2

Model-based Fuzzy Control

The task of feedback control systems is to keep controlled variables as close as possible to their reference values (set-points), which define operating points of the controlled process. A classical approach when selecting a feedback control algorithm is to first identify a linear process model at a given operating point, then to design a linear controller for that model or choose settings of a standard linear controller, usually of a PID type, see *e.g.*, [44, 85, 3, 52]. However, a linear process model is only an approximate description of reality. Moreover, the designed controllers should be robust against possible changes in the characteristics of the controlled process caused by disturbances or changes of its internal features. Robust controllers should ensure stability of the control system and basic control quality for a prescribed range of differences between the process and its model used for the design. For example, when applying classical frequency based design, certain minimal values of amplitude and phase margins must be preserved, assigned on the basis of frequency characteristics of the process model with the controller. These prescribed minimal values result from many years of practical experience, to achieve a reasonable compromise between robustness and control quality. A similar philosophy is applied when designing controllers by other methods, such as the root-locus method, pole placement, or popular in the process industries selection of PID settings using Ziegler-Nichols rules or tables of settings based on simple process models, see *e.g.*, [38, 44, 85, 105]. Such design usually ensures a correct operation of the control system in a certain neighborhood of an operating point for which the controller was designed. The smaller the nonlinearities of the process, the larger this neighborhood, in general.

Practically, however, control processes are usually nonlinear, sometimes strongly nonlinear. If they operate in a small neighborhood of the operating point and control quality requirements are not too high, then the presented design philosophy is usually sufficient. It had to be the case during many years when controllers were of mechanical construction (*e.g.*, pneumatic) as well as later, when analog electronic controllers dominated. Bringing into control practice computers and reliable microprocessor controllers radically changed

the situation. Supervisory control computers enabled effective implementation of on-line optimization of the controlled process, *i.e.*, proper adjustment of process operating points to the current technological situation defined by the process environment and requirements of the management layer. Moreover, microprocessor controllers allow for on-line realization of many complex control algorithms, including multivariable control and nonlinear control. It should be emphasized that the phenomena mentioned above condition each other strongly – on-line optimization of the running process assumes frequent, but always fast and reliable (thus - automatic) changes of the process operating points, implemented as changes in the set-point values for feedback controllers. This is possible only when controllers are able to work reliably not only in the vicinity of one set-point, but also for a range of set-points corresponding to various changes in process input and output values – *i.e.*, controllers which are able to control a nonlinear process.

Generally, the development of control algorithms capable to cope with changing operating points and environment changes has gone in two directions: adaptive control and nonlinear control. The base of *adaptive control* is an on-line adaptation of controller parameters to the changing process features, usually using a standard linear controller (*e.g.*, PID). Adaptation is conducted in a direct way or in an indirect way, by on-line identification of a linear process model corresponding to the current operating point and appropriate selection of controller parameters, usually using one of the classical methods as was described previously, see *e.g.*, [2, 103]. Such an approach is appropriate mainly in situations where we are not able to avoid the necessity of on-line identification during the control system operation. However, on-line identification carries the risk of a failure, particularly in periods of small variability of measured values. Therefore, in the domain of industrial control – in chemical, petrochemical, sugar, food *etc.* industries adaptive control in this sense has so far found quite a limited application.

The essence of *nonlinear control* (non-adaptive) is the design of a nonlinear controller using in a straightforward way a nonlinear process model, valid for a wide range of variability of process input and output values. The theory of nonlinear control is vast – see *e.g.*, [57, 64], and its description is not the aim of this book. This chapter will deal with the design and analysis of nonlinear controllers on the basis of the theory of fuzzy sets and fuzzy logic, particularly fuzzy models with the Takagi-Sugeno structure, proposed in 1985 in [131].

The idea, definitions of a *fuzzy set* and a *fuzzy logic* were proposed by L. A. Zadeh as early as in 1964 [156], at first meeting severe criticism. Although the first successful industrial application for control purposes took place in 1976 (at a Danish cement plant), the real boom in developing the theory and applications of fuzzy logic for control came at the end of the 1980s and lasts until today. It turned out that fuzzy controllers can be a strong, efficient tool, especially in places where it is difficult to have a sufficiently adequate pure analytical description of the controlled process – but where we do have at our disposal empirical knowledge, experience of operators controlling such proces-

ses, or patterns of the required behavior of the controlled processes. On the other hand, fuzzy control of nonlinear processes turned out to be a very efficient tool allowing to effectively combine elements of quality knowledge about the process with the analytical approach. Fuzzy modeling of nonlinear processes for control purposes turned out to be, next to neural network modeling, the most intensively developed approach which has been practically applied from the 1990s.

It is not the intention of this book to describe the basics of the theory of fuzzy sets and fuzzy logic together with possible applications for control purposes, as there exist many splendid books, such as *e.g.*, [155], where vast information on the subject can be found. The basics of the theory of fuzzy sets and fuzzy logic will be presented in the following section, however, only as necessary for introduction of the Takagi-Sugeno (TS) fuzzy modeling structure. We shall then present design methods and stability analysis of *fuzzy nonlinear controllers of Takagi-Sugeno structure*, also known as *fuzzy multi-regional controllers* [33, 32], or *multi-model controllers* [22]. It is not only the authors opinion that this is one of the most successful constructions of nonlinear controllers, especially from a practical, application point of view.

2.1 Takagi-Sugeno (TS) Type Fuzzy Systems

2.1.1 Fuzzy Sets and Linguistic Variables

Figure 2.1 (b) presents how membership of an element $x \in \mathbb{R}^1$ to a fuzzy set F is defined. On the other hand, Fig. 2.1 (a) presents, for comparison, the membership of an element $x \in \mathbb{R}^1$ to the set C defined in a classic way, such a set is called a *crisp set* in the theory of fuzzy sets. Each element of the numerical axis $\mathbb{R}^1$ belongs to the set C or not, the membership function $\mu_C(x)$ of the set C can take only values 0 or 1,

$$\mu_C(x) = \left\{ \begin{array}{ll} 1, & \text{if } x \in C \\ 0, & \text{if } x \notin C \end{array} \right\}$$

The membership function $\mu_F(x)$ of the fuzzy set F can also take any value between 0 and 1,

$$\mu_F(x) = \left\{ \begin{array}{ll} \in (0,1], & \text{if } x \in F \\ 0, & \text{if } x \notin F \end{array} \right\}$$

Figure 2.1 presents an example of a trapezoidal membership function $\mu_F(x)$ of the fuzzy set F. Every point of the interval $[d, e]$ belongs only to the set F, *i.e.*, with the membership function value (with the *grade of membership*) equal to 1, just like each point of the interval $[a, b]$ belongs to the crisp set C. Points of intervals (c, d) and (e, f) do not belong entirely to the set F because corresponding grades of membership are contained in the range $(0, 1)$. In this sense the borders of the set F are *fuzzy*, thus the name *fuzzy set*. For example,

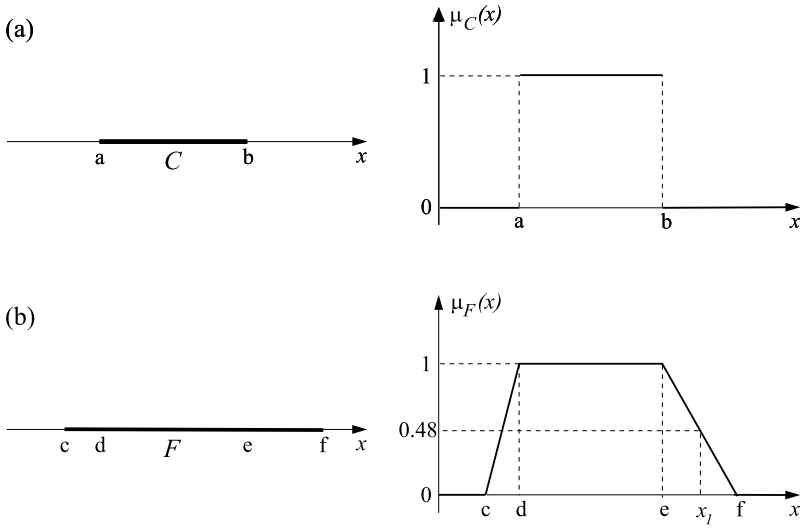

Fig. 2.1. (a) An example of a crisp set C, compared to (b) An example of a fuzzy set F; $\mu_C(x)$, $\mu_F(x)$ – membership functions of sets C and F

the point x_1 depicted in Fig. 2.1 (b) belongs to the set F with the membership function value (with the grade of membership) equal to 0.48.

The definition of a fuzzy set allows to introduce the next concept, key to the fuzzy logic, the concept of a *linguistic variable*. A linguistic variable, also known as a fuzzy variable, is an intermediate between a numerical variable and a symbolic variable (whose values are symbols – *e.g.*, a symbolic variable "shape" defined as taking three values: "circle", "square", "triangle"). The notion of the linguistic variable is explained in Fig. 2.2 by way of an example of a variable "temperature", taking values "low", "medium" and "high", defined by fuzzy sets described by membership functions $\mu_L(t)$, $\mu_M(t)$, and $\mu_H(t)$, respectively. For example, the temperature t_1 presented in Fig. 2.2 is medium with grade of membership equal to 0.79 and, at the same time, it is high with the grade of membership equal to 0.21.

Figures 2.1 and 2.2 show the application of fuzzy sets with *trapezoidal membership functions*. Together with *triangular functions* (which are special cases of trapezoidal functions, as in Fig. 2.1 with $d = e$) they are the most popular membership functions in applications where smoothness of these functions is not necessary. If, however, differentiability is required as *e.g.*, during the design of neuro-fuzzy systems (see Section 2.1.4), then the most popular in applications are sigmoidal functions, bell (bell-like) functions or Gaussian functions, see *e.g.*, [58, 155, 111].

