$ are constant system matrices in the center of fuzzy region $\boldsymbol{LX^i}$ that can be identified by classical identification procedures.

The resulting system equation is

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{n} w_i(\boldsymbol{x}) \cdot (A(\boldsymbol{x^i}) \cdot \boldsymbol{x} + B(\boldsymbol{x^i}) \cdot \boldsymbol{u}) \tag{56}$$

where $w_i(\boldsymbol{x}) \in [0, 1]$ are the normalized degrees of satisfaction of a fuzzy region $\boldsymbol{LX^i}$.

The corresponding control rule (Takagi Sugeno FC1) is

$$R_{Ci}: \text{ IF } \boldsymbol{x} = \boldsymbol{LX^i} \text{ THEN } \boldsymbol{u} = K(\boldsymbol{x^i}) \cdot \boldsymbol{x} \tag{57}$$

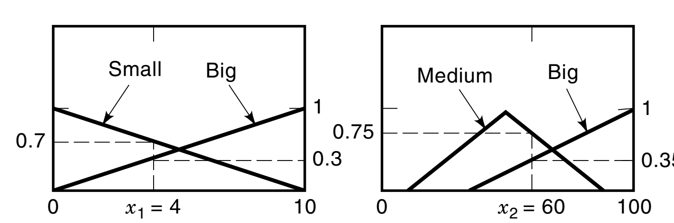

Figure 16. Fuzzification procedure for a TS controller. The fuzzification procedure is the same as that for a Mamdani controller.

and the control law for the whole state space is

$$\boldsymbol{u} = \sum_{i=1}^{n} w_i(\boldsymbol{x}) \cdot K(\boldsymbol{x^i}) \cdot \boldsymbol{x} \tag{58}$$

Together with Eq. (56) one obtains the closed-loop system

$$\dot{\boldsymbol{x}} = \sum_{i,j=1}^{n} w_i(\boldsymbol{x}) \cdot w_j(\boldsymbol{x}) \cdot (A(\boldsymbol{x^i}) + B(\boldsymbol{x^i}) \cdot K(\boldsymbol{x^j})) \cdot \boldsymbol{x} \tag{59}$$

It has to be emphasized that a system described by a set of rules like Eq. (55) is nonlinear even in the vicinity of the center of the region. This is due to the fact that $w_i(\boldsymbol{x})$ depends on the state vector $\boldsymbol{x}$. Even if $w_i(\boldsymbol{x})$ is a piecewise linear function of $\boldsymbol{x}$, the product $w_i(\boldsymbol{x}) \cdot w_j(\boldsymbol{x}) \cdot (A(\boldsymbol{x}^i) + B(\boldsymbol{x}^i) \cdot K(\boldsymbol{x}^j)) \cdot \boldsymbol{x}$ in Eq. (58) will always be a nonlinear function.

Model-Based Control with Lyapunov Linearization

In the following we discuss the case when a mathematical model of the system to be controlled is available and the fuzzy controller is formulated in terms of fuzzy rules (8,21,34,35). In this case system and controller are formulated on different semantic levels. Let the system analysis starts from the mathematical model of the system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{60}$$

and let the fuzzy controller be formulated in terms of fuzzy rules (Takagi Sugeno FC1)

$$R_{Ci}: \text{ IF } \boldsymbol{x} = \boldsymbol{LX^i} \text{ THEN } \boldsymbol{u} = \boldsymbol{LU^i} \tag{61}$$

To study stability, robustness, and performance of the closed-loop system one has to bring system and controller onto the same semantic level. Thus, formally we translate the set of control rules into an analytical structure

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}) \tag{62}$$

where, in general, the function $\boldsymbol{g}(\boldsymbol{x})$ is a nonlinear control surface being a static mapping of the state vector $\boldsymbol{x}$ to the control vector $\boldsymbol{u}$ (see Fig. 17).