The one-sided sigmoidal membership function (of a set X_i) is defined by the formula

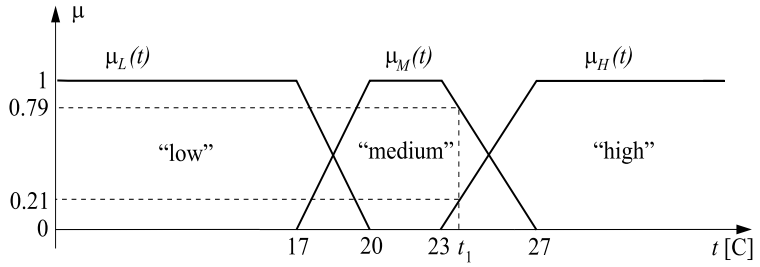

Fig. 2.2. Illustration of a linguistic variable "temperature" assuming three values: "low", "medium" and "high"

$$\mu_{X_i}(x) = \frac{1}{1 + \exp[-\alpha_i(x - c_i)]} \tag{2.1}$$

where parameters α_i and c_i define the shape of a fuzzy set X_i. For $\alpha_i < 0$ the function is left-sided open, whereas for $\alpha_i > 0$ it is right-sided open. The value c_i describes the positioning of the function ($\mu_{X_i}(c_i) = 0.5$), while α_i describes the steepness of its slope. One-sided sigmoidal functions can only define fuzzy sets corresponding to extreme values of linguistic variables. However, it is possible to build a two-sided (closed) sigmoidal membership function by the the use of two one-sided sigmoidal functions. Such a function, with its value approaching zero when $|x| \to \infty$, would represent intermediate values of a linguistic variable. The construction can be done in two simple ways: by subtracting two right-sided (or left-sided) open functions appropriately positioned in relation to each other, or by multiplying a right-sided open function by a left-sided open function positioned to its right (see *e.g.*, *Fuzzy Logic Toolbox* of the MATLAB® package). Figure 2.3 presents three sigmoidal membership functions:

- sigmf1: left-sided open with parameter values $\alpha_i = -30,\ c_i = 0.2$;
- sigmf2: two-sided created from subtracting function sigmf3 from a right-sided open function with parameter values $\alpha_i = 30,\ c_i = 0.2$;
- sigmf3: right-sided open with parameter values $\alpha_i = 15,\ c_i = 0.75$.

The popularity of sigmoidal functions results not only from the fact that it is easy to construct one-sided as well as two-sided sigmoidal functions, but also from the possibility to obtain asymmetry of two-sided functions – parameters of left and right slopes can be completely different and, at the same time, the size of the middle area (with a function value approximate to 1) can be shaped independently.

The generalized *bell membership function* is defined as follows [58, 155]

$$\mu_{X_i}(x) = \frac{1}{1 + [(\frac{x - c_i}{\alpha_i})^2]^{\beta_i}}, \tag{2.2}$$

where α_i, β_i, c_i are parameters defining the shape of a fuzzy set X_i. In particular, c_i defines the centre of symmetry, β_i influences mainly the inclination

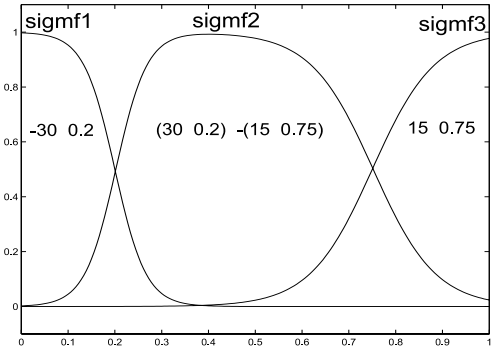

Fig. 2.3. Sigmoidal membership functions

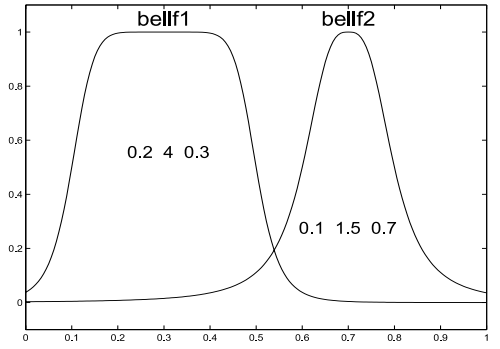

Fig. 2.4. Generalized bell membership functions

of slopes and α_i influences the width of the "bell". Figure 2.4 presents two examples of a bell function:

- bellf1: a function with parameter values $\alpha_i = 0.2,\ \beta_i = 4,\ c_i = 0.3$;
- bellf2: a function with parameter values $\alpha_i = 0.1,\ \beta_i = 1.5,\ c_i = 0.7$.

Let us note that the bell function is symmetric with respect to the vertical axis $x = c_i$. In order to obtain a two-sided asymmetric function, it should be built from two halves of bell functions with different parameters. In the literature and software packages Gaussian membership functions can be found as well, see *e.g.*, [58, 155, 111].

The linguistic variables (fuzzy variables) allow to capture roughly defined phenomena better than numerical variables, defined also as crisp variables. For example, the description of a linguistic variable "temperature", presented in Fig. 2.2 can result from a survey conducted among a representative sample of clients or communal users, *e.g.*, in order to establish conditions of turning

on heating appliances (when the temperature is low) or air-conditioning (when the temperature is high).

For a specific value of a numerical variable the process of defining its linguistic value together with the appropriate value of the membership function is called *fuzzyfication*. The example in Fig. 2.2 shows three linguistic values for the variable "temperature": "low", "medium" and "high", while in the process of fuzzyfication the linguistic values "medium" and "high", with the membership function values 0.79 and 0.21, respectively, are assigned to the fixed temperature numerical value t_1. Let us emphasize that a value of a linguistic variable is defined by a membership function assigned to this linguistic value, *i.e.*, it is identical with the fuzzy set defined by the corresponding membership function.

2.1.2 Fuzzy Reasoning

The basic element of a fuzzy system used *e.g.*, for modeling or control, is a *set of fuzzy inference rules* also known as a *knowledge base*, like in expert systems. It consists of inference rules operating on fuzzy (linguistic) variables, thus they are described as *fuzzy rules*.

Each inference rule consists of two elements: the IF-part, called an *antecedent* of a rule, and the THEN-part, also called a *consequent* of the rule. The structure of a single rule can thus be presented as follows:

$$\text{IF } <\text{antecedent}> \text{ THEN } <\text{consequent}>$$

The antecedent defines the condition, and the consequent – the conclusion which will be implemented if the condition is true.

The antecedent of a fuzzy rule consists, in the simplest case, of a single condition. Then the rule takes the following form:

$$\text{IF } x \text{ is } A \text{ THEN } <\text{consequent}>$$

where x is a linguistic variable, while A is a fuzzy set defined by a membership function $\mu_A(x)$. In the general case the rule antecedent can contain many simple conditions connected by the operators of conjunction (and), disjunction (or) and negation (not), *e.g.*,

$$\text{IF } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } (\text{not} A_2) \cdots \text{ or } x_k \text{ is } A_k \cdots \text{ THEN } <\text{consequent}>$$

The consequent of a fuzzy inference rule can, generally, take one of the following forms (*e.g.*, [155]):

1. Crisp consequent:

$$\text{IF } <\text{antecedent} > \text{ THEN } y = y_a$$

where y_a is a numerical value or a symbolic value,

2. Fuzzy consequent:

$$\text{IF } <\text{antecedent}> \text{ THEN } y \text{ is } Y_k$$

where y is a fuzzy variable (linguistic) and Y_k is a fuzzy set,

3. Functional consequent:

$$\text{IF } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \cdots \text{ and } x_n \text{ is } A_n \text{ THEN } y = f(x_1, x_2, \ldots, x_n)$$

where f is a certain function of variables $x_1, x_2, \ldots, x_n$.

In this book we shall be interested only in functional consequents proposed in 1985 by Takagi and Sugeno [131]. It turned out that using the inference rules with functional consequents enables effective modeling of nonlinear dependencies using a small number of rules. Fuzzy systems which use rules with functional consequents are called, from the names of the authors, the *Takagi-Sugeno fuzzy systems* or simply *TS fuzzy systems*. A term also sometimes used is that of Takagi-Sugeno-Kang (TSK) systems, as Kang was one of Professor Sugeno's younger co-workers developing fuzzy systems of that structure. One may also come across a term *multiple model systems*, see *e.g.*, [22]. Let us note that the functional consequent is reduced to a crisp one (numerical) when the function f assumes an extreme form of a constant, $f(x_1, x_2, \ldots, x_n) = a_0$. The most frequently applied functional consequents are those in the form of first order polynomials (affine functions)

$$f(x_1, x_2, \ldots, x_n) = a_0 + \sum_{i=1}^{n} a_i x_i$$

where $a_0, a_1, \ldots, a_n$ are parameters (polynomial coefficients). There are two reasons for this: the first and most important one is that using only affine functions is sufficient for a satisfactory precise modeling of even strongly nonlinear dependencies and with not a very large number of fuzzy sets in rule antecedents, provided a correctly selected number and location of these sets is chosen. Secondly, there exists a number of simple, well-established methods of construction and identification of linear models and of design of linear controllers which can be directly applied in individual sub-regions separated by antecedents of fuzzy TS system rules. Therefore, in this chapter we shall only use linear or affine functions of rule consequents, although the presented modeling and control structures can also be applied in a more general case with nonlinear functions.

Let us remember that a set of all inference rules of a fuzzy system, *e.g.*, of a fuzzy model of a nonlinear process, is called the *knowledge base* of that system. We shall assume that the knowledge base of a TS fuzzy system consists of rules in the following form

$$\mathrm{R}^i\text{: IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \cdots \text{ and } x_n \text{ is } A_n^i$$
$$\text{THEN } y^i = a_0^i + \sum_{j=1}^{n} a_j^i x_j \qquad (2.3)$$

$i = 1, ..., r$. Each fuzzy set A_j^i in the antecedent of a rule R^i is one of the elements of a set of linguistic values (fuzzy sets) of a variable x_j

$$A_j^i \in \mathbb{X}_j = \{X_{j1}, X_{j2}, ..., X_{jr_j}\} \qquad (2.4)$$

where r_j is a number of elements of the set $\mathbb{X}_j$, $j = 1, ..., n$. The presented structure of the knowledge base of the TS fuzzy system will be illustrated by way of an example.

Example 2.1
Let us consider modeling of a two-dimensional nonlinear dependence $y = f(x_1, x_2)$ by a TS fuzzy system, using a two-element representation of each of the linguistic variables x_1 and x_2

$$x_1 \in \{\text{“small”, “big”}\} = \{X_{1m},\ X_{1d}\}$$
$$x_2 \in \{\text{“small”, “big”}\} = \{X_{2m},\ X_{2d}\}$$

The knowledge base will be the set of four rules R^i, $i = 1, 2, 3, 4$:

$$\mathrm{R}^1 : \text{IF } x_1 \text{ is } X_{1m} \text{ and } x_2 \text{ is } X_{2m} \text{ THEN } y = y^1 = a_0^1 + a_1^1 x_1 + a_2^1 x_2$$
$$\mathrm{R}^2 : \text{IF } x_1 \text{ is } X_{1m} \text{ and } x_2 \text{ is } X_{2d} \text{ THEN } y = y^2 = a_0^2 + a_1^2 x_1 + a_2^2 x_2$$
$$\mathrm{R}^3 : \text{IF } x_1 \text{ is } X_{1d} \text{ and } x_2 \text{ is } X_{2m} \text{ THEN } y = y^3 = a_0^3 + a_1^3 x_1 + a_2^3 x_2$$
$$\mathrm{R}^4 : \text{IF } x_1 \text{ is } X_{1d} \text{ and } x_2 \text{ is } X_{2d} \text{ THEN } y = y^4 = a_0^4 + a_1^4 x_1 + a_2^4 x_2$$

where X_{1m}, X_{2m}, and X_{1d}, X_{2d} are fuzzy sets defining “small” and “large” values of the linguistic variables x_1 i x_2, while the functional consequents are linear approximations of the modeled dependence over appropriate two-dimensional fuzzy sets. By presenting the formulated above four general rules in the form (2.3), we get

$$A_1^1 = X_{1m}, \quad A_2^1 = X_{2m}$$
$$A_1^2 = X_{1m}, \quad A_2^2 = X_{2d}$$
$$A_1^3 = X_{1d}, \quad A_2^3 = X_{2m}$$
$$A_1^4 = X_{1d}, \quad A_2^4 = X_{2d}$$

where the sets A_1^i, $i = 1, ..., 4$, are elements of the set (2.4) taking the form

$$\mathbb{X}_1 = \{X_{11},\ X_{12}\} = \{X_{1m},\ X_{1d}\}$$

where $r_1 = 2$, while A_2^i, $i = 1, ..., 4$, are elements of the set (2.4) taking the form

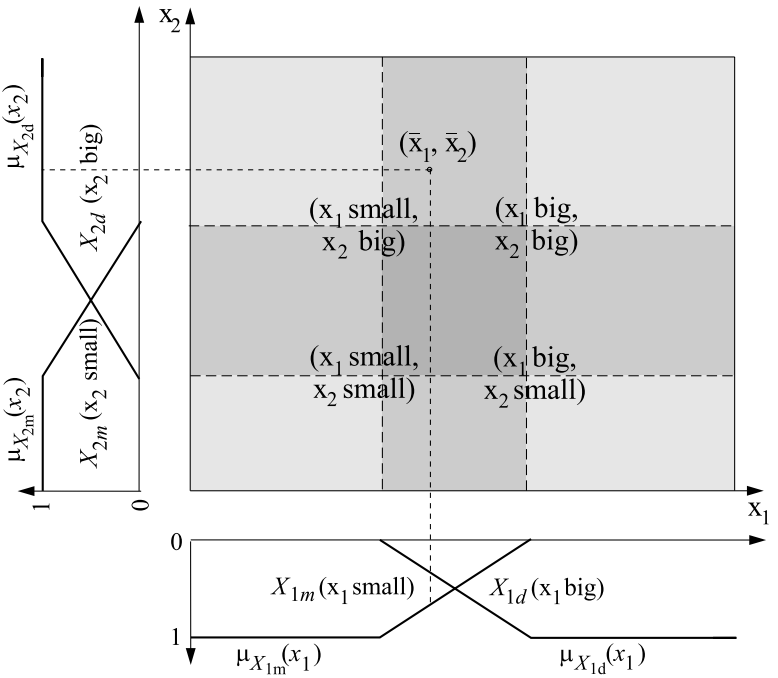

Fig. 2.5. A division of a two-dimensional domain in the case of two-element linguistic variables x and y

$$\mathbb{X}_2 = \{X_{21},\ X_{22}\} = \{X_{2m},\ X_{2d}\}$$

where $r_2 = 2$.

An example of the discussed domain of the modeled dependence is presented in Fig. 2.5. The point $(\bar{x}_1, \bar{x}_2)$ marked in Fig. 2.5 belongs, with non-zero values of the membership functions, to the sets $X_{1m} \times X_{2d}$ (x_1 small, x_2 big) and $X_{1d} \times X_{2d}$ (x_1 big, x_2 big). Particularly, the coordinate $\bar{x}_1$ belongs to the set X_{1m} with the membership function value $\mu_{X_{1m}}(\bar{x}_1) = 0.62$ and to the set X_{1d} with the membership function value $\mu_{X_{1d}}(\bar{x}_1) = 0.38$, while the coordinate $\bar{x}_2$ belongs only to set X_{2d}, with the membership function value $\mu_{X_{2d}}(\bar{x}_2) = 1.0$.

The presented example illustrates not only a set of rules, but it also shows how naturally Cartesian products of fuzzy sets are created. □

A set of rules is used for the fuzzy reasoning. Generally, the fuzzy reasoning consists of three basic stages and, for some applications, additionally of a fourth stage (see *e.g.*, [155]):

1. Calculation of *levels (degrees) of activation* of all rules, also described as *firing strengths* of the rules, see *e.g.*, [59], corresponding to current numerical values of input variables.

2. Evaluation of *conclusions* of individual rules.
3. Combining the conclusions of all rules into one *final conclusion.*
   In the general case the (output) variable of a rule consequent is a fuzzy variable, thus the final conclusion is fuzzy, *i.e.*, $y \in Y$, where y is an output variable of a fuzzy system and Y is a fuzzy set created as a result of stages 1, 2 and 3 of the fuzzy reasoning. For certain applications, *e.g.*, for control, the obtained fuzzy value of the output variable should be further transformed into a crisp numerical form – then the following is performed:
4. *Defuzzification* – transforming a fuzzy value of the output variable into a numerical value.

The fuzzy reasoning, and in particular its third stage, is much more simplified when consequents of all rules are not fuzzy, if they are crisp or functional. This situation occurs in the case of TS fuzzy systems, which are the subject of our interest in this book. Therefore, we shall restrict our attention to this case only – referring a reader interested in more general aspects of fuzzy reasoning to the vast literature on the subject, *e.g.*, [58, 155].

Fuzzy Reasoning in TS Structures

In the case of TS fuzzy structures fuzzy reasoning can be divided into the following three stages:

1. Calculation of *activation levels* of individual inference rules corresponding to the current numerical values of the input variables.
2. Calculation of values of *functional consequents* of all individual inference rules.
3. Calculation of the *final conclusion* value – weighted and normalized sum of values of the output variables obtained in the rule consequences by considering the activation levels of the individual rules.

In the considered case of the TS fuzzy system each inference rule R^i has the form (2.3), $i = 1, ..., r$, where r is the number of rules. Calculating the level of activation w^i of the i-th rule, even in the considered case of the antecedent containing only simple conditions all connected by the conjunction operator, is *not a unique operation.* The multiplication operator or the minimum operator are frequently used here. When using the *multiplication operator* for the input variable values $x = [x_1\ x_2\ \ldots\ x_n]^T$ the level of activation of the i-th rule is given by the *algebraic product*

$$w^i(x) = \mu_{A_1^i}(x) \cdot \mu_{A_2^i}(x) \cdot \ \cdots \ \cdot \mu_{A_n^i}(x), \quad i = 1, ...r$$

while with the application of the *minimum operator* the level of activation is given by the *logical product*

$$w^i(x) = \min\{\mu_{A_1^i}(x), \mu_{A_2^i}(x), \ldots, \mu_{A_n^i}(x)\}, \quad i = 1, ...r$$

In the two-dimensional Example 2.1 presented in the previous section, for the value $x = \bar{x} = [\bar{x}_1\ \bar{x}_2]^T$ of the vector of the input variables (see Fig. 2.5) we obtain the following values of activation levels of individual rules, when using the minimum operator:

$$
\begin{aligned}
w^1(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1m}}(\bar{x}_1) = 0.62,\ \mu_{X_{2m}}(\bar{x}_2) = 0\} = 0 \\
w^2(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1m}}(\bar{x}_1) = 0.62,\ \mu_{X_{2d}}(\bar{x}_2) = 1\} = 0.62 \\
w^3(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1d}}(\bar{x}_1) = 0.38,\ \mu_{X_{2m}}(\bar{x}_2) = 0\} = 0 \\
w^4(\bar{x}_1, \bar{x}_2) &= \min\{\mu_{X_{1d}}(\bar{x}_1) = 0.38,\ \mu_{X_{2d}}(\bar{x}_2) = 1\} = 0.38
\end{aligned}
$$

Further in this example, the following values of the output variable will be the values of the consequents of individual rules:

$$
\begin{aligned}
\mathrm{R}^1: &\quad \bar{y}^1 = a_0^1 + a_1^1\bar{x}_1 + a_2^1\bar{x}_2 \\
\mathrm{R}^2: &\quad \bar{y}^2 = a_0^2 + a_1^2\bar{x}_1 + a_2^2\bar{x}_2 \\
\mathrm{R}^3: &\quad \bar{y}^3 = a_0^3 + a_1^3\bar{x}_1 + a_2^3\bar{x}_2 \\
\mathrm{R}^4: &\quad \bar{y}^4 = a_0^4 + a_1^4\bar{x}_1 + a_2^4\bar{x}_2
\end{aligned}
$$

Let us note that using the product operator in the above example leads to the same result, which is a rather special case. Non-zero levels of activation calculated by minimum and product operators result in different values if at least two grades of membership occurring in a given rule are smaller than 1. But, directly from definitions of both operators, it is clear that zero levels of activation always occur for the same rules.

The last stage of fuzzy reasoning is a weighted, normalized sum of values of the output variables of consequents of all individual rules. For r rules described by (2.3) this is obtained according to the formula

$$
y = \frac{\sum_{i=1}^{r} w^i(x) \cdot y^i}{\sum_{l=1}^{r} w^l(x)} = \frac{\sum_{i=1}^{r} w^i(x)[a_0^i + \sum_{j=1}^{n} a_j^i x_j]}{\sum_{l=1}^{r} w^l(x)} \tag{2.5}
$$

where y^i are values of the consequents of individual rules at a point x, and y is the final conclusion – the value of the output variable of a TS fuzzy system at the point x. For the value $\bar{x}$ of the input variables in Example 2.1 considered above we have

$$
\begin{aligned}
\bar{y} &= \frac{\sum_{i=1}^{4} w^i(\bar{x}) \cdot \bar{y}^i}{\sum_{l=1}^{4} w^l(\bar{x})} = \\
&= 0.62 \cdot (a_0^2 + a_1^2\bar{x}_1 + a_2^2\bar{x}_2) + 0.38 \cdot (a_0^4 + a_1^4\bar{x}_1 + a_2^4\bar{x}_2)
\end{aligned}
$$

Let us note that in the above example $\sum_{l=1}^{r} w^l(x) = 1$ for each value of x, which results from the construction of the membership functions. Generally, this condition does not have to be satisfied, thus there is a normalizing sum in

the fraction denominator in the formula (2.5), consisting of values of activation levels of all rules. Values

$$\tilde{w}^i(x) = \frac{w^i(x)}{\sum_{l=1}^{r} w^l(x)}, \quad i = 1, ..., r \tag{2.6}$$

are called *normalized activation levels* of the inference rules of a fuzzy model. They always satisfy the condition $\sum_{i=1}^{r} \tilde{w}^i(x) = 1$.