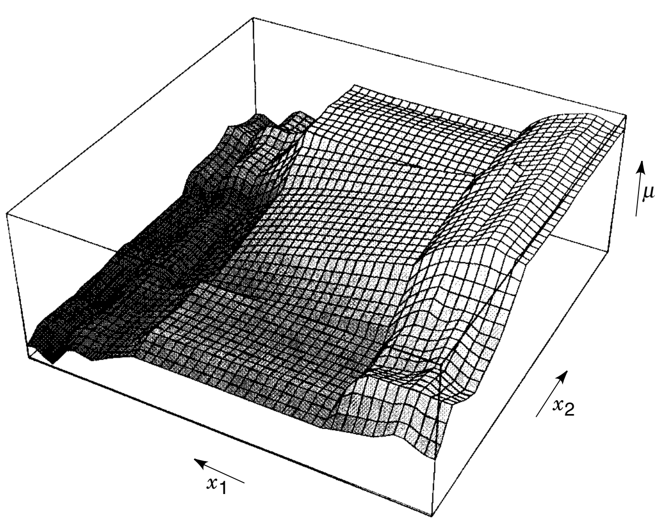

Figure 17. Nonlinear control surface $\boldsymbol{u} = \boldsymbol{g}(x_1, x_2)$. Nonlinear mapping is a translation of fuzzy rules into a numerical input/output relation.

The control surface provides information about local and global properties of the controller. For example, the local gain for a specific state vector can be obtained by means of the tangential plane being attached to the corresponding point in state space. From this information one can conclude whether the controlled system is locally stable. Furthermore, one obtains a geometrical insight into how the control gain changes as the state trajectory moves in the state space.

Another aspect is the following. To study the local behavior of the system around specific points in state space, we linearize the system around them and study the closed-loop behavior in the linearized region. Let, for example, the system of Eq. (60) be linearized around a desired state $\boldsymbol{x_d}$ and a corresponding state vector $\boldsymbol{u_d}$:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_d}, \boldsymbol{u_d}) + A(\boldsymbol{x_d}, \boldsymbol{u_d}) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) + B(\boldsymbol{x_d}, \boldsymbol{u_d}) \cdot (\boldsymbol{u} - \boldsymbol{u_d}) \tag{63}$$

where

$$A(\boldsymbol{x_d}, \boldsymbol{u_d}) = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x_d}, \boldsymbol{u_d}} \quad \text{and} \quad B(\boldsymbol{x_d}, \boldsymbol{u_d}) = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x_d}, \boldsymbol{u_d}}$$

are Jacobians.

An appropriate control law is

$$\boldsymbol{u} = \boldsymbol{u_d} + K(x_d) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) \tag{64}$$

where $K(\boldsymbol{x_d})$ is the gain matrix. Since the system of Eq. (60) changes its behavior with the setpoint $\boldsymbol{x_d}$, the control law of Eq. (64) changes with the setpoint $\boldsymbol{x_d}$ as well. To design the controller for the closed-loop system at any arbitrary point $\boldsymbol{x_d}$ in advance, we approximate Eq. (63) by a set of TS fuzzy rules

$$R_{Si}: \text{ IF } \boldsymbol{x_d} = \boldsymbol{LX^i}$$
$$\text{THEN } \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_d}, \boldsymbol{u_d}) + A(\boldsymbol{x^i}, \boldsymbol{u^i}) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) + B(\boldsymbol{x^i}, \boldsymbol{u^i}) \cdot (\boldsymbol{u} - \boldsymbol{u_d}) \tag{65}$$

The resulting system equation is (Takagi Sugeno FC2)

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_d}, \boldsymbol{u_d}) + \sum_{i=1}^{n} w_i(\boldsymbol{x_d}) \cdot (\boldsymbol{A}(\boldsymbol{x^i}, \boldsymbol{u^i}) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) + \boldsymbol{B}(\boldsymbol{x^i}, \boldsymbol{u^i}) \cdot (\boldsymbol{u} - \boldsymbol{u_d})) \tag{66}$$

This is a linear differential equation because the weights w_i depend on the desired state vector $\boldsymbol{x_d}$ instead of on $\boldsymbol{x}$.

The corresponding set of control rules is

$$R_{Ci}: \text{ IF } \boldsymbol{x_d} = \boldsymbol{LX^i} \text{ THEN } \boldsymbol{u} = \boldsymbol{u_d} + K(\boldsymbol{x^i}) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) \tag{67}$$

with the resulting control law

$$\boldsymbol{u} = \sum_{i=1}^{n} w_i(\boldsymbol{x_d}) \cdot (\boldsymbol{u_d} + K(\boldsymbol{x^i}) \cdot (\boldsymbol{x} - \boldsymbol{x_d})) \tag{68}$$

Substituting Eq. (68) into Eq. (66), we obtain the equation for the closed-loop system