From the point of view of the relations between the input variables x_1, x_2, ..., x_n and the output variable y, the TS fuzzy system can be treated as a functional mapping with features dependent on properties of the membership functions and rule consequent functions. If the membership functions are differentiable, as it is in the case of sigmoidal functions, the functions of the rule consequents are differentiable (they are usually affine) and, additionally, if levels of activation are defined by the product operator, then the nonlinear mapping generated by the TS fuzzy system will also be a continuous and differentiable mapping. In the two-dimensional case it is convenient to present the mapping of a TS fuzzy system in graphic form, as a surface $y = \varphi(x_1, x_2)$.

Example 2.2
Figure 2.6 presents the surface of a TS fuzzy system with two input variables x_1 and x_2, with fuzzy sets described by sigmoidal membership functions presented in Figures 2.7 and 2.8 and the following rule base of the general structure (2.3):

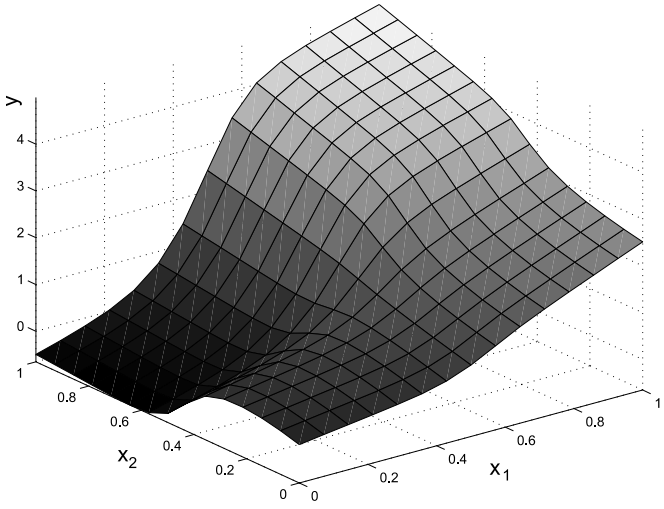

Fig. 2.6. Surface of the TS fuzzy system in Example 2.2

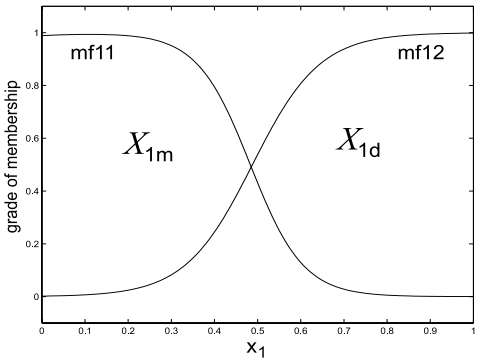

Fig. 2.7. Membership functions of fuzzy sets of the variable x_1

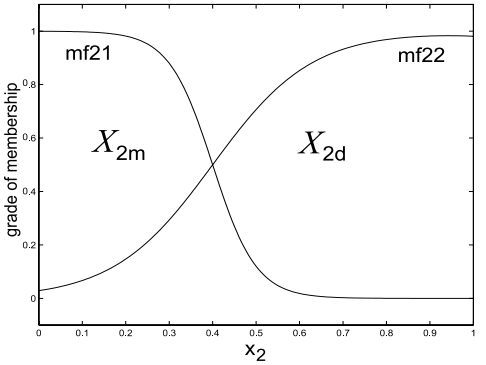

Fig. 2.8. Membership functions of fuzzy sets of the variable x_2

$$\begin{aligned}
\mathrm{R}^1: &\ \text{IF } x_1 \text{ is } X_{1m} \text{ and } x_2 \text{ is } X_{2m} \text{ THEN } y = 0.2 + x_1 + 2x_2 \\
\mathrm{R}^2: &\ \text{IF } x_1 \text{ is } X_{1m} \text{ and } x_2 \text{ is } X_{2d} \text{ THEN } y = -1 + 2x_1 + 0.5x_2 \\
\mathrm{R}^3: &\ \text{IF } x_1 \text{ is } X_{1d} \text{ and } x_2 \text{ is } X_{2m} \text{ THEN } y = 1 + 2x_1 + 2x_2 \\
\mathrm{R}^4: &\ \text{IF } x_1 \text{ is } X_{1d} \text{ and } x_2 \text{ is } X_{2d} \text{ THEN } y = -0.5 + 3x_1 + 1x_2
\end{aligned}$$

Let us note that in spite of simple affine consequents the obtained mapping is strongly nonlinear. □

2.1.3 Design of TS Fuzzy Models

Design of a TS fuzzy system, especially a fuzzy model of a known or unknown dependence between input and output variables, should be performed as follows:

1. *Fuzzy partitioning*: Dividing the range of variability of each input variable x_j into partly overlapping sets X_{jk}, $k = 1, ... r_j$, see (2.4), *i.e.*, defining the number of fuzzy sets and assigning a shape and values of parameters of the membership function of each set.
2. Defining the structure and parameters of functional consequents of individual rules. The number of rules r should be equal to the number of created sub-domains (multidimensional fuzzy subsets $A_1^i \times A_2^i \times \cdots \times A_n^i$, for the rules (2.3)) of the considered domain $X \in \mathbb{R}^n$ of input variables x_j, $j = 1, ..., n$, – one TS rule for each sub-domain.
3. Selecting the method of calculation of activation levels of individual rules: choosing the product or the minimum operator.

The first two points are the most important, the third one is a simple technical operation. A *proper selection of fuzzy sets* for input variables is a key to success. Too small a number of these sets and wrong positioning in relation to each other leads to unsatisfactory design of the fuzzy model, which does not satisfy the quality requirements and is too imprecise. Assuming too large a number of fuzzy sets leads to an oversized model with too many parameters; the design is then more difficult and the model is slower in operation.

There exist many proposals on how to perform the process of fuzzy partitioning in an automatic way, *e.g.*, using a data base consisting of a sufficiently representative set of values of input variables and corresponding output variables representing the modeled dependence. However, in practice it is still most efficient to take a human-made decision about the number of sets and their initial positioning, in an interactive mode, if necessary. This is precisely the place to employ empirical knowledge about the problem, about the character of nonlinearity. Human involvement is here not a drawback of the approach, on the contrary – it is its strong point, as the role of a human expert is well defined. Using expert knowledge is necessary to define a very important specific thing – proper general structure of a fuzzy system. The key is not to precisely define each membership function, but to define the number of these functions and their initial, approximate positioning. Precise tuning of parameters of membership functions is usually done automatically by using an appropriate optimization algorithm during a later phase of the design.

The way structure and parameters of functional consequents are defined depends on an application. Usually, polynomial functions of first order are employed. First of all, this is usually sufficient. Secondly, this leads to the possibility of simple and profitable interpretation of the fuzzy system as a neural network – as it will be presented later on. In a case of a fuzzy modeling the consequent functions will be affine approximations of the modeled dependence in individual sub-domains, *e.g.*, linearizations of this dependence at properly selected characteristic points of these sub-domains. In a case of the design of a fuzzy system as a controller, the functional consequents will be functions representing local control algorithms, designed for local models of the controlled process or directly for sets of data characterizing the process.

The selection of the type of the operator (minimum or product) for calculation of the activation levels of the rules is not very significant for the operation of a TS fuzzy system – it is arbitrary and dictated by the needs of the design method. The product operator is differentiable (the minimum operator is not), therefore it should be used if the nonlinear mapping represented by a TS fuzzy system should be differentiable. Especially, this situation occurs when membership function parameters are tuned in a learning mode of a fuzzy neural network.

2.1.4 TS System as a Fuzzy Neural Network

A TS fuzzy system can be presented in the form of a neural network structure, called a *fuzzy neural network* (FNN), or an ANFIS system (Adaptive Neuro-Fuzzy Inference System), see *e.g.*, [59, 58, 155]. Such a network can be interpreted as a multilayer perceptron in which nonlinear neuron nodes correspond to nonlinear membership functions. An example of the structure of a fuzzy neural network will be presented for a specific simple TS fuzzy system, where the domain of each of the two input variables x_1 and x_2 is divided into three fuzzy sets ($X_{11},\ X_{12}, X_{13}$ and $X_{21},\ X_{22},\ X_{23}$, respectively) defined by three sigmoidal membership functions: left-sided, two-sided and right-sided,

$$\mu_{X_{j1}}(x_j) = \frac{1}{1+\exp[-\alpha_{j1}(x_j - c_{j1})]},\ \ \alpha_{j1} < 0$$
$$\mu_{X_{j2}}(x_j) = \frac{1}{1+\exp[-\alpha_{j2+}(x_j - c_{j2+})]} - \frac{1}{1+\exp[-\alpha_{j2-}(x_j - c_{j2-})]}$$
$$\mu_{X_{j3}}(x_j) = \frac{1}{1+\exp[-\alpha_{j3}(x_j - c_{j3})]},\ \ \alpha_{j3} > 0$$

where $\alpha_{j2+} > 0,\ \alpha_{j2-} > 0,\ c_{j1} < c_{j2+} < c_{j2-} < c_{j3},\ \ j = 1, 2$. Let us consider a rule base consisting of three rules with antecedents containing only two simple conditions,

$$\mathrm{R}^1 : \text{IF}\ \ x_1 \text{ is } X_{11}\ \ \text{and}\ \ x_2 \text{ is } X_{21}\ \ \text{THEN}\ \ y^1 = f_1(x) = a_0^1 + a_1^1 x_1 + a_2^1 x_2$$
$$\mathrm{R}^2 : \text{IF}\ \ x_1 \text{ is } X_{12}\ \ \text{and}\ \ x_2 \text{ is } X_{23}\ \ \text{THEN}\ \ y^2 = f_2(x) = a_0^2 + a_1^2 x_1 + a_2^2 x_2$$
$$\mathrm{R}^3 : \text{IF}\ \ x_1 \text{ is } X_{13}\ \ \text{and}\ \ x_2 \text{ is } X_{22}\ \ \text{THEN}\ \ y^3 = f_3(x) = a_0^3 + a_1^3 x_1 + a_2^3 x_2$$

where $x = [x_1\ \ x_2]^T$. We are considering three rules only for simplicity of presentation, a full base would contain 9 rules. The multiplication (product) operator is used for assigning levels of activation

$$w^1(x) = \mu_{X_{11}}(x) \cdot \mu_{X_{21}}(x)$$
$$w^2(x) = \mu_{X_{12}}(x) \cdot \mu_{X_{23}}(x)$$
$$w^3(x) = \mu_{X_{13}}(x) \cdot \mu_{X_{22}}(x)$$

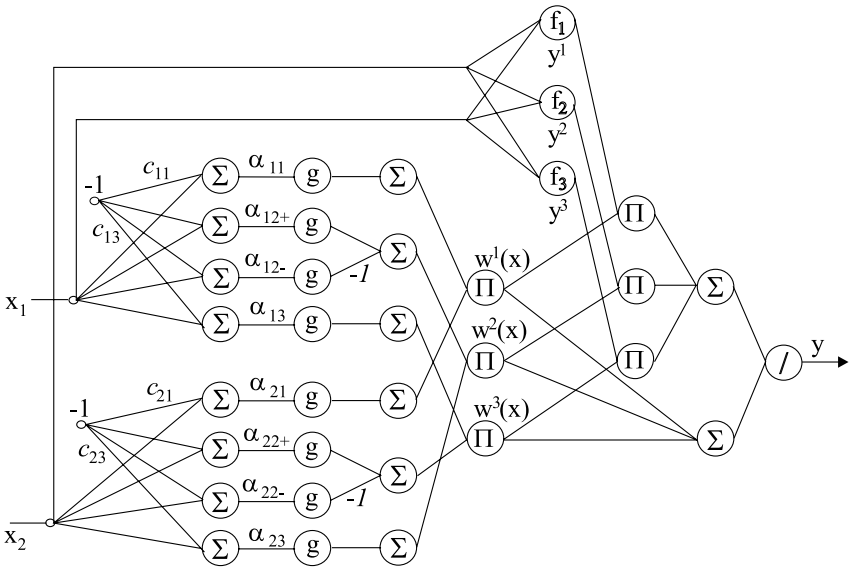

Fig. 2.9. Presentation of a TS fuzzy system, with two input variables and three rules, in the form of a fuzzy neural network (the case of constant parameters of functions f_i in the consequents)

while the system output is given according to the standard formula

$$y = \frac{\sum_{i=1}^{3} w^i(x) \cdot y^i}{\sum_{l=1}^{3} w^l(x)} = \sum_{i=1}^{3} \tilde{w}^i(x) \cdot y^i \tag{2.7}$$

where $\tilde{w}^i(x)$ denote normalized activation levels of the rules.

The structure of the described fuzzy neural network (FNN), for the case of constant (*i.e.*, not adjustable by the network) parameters of functions in the rule consequents, is presented in Fig. 2.9. The network consists of eight layers (not including the trivial layer consisting of input nodes):

- The first four layers are connected with the rule antecedents and they are responsible for calculation of activation levels $w^i(x)$ of individual rules. The first three layers calculate values of the membership functions, where nodes of the second layer model nonlinear functions of the type

$$g(z) = \frac{1}{1 + \exp[-z]}$$

while nodes of the fourth layer, denoted by "Π", are the multiplication nodes.

- Each node of the fifth layer calculates the value of the function $f_i(x)$ of one rule consequent; these calculations can be assigned to one node for each rule when coefficients of the functions are assumed to be constant (not tunable) network parameters.
- The sixth, seventh and eighth layers implement the final conclusion, according to (2.7).

In Fig. 2.9 the following *convention* is used: parameters c_{ji} and α_{ji} are placed near the arches, as close to arch centers as possible – they are tuneable network parameters, by which signals leaving the nodes are multiplied (in case of c_{ji} parameters, c_{j2+} and c_{j2-} were not shown due to lack of space, whereas c_{j3} were placed closer to the beginning of the arches, $j = 1, 2$). Network arches which do not have any assigned values should be treated as arches with the value "1". Concerning nodes transforming signals, the markings by these nodes denote node output signals.

A fuzzy neural network can be used to adjustment (optimization) of parameters α_{ji} and c_{ji} of the membership functions, by employing one of the standard network learning algorithms, see *e.g.*, [58, 155]. Certainly, it is possible in a situation when a set of input-output type learning data is available, in our example problem the data describing the dependence $x \rightarrow y$. By performing TS fuzzy modeling of a known nonlinear dependence one can generate such a set. However, it is a typical situation when a TS fuzzy model of an unknown dependence is constructed, based only on a set of input-output data. After the user has selected the number of fuzzy sets, initial shapes of membership functions and structure of rule consequents, it is possible to tune parameters of the membership functions as well as parameters of the consequent functions using methods of network learning. There are two methods available:

- We can extend the network replacing the function nodes "f_i" by additional network fragments presented in Fig. 2.10 – and then tune parameters of

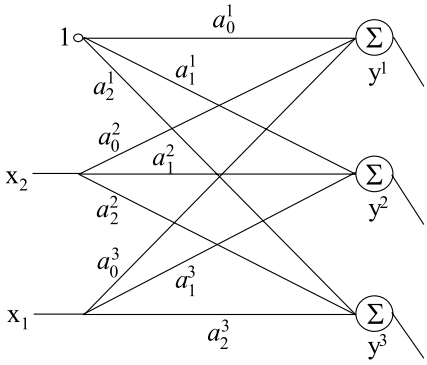

Fig. 2.10. Fragment of the network structure implementing functions of rule consequents

the membership functions and of the rule consequents by a single, selected method of network learning, *e.g.*, using a gradient optimization with gradient calculation according to the back propagation algorithm.

- A *hybrid optimization algorithm* can be used (see [58]): the parameters of the rule antecedents are tuned by a method of network learning (a gradient optimization using the back propagation algorithm to evaluate the gradient), alternately with tuning the parameters of the rule consequents by the least squares method. The authors of [58] have conducted comparative research which indicated better properties of the hybrid algorithm.

Example 2.3

The example will present results of fuzzy modeling of static characteristics of a distillation column shown in Fig. 2.11, used for separation of a two-component mixture of methanol and water. A dynamic model of the column was constructed [67], in the form of a set of differential and algebraic equations describing physical phenomena occurring on individual shelves of the column, in the reflux tank and in the bottom part of the column [81, 112, 46]. Models of controllers stabilizing liquid levels of the top and bottom products by manipulating outflows of the distillate D and of the bottom product B were then added.

Assuming all time derivatives in the model equations constant and equal to zero, a set of nonlinear algebraic equations was obtained, which describes static characteristics of the column. Input values (process control inputs) of

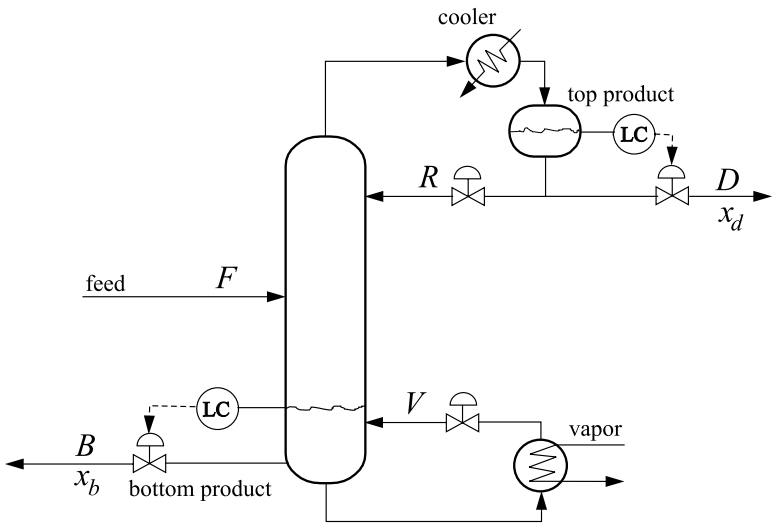

Fig. 2.11. Distillation column, Example 2.3

these characteristics are: the reflux (a stream of recycle from the reflux tank to the column) flow rate R and the heating steam flow rate V $[kmol/h]$. The outputs are concentrations x_d and x_b of the product (methanol) in the distillate D and in the bottom product B, respectively. Figure 2.12 presents original, nonlinear surfaces of the static characteristics of the column.

Starting to build a TS fuzzy model, a division of each of the domains of the input variables R and V into two partitions (two fuzzy sets) was assumed and sigmoidal membership functions were chosen (arbitrarily), as presented in Fig. 2.13 (a) and Fig. 2.14 (a). In this way, four fuzzy sub-domains (four two-dimensional fuzzy sets) were created in the domain of each of the characteristics, compare with Fig. 2.5 and with Example 2.1. In each of these sub-domains affine models for concentrations x_d and x_b were assumed,

$$x_d = a^i_{d0} + a^i_{d1}R + a^i_{d2}V, \quad i = 1, ..., 4$$
$$x_b = a^i_{b0} + a^i_{b1}R + a^i_{b2}V, \quad i = 1, ..., 4$$

For identification of parameters a^i_{dj} and a^i_{bj} of local models a set of data was created for modeling purposes: in 441 (21 x 21) evenly distributed points covering the entire range of variability of R and V the values of static characteristics (these shown in Fig. 2.12) were calculated. Next, parameters of the local models were tuned to this data using the method of minimization of the mean square deviation. The surfaces of TS fuzzy models obtained in this way are presented in Fig. 2.15.