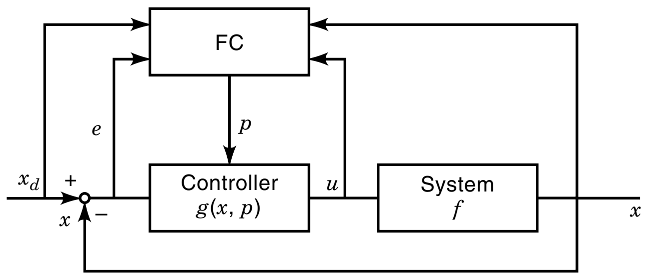

Figure 18. Supervisory control. The supervisor changes controller parameters by means of input/output data $\boldsymbol{u}$ and $\boldsymbol{x}$ and desired values $\boldsymbol{x}_d$.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x_d}, \boldsymbol{u_d}) + \sum_{i,j=1}^{n} w_i(\boldsymbol{x_d}) \cdot w_j(\boldsymbol{x_d}) \cdot (A(\boldsymbol{x^i}, \boldsymbol{u^i}) + B(\boldsymbol{x^i}, \boldsymbol{u^i}) \cdot K(\boldsymbol{x^j})) \cdot (\boldsymbol{x} - \boldsymbol{x_d}) \tag{69}$$

Denoting $A(\boldsymbol{x}^i, \boldsymbol{u}^i) + B(\boldsymbol{x}^i, \boldsymbol{u}^i) \cdot K(\boldsymbol{x}^j)$ by A_{ij}, asymptotic stability of $\boldsymbol{x} - \boldsymbol{x}_d$ is guaranteed if there exists a common positive definite matrix P such that the Lyapunov inequalities

$$A_{ij}^T P + P A_{ij} < 0 \tag{70}$$

hold, where A_{ij} are Hurwitz matrices (34). With this result one is able to study the stability, robustness, and performance of the closed-loop system around an arbitrary setpoint $\boldsymbol{x}_d$ just by considering the system at predefined operating points $\boldsymbol{x}^i$.

SUPERVISORY CONTROL

A commonly used control technique is supervisory control, which is a method to connect conventional control methods and so-called intelligent control methods (see Fig. 18). This control technique works in such a way that one or more controllers are supervised by a control law on a higher level. Applications to supervisory control for a milling machine and a steam turbine are reported in Refs. 36 and 37. Normally, the low-level controllers perform a specific task under certain conditions. These conditions can be

- Keeping a predefined error between desired state and current state
- Performing a specific control task (e.g., approaching a solid surface by a robot arm)
- Being at a specific location of the state space

Usually, supervisors intervene only if some of the predefined conditions fail. If so, the supervisor changes the set of control parameters or switches from one control strategy to another.

Often, supervisory algorithms are formulated in terms of IF-THEN rules. Fuzzy IF-THEN rules avoid hard switching between set of parameters or between control structures. It is therefore useful to build fuzzy supervisors in the cases when "soft supervision" is required.

A formal approach may be the following. Let

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{71}$$

be the model of the system and

$$\boldsymbol{u} = \boldsymbol{g_p}(\boldsymbol{x}, \boldsymbol{p}) \tag{72}$$

be the control law, where $\boldsymbol{p}$ is a parameter vector that has to be determined by the supervisor. The subscript $\boldsymbol{p}$ means that with the change of $\boldsymbol{p}$ the structure of the control law may also change. Then the supervisory law can be written as

$$\boldsymbol{p} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{c}) \tag{73}$$

where $\boldsymbol{c}$ is the vector of conditions. For example,

$$\boldsymbol{c} = (|\boldsymbol{x} - \boldsymbol{x_d}| > K1;\ |\dot{\boldsymbol{x}}_{\boldsymbol{d}}| < K2)^T$$

where $K1$ and $K2$ are constant bounds. The corresponding supervisory fuzzy rule is

$$\text{IF} \quad \boldsymbol{x} = \boldsymbol{LX^i} \quad \text{AND} \quad \boldsymbol{c} = \boldsymbol{LC^i} \quad \text{THEN} \quad \boldsymbol{p} = \boldsymbol{p^i}$$

with $\boldsymbol{LC}^i = (|\boldsymbol{x} - \boldsymbol{x}_d| > K1^i;\ |\dot{\boldsymbol{x}}_d| < K2^i)^T$.

Supervision is related to gain scheduling. The distinction between the two is that gain scheduling changes the controller gains with respect to a slowly time varying scheduling variable while the control structure is preserved (38–40). On the other hand, supervision can both change the control gains and the control structure and can deal with fast-changing system parameters as well (41).