Next, the model parameters were optimized using the hybrid method, performing alternately:

- steps modifying the parameters of the membership functions, in the direction of a negative gradient (calculated by the back propagation method)

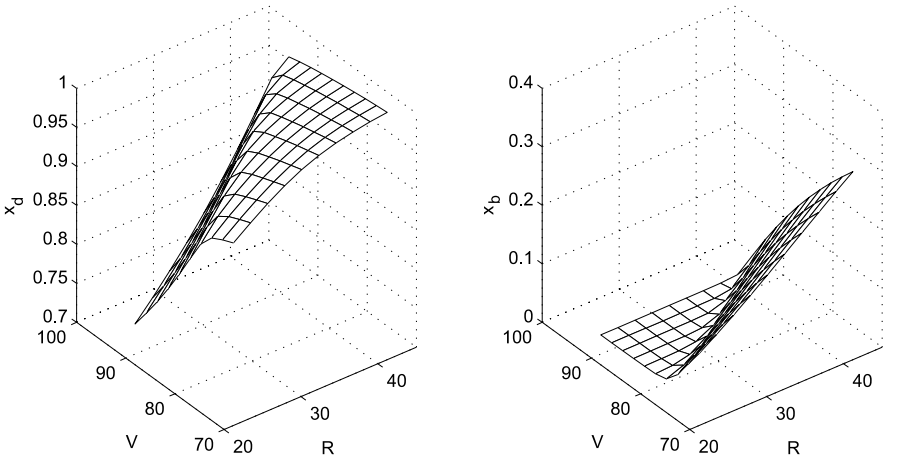

Fig. 2.12. Surfaces of the static characteristics of the plant, Example 2.3

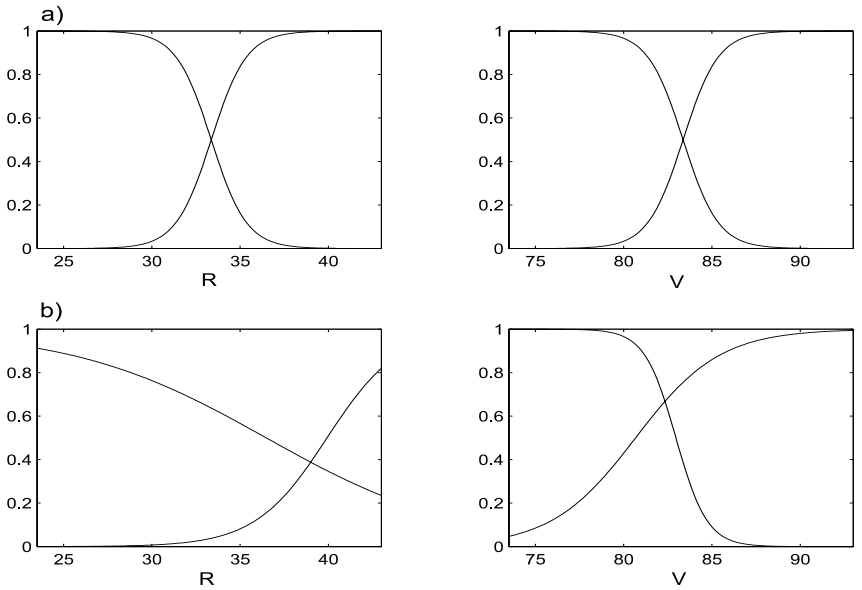

Fig. 2.13. Membership functions of fuzzy sets of the TS model of concentration x_d: a) initial, b) after parametric optimization

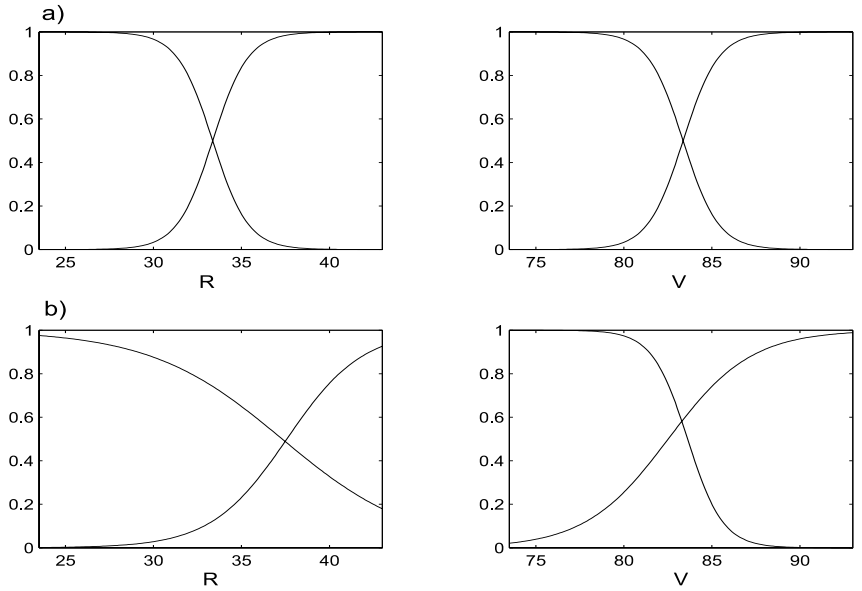

Fig. 2.14. Membership functions of fuzzy sets of the TS model of concentration x_b: a) initial, b) after parametric optimization

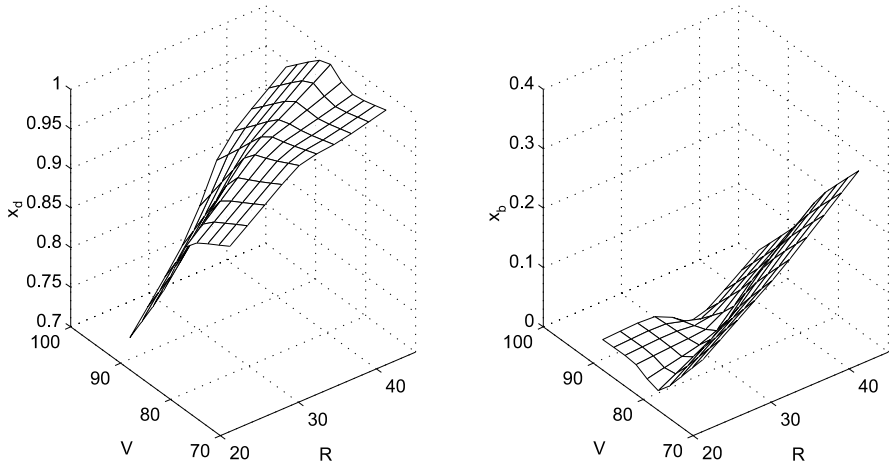

Fig. 2.15. Surfaces of TS fuzzy models of static characteristics of concentrations x_d and x_b, for initial values of parameters of membership functions

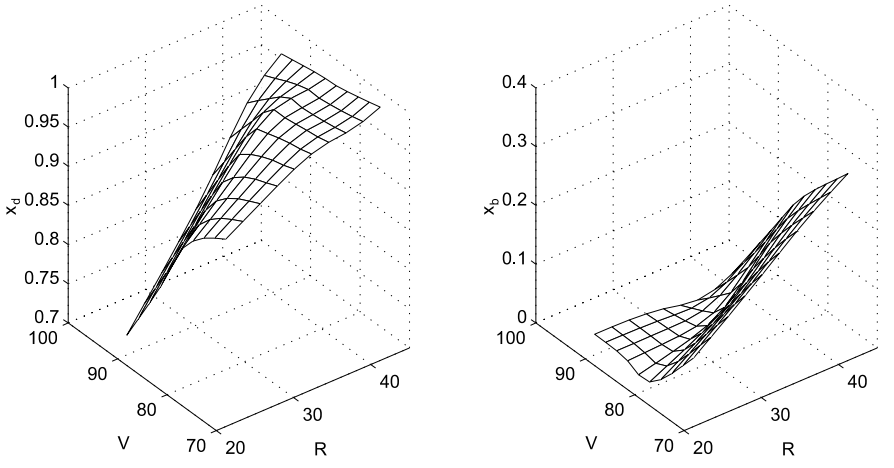

Fig. 2.16. Surfaces of TS fuzzy models of static characteristics of concentrations x_d and x_b, for optimal values of parameters of membership functions

of the performance function representing the mean square modeling error and

- steps tuning the parameters of the affine models (functions of the rule consequents), by the least squares method.

There were 500 iterations performed, after that the value of the modeling error was 8.45×10^{-3} for the model of x_d and 7.49×10^{-3} for the model of x_b.

Shapes of the membership functions obtained in this way are presented in Fig. 2.13 (b) and Fig. 2.14 (b), while surfaces of the fuzzy models are shown in Fig. 2.16. What draws our attention is a high modeling accuracy of the strongly nonlinear mapping obtained using a small number of fuzzy sets. □

2.2 Discrete-time TS Fuzzy Control

By a *TS fuzzy control algorithm* we shall describe a TS fuzzy system in which the consequent of each rule is a function defining a control algorithm, designed to control the process locally in a sub-domain (multidimensional local fuzzy set) associated with the rule antecedent. The presented approach is also called in the literature a *parallel distributed compensation* (PDC), see *e.g.*, [150], a *fuzzy multi-regional control algorithm* [33, 32] or a *multi-model control algorithm* [22].

Control algorithms of rule consequents of TS fuzzy controllers are generally designed locally, for local models describing the process in particular sub-domains. Generally, the local process models can be linear or nonlinear, as can the local control algorithms. In practice, linear local models and linear local control algorithms are applied, as they are sufficient for a nonlinear description if only the sub-domains are properly selected in relation to the nonlinearity of the process. When using a locally linear approach the design is very effective; it is possible to use known and practically well verified simple control algorithms. Thus, in this book, we shall restrict our attention to affine or linear local models and control algorithms. The structure of the design of rules of the TS fuzzy control algorithm in the case of a process described by a TS fuzzy model is presented in Fig. 2.17.

Design of a TS fuzzy controller can be divided into the following stages:

1. a) Define variables occurring in the rule antecedents. Values of these variables will describe the membership of the current state (operating point) of the process to *local sub-domains*, in which the process can be approximated by affine models, for control purposes.
   b) Divide the range of variability of each variable x_j of the rule antecedents into overlapping fuzzy sets X_{jk} (in relation to the process nonlinearity), defining the number r_j (see (2.4)) of these sets and assigning a shape and values of parameters of the membership function of each set. This creates a division of the whole domain into a number of overlapping sub-domains (multidimensional fuzzy sets $A_1^i \times A_2^i \times \cdots \times A_n^i$, for the rules (2.3)).
   c) Formulate an affine or linear process model for each sub-domain constructed so far (each sub-domain is defined by one fuzzy rule, see Example 2.2).