ADAPTIVE CONTROL

Many dynamic systems have a known structure but uncertain or slowly varying parameters. Adaptive control is an approach to the control of such systems. Adaptive controllers, whether designed for linear or nonlinear systems, are inherently nonlinear. We distinguish between *direct* and *indirect* adaptive control methods. Direct adaptive methods start with sufficient knowledge about the system structure and its parameters. Direct change of controller parameters optimizes the system's behavior with respect to a given criterion.

In contrast, the basic idea of indirect adaptive control methods is to estimate the uncertain parameters of the system under control (or, equivalently, the controller parameters) on-line, and use the estimated parameters in the computation of the control law. Thus an indirect adaptive controller can be regarded as a controller with on-line parameter estimation. There do exist systematic methods for the design of adaptive controllers for the control of linear systems. There also exist adaptive control methods that can be applied to the control of nonlinear systems. However, the latter methods require measurable states and a linear parametrization of the dynamics of the system under control (i.e., that parametric uncertainty be expressed linearly in terms of a number of adjustable parameters). This is required in order to guarantee stability and tracking convergence. However, when adaptive control of nonlinear systems is concerned, most of the adaptive control methods can only be applied to SISO nonlinear systems. Since robust control methods are also used to deal with parameter uncertainty, adaptive control methods can be considered as an alternative and complimentary to robust control methods. In principle, adaptive control is superior to robust control in dealing with uncertainties in uncertain or slowly varying parameters.

The reason for this is the learning behavior of the adaptive controller: Such a controller improves its performance in the process of adaptation. On the other hand, a robust controller simply attempts to keep a consistent performance. Furthermore, an indirect adaptive controller requires little a priori information about the unknown parameters. A robust controller usually requires reasonable a priori estimates of the parameter bounds.

Conversely, a robust controller has features that an adaptive controller does not possess, such as the ability to deal with disturbances, quickly varying parameters, and unmodeled dynamics.

In control with a fuzzy controller, there exist a number of direct adaptive control methods aimed at improving the fuzzy controller's performance on-line. The FC's parameters that can be altered on-line are the scaling factors for the input and output signals, the input and output membership functions, and the fuzzy IF-THEN rules. An adaptive fuzzy controller, its adjustable parameters being the fuzzy values and their membership functions, is called a self-tuning fuzzy controller. An adaptive fuzzy controller that can modify its fuzzy IF-THEN rules is called a self-organizing fuzzy controller. Detailed description of the design methods for these two types of direct adaptive fuzzy controllers can be found in Ref. 42. Descriptions of indirect adaptive fuzzy controllers can be found in Ref. 8.

The methods for the design of a self-tuning fuzzy controller can be applied independent of whether its fuzzy IF-THEN rules are derived using model-based fuzzy control or a heuristic design approach and are thus applicable to the different types of fuzzy controllers.

Since tuning and optimization of controllers is related to adaptive control, we use Fig. 19 to illustrate this relationship. In this scheme an adaptation block is arranged above the controller to force the closed-loop system to behave according to a parallel installed reference model. The task is to change the parameters of the controller by means of the adaptation block. Tuning or optimization is performed with the following steps:

1. Optimization criteria are needed that are sufficient for a relevant improvement of the behavior of the system under control. One criterion mostly used is the integral criterion

$$J = \int_0^T (\boldsymbol{e}^{\boldsymbol{T}} Q\boldsymbol{e} + \boldsymbol{u}^{\boldsymbol{T}} R\boldsymbol{u})\, dt \qquad (74)$$

where $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}_d$ is the error, and Q, R are weighting matrices. Another performance criterion can be formulated by fuzzy rules; for example,

IF rise time = SMALL
AND settling time = MEDIUM
THEN performance = HIGH

2. The next point is to choose an appropriate optimization technique (e.g., gradient decent with constant searching step width, or *Rosenbrock's* method with variable searching step widths).
3. A crucial point is to choose a tuning hierarchy (43) that considers the different impacts of the control parameters on stability, performance, and robustness of the closed-loop system:

   Tune the output scaling factors.
   Tune the input scaling factors.
   Tune the membership functions.