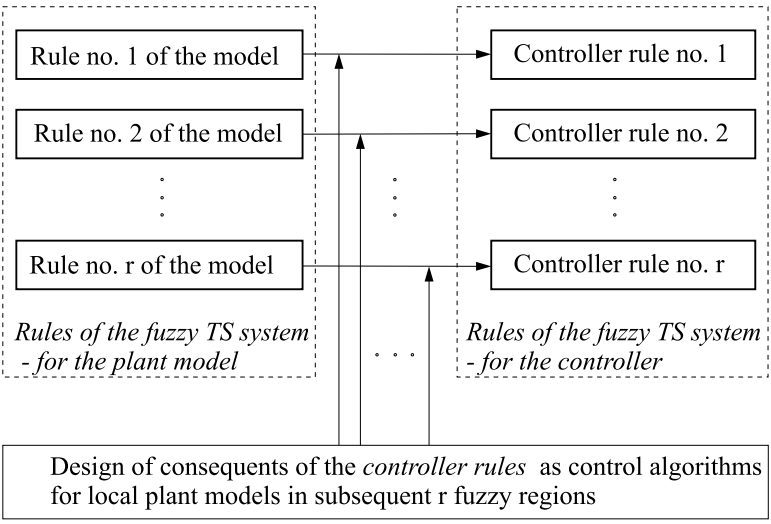

Fig. 2.17. Design of a TS fuzzy controller for a process modeled by a TS fuzzy system

The procedure defined above means a design of a TS fuzzy model of the considered nonlinear control process, with functional consequents which are linear dynamic models of the process in particular local sub-domains. The range of variability of the input and output signals should cover the entire operation domain of the designed controller.

2. For each local affine model (sub-model) of the process *design a linear controller* operating correctly in the assigned sub-domain, using a selected technique appropriate to the applied process model, *e.g.*, a state-feedback controller for models in the form of state equations, a PID controller for ARX models, *etc.* Create a TS fuzzy controller – a TS fuzzy system with rule antecedents created in the previous point (division into fuzzy sets as in the TS fuzzy model of the process) and functional consequents which are the designed control algorithms.

3. Perform *analysis* of the obtained TS fuzzy algorithm using *simulation and theoretical analysis*: if it is possible, check analytically whether stability conditions are satisfied and simulate the operation of the closed-loop control system in predicted conditions. If results are not satisfactory, return to the previous point and correct parameters of local controllers. Repeat the simulation and theoretical analysis of the obtained control system. If there are difficulties in obtaining the desired quality of control by means of tuning the local controllers only, return to the first stage: modify the fuzzy sets of the rule antecedents, *i.e.*, the parameters of the membership

functions, or even the number of partitions (number of fuzzy sets) into which the ranges of individual variables occurring in the rule antecedents were divided. Repeat the following design and analysis of the corrected control system.

It should be noted that the last point is significant. Properties of a TS fuzzy controller are not a simple collection of properties of local controllers. There are examples known, see *e.g.*, [133], when in spite of the stability of all local controllers, the TS fuzzy nonlinear controller is unstable.

Increasing the reliability and power of computer systems together with strong competition on the market of industrial products unavoidably lead to the application of multilayer control with on-line optimization of the operating points. Thus, in many loops of the direct control layer it is necessary to consider applications of nonlinear controllers, because in the situation of current, automatic changes in values of the set-points and significant nonlinearities, the frequently occurring PID linear controllers do not ensure adequate control quality. In such cases they have to be tuned for the worst case, *i.e.*, for operating points critical for stability and robustness – thus they operate too slowly in less critical conditions. The TS fuzzy controllers, discussed in this chapter, are appropriate not only for the set-point control layer (where sampling periods are usually larger) but they can be widely used for the direct feedback control. Moreover, if functions in the rule consequents of a TS fuzzy controller are linear, *e.g.*, describing the classical PID control algorithm, then this controller is a *natural nonlinear generalization of a linear PID controller*, a generalization which should be easily understood and accepted by the staff operating the control system.

The time domain is natural for the TS fuzzy systems used for process modeling or for constructing dynamic systems such as controllers, because conditions occurring in the rule antecedents of a process model or of a controller are formulated for variables in the time domain. Therefore, in functional consequents of fuzzy rules there occur affine or linear dynamic models which define values of output variables of a process or a controller at a given moment. A standard, practical approach when selecting settings of an industrial PID controller is the use of a largely simplified process model, mainly in the form of a simple transfer function, obtained experimentally on the basis of an output response to a step function or to a rectangular impulse – and reading the appropriate settings from predefined tables, see *e.g.*, [38, 44]. There is nothing wrong with the application of this method for the design of a fuzzy PID controller, only local process models corresponding to individual local fuzzy sets (sub-domains) should be presented in the form of appropriate transfer functions and used for selection of the local PID settings.

The process of tuning the parameters of the local controllers, or even jointly parameters of the controllers and the membership functions, can also be done globally and in an automatic way – if we have a credible simulation model of the controlled process at our disposal. Then, we can use optimization

of a function formulated as an appropriate control quality index, usually also with constraints, *e.g.*, on the overshoot. On the other hand, if we have a set of data representing desired behavior of the control system, then by presenting the designed TS fuzzy controller in the form of a fuzzy neural network, it is possible to apply an algorithm for optimization (teaching) of such a network (see Section 2.1.4).

The discussion presented above concerns general problems of the design of nonlinear TS fuzzy controllers. In the current state of technology these controllers are implemented in microcomputers, so they are discrete (digital) controllers. However, if the sampling period is small enough in relation to the dominant process time constant, then a fully authorized method of the design is the synthesis of a continuous control algorithm – and then application of its digital representation only, see *e.g.*, [44, 52]. For that reason, after presenting design of discrete-time TS fuzzy controllers in the following sections, we shall discuss the same problem for the continuous-time case.

2.2.1 Discrete TS Fuzzy State-feedback Controllers

The first case of a TS fuzzy controller discussed in the literature was a state-feedback controller [133, 150], based on a TS fuzzy model with consequents in the form of linear state equations. It was only later that output-feedback controllers were proposed, based on input-output type models.

Let us consider a TS fuzzy discrete model of a dynamic process with the rules of the following form:

$$\begin{aligned} \mathrm{R}_p^i: \quad \text{IF} \quad & x_1(k) \text{ is } A_1^i \text{ and } x_2(k) \text{ is } A_2^i \text{ and } \cdots \text{ and } x_{n_x}(k) \text{ is } A_{n_x}^i \\ & \text{THEN} \quad x^i(k+1) = \mathbf{A}_i x(k) + \mathbf{B}_i u(k) \end{aligned} \tag{2.8}$$

where

$$\begin{aligned} x(k) &= [x_1(k)\ x_2(k)\ \cdots\ x_{n_x}(k)]^T \\ u(k) &= [u_1(k)\ u_2(k)\ \cdots\ u_{n_u}(k)]^T \end{aligned}$$

are state and control input vectors of a dynamic process at sampling instant k, respectively, $A_j^i \in \mathbb{X}_j = \{X_{j1}, ..., X_{jr_j}\}$ are fuzzy sets of the coordinate x_j of the state vector x (see (2.4)), $j = 1, \ldots, n_x$, while $\mathbf{A}_i$ and $\mathbf{B}_i$ are the state and control matrices of local linear models created for local fuzzy sets (sub-domains), $i = 1, ..., r$.

Using a *grid partition* of the domain of the input variables [59] (see the partition presented in Fig. 2.5 as an example), we obtain a fuzzy system with the number of sub-domains (and rules) equal to

$$r = r_1 \cdot r_2 \cdot \ \cdots \ \cdot r_{n_x} = \prod_{j=1}^{n_x} r_j$$

As a result of this, the number of local fuzzy sets grows rapidly with the increase of dimensionality n_x of the state vector x and the number r_j of fuzzy sets corresponding to each coordinate x_j of the state vector. Thus, it is important that the domains of individual state variables are divided into as small a number of fuzzy sets as possible, *i.e.*, that the numbers r_j are small. This is possible, without losing modeling accuracy, by using the TS structures - as opposed to classic fuzzy modeling, in which the rule consequents are fuzzy sets.

For given values of the state and control input vectors, $x(k)$ and $u(k)$, the output of a fuzzy model, *i.e.*, the state at the next sampling instant, is calculated according to the general formula (2.5), *i.e.*,

$$x(k+1) = \frac{\sum_{i=1}^{r} w^i(k)[\mathbf{A}_i x(k) + \mathbf{B}_i u(k)]}{\sum_{l=1}^{r} w^l(k)} \tag{2.9}$$

where $w^i(k)$ are levels of activation of individual rules (2.8) at $k-$th sampling instant,

$$w^i(k) = \prod_{j=1}^{n_x} \mu_{A_j^i}(x_j(k))$$

We have taken above the natural assumption that always $\sum_{l=1}^{r} w^l(k) > 0$, *i.e.*, for each value of the state variables from their domain the model is well defined – at least one rule is activated. It is convenient to operate with normalized rule activation levels, see (2.6), which from definition satisfy the condition $\sum_{i=1}^{r} \tilde{w}^i(k) = 1$. Then the model of an autonomous process (with $u(k) \equiv 0$)) takes the form

$$x(k+1) = \sum_{i=1}^{r} \tilde{w}^i(k) \mathbf{A}_i x(k) \tag{2.10}$$

For such a dynamic model, using directly the Lyapunov theorem (see *e.g.*, [61, 148]), it is relatively easy to derive a sufficient condition of asymptotic stability [133].