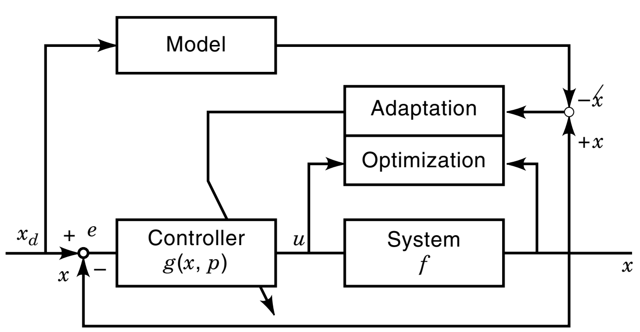

Figure 19. Adaptive control. Using a model of the system the scheme shows an indirect adaptation strategy. Direct adaptation works without an explicit model of the plant.

BIBLIOGRAPHY

1. L. A. Zadeh, Fuzzy sets, *Inf. Control,* **8**: 338–353, 1965.
2. E. H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Man-Machine Studies,* **7** (1): 1–13, 1975.
3. J. J. Østergaard, Fuzzy logic control of a heat exchanger process. In M. M. Gupta (ed.), *Fuzzy Automata and Decision Processes,* Amsterdam: North-Holland, 1977, pp. 285–320.
4. L. Holmblad and J. J. Østergaard, Control of a cement kiln by fuzzy logic. In M. M. Gupta and E. E. Sanchez (eds.), *Fuzzy Information and Decision Processes,* Amsterdam: North-Holland, 1982, pp. 389–399.
5. M. Vidyasagar, *Nonlinear Systems Analysis,* Englewood Cliffs, NJ: Prentice Hall, 1993.
6. W. Pedrycz, *Fuzzy Control and Fuzzy Systems,* 2nd revised ed., Research Studies, 1992.
7. W. Pedrycz, Fuzzy control engineering: Reality and challenges, *IEEE Int. Conf. Fuzzy Syst. 1995, Fuzz-IEEE/IFES'95, Proc.,* Yokohama, March 1995, pp. 437–446.
8. T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modelling and control, *IEEE Trans. Syst. Man Cybern.,* **SMC-15**: 1, 116–132, 1985.
9. L. Wang, Fuzzy systems are universal approximators, *IEEE Int. Conf. Fuzzy Syst. 1992, Fuzz-IEEE'92, Proc.,* San Diego, March 8–12, pp. 1163–1169.
10. B. Kosko, Fuzzy systems as universal approximators, *IEEE Int. Conf. Fuzzy Syst. 1992, Fuzz-IEEE'92, Proc.,* San Diego, March 8–12, 1992, pp. 1153–1162.
11. L. Koczy and S. Kovacs, Linearity and the cnf property in linear fuzzy rule interpolation, *IEEE Int. Conf. Fuzzy Syst. 1994, Fuzz-IEEE'94, Proc.,* Orlando, June 26–29, 1994, pp. 870–875.
12. J-J. E. Slotine and W. Li, *Applied Nonlinear Control,* Englewood Cliffs, NJ: Prentice-Hall, 1991.

13. R. M. Tong, Some properties of fuzzy feedback systems, *IEEE Trans. Syst. Man Cybern.,* **SMC-10**: 327–330, 1980.

14. S. Yasunobu and S. Miyamoto, Automatic train operation system by predictive fuzzy control. In M. Sugeno (ed.), *Industrial Applications of Fuzzy Control,* New York: Elsevier Science, 1985, pp. 1–18.

15. C. E. García, D. M. Prett, and M. Morari, Model predictive control: Theory and practice—a survey, *Automatica,* **25** (3): 335–348, 1989.

16. J. Valente de Oliveira, Long-range predictive adaptive fuzzy relational control, *Fuzzy Sets Syst.,* **70**: 337–357, 1995.

17. I. Scrjanc, K. Kavsek-Biasizzo, and D. Matko, Fuzzy predictive control based on fuzzy model, *EUFIT '96,* Aachen Germany, 1996, pp. 1864–1869.

18. C. W. de Silva and A. G. J. MacFarlane, Knowledge-based control with applications to robots. Lecture Notes in *Control and Information Sciences 123,* Springer-Verlag, Berlin, 1989.

19. G.-C. Hwang and S.-C. Li, A stability approach to fuzzy control design for nonlinear systems, *Fuzzy Sets Syst.,* **48**: 279–287, 1992.

20. S. Kawaji and N. Matsunaga, Fuzzy control of VSS type and its robustness, *IFSA'91 Brussels,* July 7–12, 1991, *preprints vol. "Engineering,"* pp. 81–88.

21. R. Palm, Sliding mode fuzzy control, *IEEE Int. Conf. Fuzzy Syst. 1992, Fuzz-IEEE'92, Proc.,* San Diego, March 8–12, 1992, pp. 519–526.

22. K. S. Ray and D. D. Majumder, Application of circle criteria for stability analysis of linear SISO and MIMO systems associated with fuzzy logic controller, *IEEE Trans. Syst. Man Cybern.,* **14**: 345–349, 1984.

23. K. S. Ray, S. Ananda, and D. D. Majumder, L-stability and the related design concept for SISO linear systems associated with fuzzy logic controller, *IEEE Trans. Syst. Man Cyber.,* **14**: 932–939, 1992.

24. K. L. Tang and R. J. Mulholland, Comparing fuzzy logic with classical control designs, *IEEE Trans. Syst. Cybern.,* **SMC-17**: 1085–1087, 1987.

25. B. A. M. Wakileh and K. F. Gill, Use of fuzzy logic in robotics, *Computers in Industry,* **10**: 35–46, 1988.

26. S. M. Smith, A variable structure fuzzy logic controller with runtime adaptation, *Proc. FUZZ-IEEE'94,* Orlando, Florida, July 26–29, 1994, pp. 983–988.

27. V. J. Utkin, Variable structure systems: A survey, *IEEE Trans. Autom. Control,* **22**: 212–222, 1977.

28. C. S. Hsu, A theory of cell-to-cell dynamical systems, *J. Appl. Mech.,* **47**: 940–948, 1980.

29. Y. Y. Chen and T. C. Tsao, A description of the dynamical behavior of fuzzy systems, *IEEE Trans. Syst. Man Cybern.,* **19**: 745–755, 1989.

30. S. M. Smith and D. J. Comer, An algorithm for automated fuzzy logic controller tuning, *Proc. IEEE Int. Conf. Fuzzy Syst. 1992,* pp. 615–622.

31. M. Papa, H-M. Tai, and S. Shenoi, Design and evaluation of fuzzy control systems using cell mapping, *VI IFSA World Congress,* Sao Paulo, Brazil, 1995, pp. 361–364.

32. H. Kang and G. Vachtsevanos, Nonlinear fuzzy control based on the vector field of the phase portrait assignment algorithm, *Proc. Amer. Control Conf. 1990,* pp. 1479–1484.

33. H-T. Hu, H-M. Tai, and S. Shenoi, Incorporating cell map information in fuzzy controller design, *3rd IEEE Int. Conf. Fuzzy Syst.,* Orlando, 1994, pp. 394–399.

34. K. Tanaka and M. Sugeno, Stability analysis and design of fuzzy control systems, *Fuzzy Sets Syst.,* **45**: 135–156, 1992.

35. U. Rehfuess and R. Palm, Design of Takagi-Sugeno controllers based on linear quadratic control, *Proc. First Int. Symp. Fuzzy Logic,* Zurich, Switzerland, May 26–27, 1995, pp. C10–C15.

36. R. H. Haber et al., Two approaches for a fuzzy supervisory control system of a vertical milling machine, *VI IFSA Congress,* Sao Paulo, Brazil, 1995, pp. 397–400.

37. V. V. Badami et al., Fuzzy logic supervisory control for steam turbine prewarming automation, *3rd IEEE Int. Conf. Fuzzy Syst.,* Orlando, 1994, pp. 1045–1050.

38. W. J. Rugh, Analytical framework for gain scheduling, *IEEE Control Syst. Mag.,* **11** (1): 79–84, 1991.

39. R. A. Nichols, R. T. Reichert, and W. J. Rugh, Gain scheduling for H-infinity controllers: A flight control example, *IEEE Trans. Control Syst. Technol.,* **1**: 69–79, 1993.

40. J. S. Shamma, *Analysis and design of gain scheduled control systems,* Ph.D. thesis No. LIDS-TH-1770, Lab. for Information and Decision Sciences, MIT, Cambridge, MA.

41. L-X Wang, Supervisory controller for fuzzy control systems that guarantees stability, *3rd IEEE Int. Conf. Fuzzy Syst.,* Orlando, 1994, pp. 1035–1039.

42. D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control,* 2nd ed., Berlin: Springer-Verlag, 1996.

43. R. Palm, Tuning of scaling factors in fuzzy controllers using correlation functions, *Proc. FUZZ-IEEE'93,* San Francisco, California, March 28–April 1, 1993, pp. 691–696.

RAINER PALM
Siemens AG

FUZZY CONTROL.

See FUZZY LOGIC; POSSIBILITY THEORY.