Theorem 2.1 *The equilibrium point of a dynamic system (2.10) described by the rules (2.8) with $u(k) = 0$ is globally asymptotically stable, if there exists a symmetric and positive definite matrix $\mathbf{P}$, such that for the state matrix $\mathbf{A}_i$ of each local model the following inequality is satisfied*

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < \mathbf{0}, \quad i = 1, 2, ..., r \tag{2.11}$$

Proof. Let us formulate a scalar function of the form

$$V(x(k)) = x^T(k) \mathbf{P} x(k) \tag{2.12}$$

where $\mathbf{P}$ is a symmetric positive definite matrix. The function (2.12) satisfies the following conditions:

$$\begin{aligned} V(0) &= 0 \\ V(x(k)) &> 0 \ \text{ for } \ x \neq 0 \\ V(x(k)) &\to \infty \ \text{ for } \ \|x(k)\| \to \infty \end{aligned}$$

Moreover, a change of its value along a trajectory of the dynamic system (2.10) is defined by the formula

$$\begin{aligned} \Delta V(x(k)) &= V(x(k+1)) - V(x(k)) \\ &= x^T(k+1)\mathbf{P}x(k+1) - x^T(k)\mathbf{P}x(k) \\ &= x^T(k)\left[\sum_{i=1}^{r}\tilde{w}^i(k)\mathbf{A}_i^T\mathbf{P}\sum_{j=1}^{r}\tilde{w}_j(k)\mathbf{A}_j - \mathbf{P}\right]x(k) \\ &= \sum_{i=1}^{r}\sum_{j=1}^{r}\tilde{w}^i(k)\tilde{w}^j(k)x^T(k)[\mathbf{A}_i^T\mathbf{P}\mathbf{A}_j - \mathbf{P}]x(k) \\ &= \sum_{i=1}^{r}(\tilde{w}^i(k))^2x^T(k)[\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P}]x(k) + \\ &+ \sum_{i=1}^{r}\sum_{j<i}^{r}\tilde{w}^i(k)\tilde{w}^j(k)x^T(k)[\mathbf{A}_i^T\mathbf{P}\mathbf{A}_j + \mathbf{A}_j^T\mathbf{P}\mathbf{A}_i - 2\mathbf{P}]x(k) \end{aligned}$$

We have

$$\begin{aligned} \mathbf{A}_i^T\mathbf{P}\mathbf{A}_j + \mathbf{A}_j^T\mathbf{P}\mathbf{A}_i - 2\mathbf{P} = &-(\mathbf{A}_i - \mathbf{A}_j)^T\mathbf{P}(\mathbf{A}_i - \mathbf{A}_j) \\ &+ [\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P}] + [\mathbf{A}_j^T\mathbf{P}\mathbf{A}_j - \mathbf{P}] \end{aligned}$$

and it follows from the positive definiteness of the matrix $\mathbf{P}$ that

$$-(\mathbf{A}_i - \mathbf{A}_j)^T\mathbf{P}(\mathbf{A}_i - \mathbf{A}_j) \leq \mathbf{0}$$

Moreover,

$$\tilde{w}^i(k) \geq 0 \quad \text{for every } i = 1, 2, ..., r$$

thus $\Delta V(x(k)) < 0$ for $x(k) \neq 0$ if only $\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P} < \mathbf{0}$ for every $i = 1, 2, ..., r$. Then the function $V(x(k))$ is a Lyapunov function of the nonlinear dynamic system, which concludes the proof. □

It should be emphasized that the above theorem requires that only one matrix $\mathbf{P}$ common for all local linear dynamic models $x^i(k+1) = \mathbf{A}_i x(k)$ satisfies the stability condition. To find a solution (or to prove it does not exist) of the system of linear matrix inequalities

$$\begin{aligned} \mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P} < 0, \quad i = 1, 2, ..., r \\ \mathbf{P} > 0, \ \ \mathbf{P} \text{ symmetric} \end{aligned} \tag{2.13}$$

is not difficult now; one can use procedures from the *LMI Toolbox* of the MATLAB® package (LMI - Linear Matrix Inequalities).

It is worth mentioning here that for nonsingular matrices $\mathbf{A}_i$, *a necessary condition for the existence of a matrix* $\mathbf{P}$ *satisfying the Theorem 2.1 is that each of the product matrices* $\mathbf{A}_i\mathbf{A}_j$, $i, j = 1, 2, ..., r$ *is a matrix of an asymptotically stable system, i.e.*, that the modulus of any of its eigenvalues is smaller than 1 [133]. Namely, it follows from the condition (2.11) that

$$\mathbf{P} - (\mathbf{A}_j^{-1})^T\mathbf{P}(\mathbf{A}_j)^{-1} < 0$$

Adding the above inequality and (2.11) results in

$$\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - (\mathbf{A}_j^{-1})^T\mathbf{P}(\mathbf{A}_j)^{-1} < 0$$

which can be converted to the form

$$\mathbf{A}_j^T\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i\mathbf{A}_j - \mathbf{P} < 0, \quad i = 1, 2, ..., r$$

It follows from the last inequality that the matrix $\mathbf{A}_i\mathbf{A}_j$ must be asymptotically stable. Thus, proving that one of the matrices $\mathbf{A}_i\mathbf{A}_j$ is not asymptotically stable determines non-existence of the matrix $\mathbf{P}$ satisfying the conditions of Theorem 2.1.

Recently, a version of the Theorem 2.1 with weakened conditions has been obtained [151], for dynamic systems with trajectories of limited transitions between different subsets of the whole operating space. To present these results, the whole process operating state-space must be additionally partitioned into subsets of *constant activation of fuzzy rules* [60, 151], which in the sequel will be called *constant activation cells*, or *activation cells*, for brevity. The activation cells will be denoted by S_l, $l = 1, ..., L$, where L is the number of all cells. Within each cell activation level of every fuzzy rule is either zero (inactive rule) or positive (active rule - its activation level may be zero only on the cell boundary).

The constant activation cells may be divided into those where the process is in an *operating regime* and those corresponding to *interpolating regimes*. The cell is with an operating regime when $\tilde{w}^l(x(k)) = 1$ for a certain $l \in L$ and all other normalized activation levels are equal to zero. Thus, the system dynamics is then fully defined by a local linear model being the consequent of the fuzzy rule defining the fuzzy sub-domain containing this cell. On the other hand, activation cells with interpolating regimes are defined as those where $0 < \tilde{w}^l(x(k)) < 1$ for $x(k) \in \text{int}S_l$ (interior of S_l) for at least two $l \in L$ (at least two, because $\sum_{l \in L} \tilde{w}^l(x(k)) = 1$ for every sampling instant k). Notice that all activation cells are crisp and not overlapping sets, creating a partition of the overall process state-space which is closely related to the partition of this space into fuzzy sub-domains defined so far (see the beginning of Section 2.2), but different, containing more elements.

An example of a partition of the problem state-space into fuzzy sub-domains and into cells of constant activation of fuzzy rules is shown in Fig. 2.18, for a TS fuzzy model consisting of the following three rules

$$\begin{aligned}
\mathrm{R}_p^1: &\ \text{IF} \quad x_1(k) \text{ is } X_1 \quad \text{THEN} \quad x(k+1) = A_1 x(k) + B_1 u(k) \\
\mathrm{R}_p^2: &\ \text{IF} \quad x_1(k) \text{ is } X_2 \quad \text{THEN} \quad x(k+1) = A_2 x(k) + B_2 u(k) \\
\mathrm{R}_p^3: &\ \text{IF} \quad x_1(k) \text{ is } X_3 \quad \text{THEN} \quad x(k+1) = A_3 x(k) + B_3 u(k)
\end{aligned}$$

where $x = (x_1, x_2)$, $0 \le x_2 \le 3$, X_i are fuzzy sets defined by membership functions $\mu_{X_i}(x_1)$, $i = 1, 2, 3$, and 3 two-dimensional fuzzy sub-domains are defined by Cartesian products $X_i \times [0, 3]$, $i = 1, 2, 3$.

Note that, to preserve strict mathematical correctness, the partitioning of the process state-space into activation cells is possible only for trapezoidal (or trapezoidal-like) membership functions, as sigmoidal, gaussian or generalized bell functions have positive activation levels over the whole universe of discourse.

Denoting, for each activation cell S_l, by $K(l)$ a set containing indexes of all fuzzy rules with nonzero activation level over S_l (for a cell S_l with operating regime its corresponding $K(l)$ contains a single element), we can describe the autonomous (control free) fuzzy system dynamics as follows

$$x(k+1) = \sum_{i \in K(l)} \tilde{w}^i(k)[\mathbf{A}_i x(k)], \quad x(k) \in S_l, \quad l \in L \tag{2.14}$$

where $0 < \tilde{w}^i(k) = \tilde{w}^i(x(k)) \le 1$ for each $i \in K(l)$ and $\sum_{i \in K(l)} \tilde{w}^i(k) = 1$.

Define also, for further use, a set Ω representing *all possible one-step system state transitions* between different constant activation cells S_l,

$$\Omega = \{(l, m): \ x(k) \in S_l, \ x(k+1) \in S_m, \ \text{for every } l, m \in L\} \tag{2.15}$$

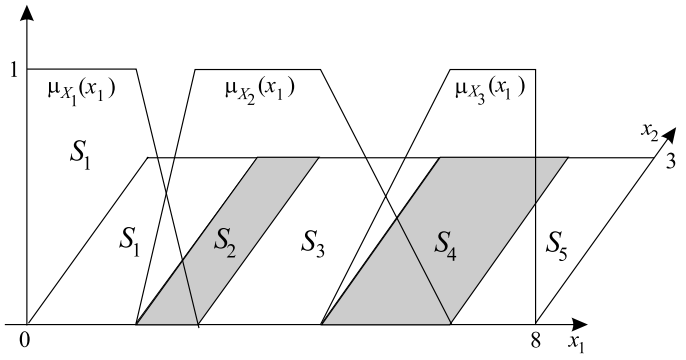

Fig. 2.18. Membership functions $\mu_{X_i}(x_1)$ defining 3 fuzzy sets X_i, 3 fuzzy sub-domains $X_i \times [0, 3]$ and corresponding 5 constant activation cells: 3 with operating regimes (S_1, S_3, S_5 – white) and 2 with interpolating regimes (S_2, S_4 – grey)

where $m \neq l$ when the system transits (in one step) from a cell S_l to the cell S_m, whereas $m = l$ when the system stays in the same cell S_l. We are now in a position to formulate a version of Theorem 2.1 with weakened stability conditions, due to the use of a piecewise-quadratic Lyapunov function, instead of a quadratic one.

Theorem 2.2 [151] *The equilibrium point of a dynamic system (2.14) described by the rules (2.8) with $u(k) \equiv 0$ is globally asymptotically stable, if there exist L symmetric and positive definite matrices $\mathbf{P}_l$, $l \in L$ such that the following set of inequalities is satisfied*

$$\mathbf{A}_i^T \mathbf{P}_m \mathbf{A}_i - \mathbf{P}_l < 0, \quad \text{for every } (l,m) \in \Omega, \quad i \in K(l) \tag{2.16}$$

Proof. The reasoning is similar as in the proof of Theorem 2.1, only the system dynamics described by (2.14) and the following piecewise-quadratic Lyapunov function candidate

$$V(k) = x(k)^T \mathbf{P}_l x(k), \; x(k) \in S_l, \; l \in L \tag{2.17}$$

must be used, instead of the function (2.12). □

Theorems 2.1 and 2.2 state only sufficient conditions for stability – if these conditions are not satisfied then the question of stability remains open. According to the best knowledge of the author, necessary and sufficient stability conditions have yet not been formulated. Some attempts were made to obtain a formulation of sufficient stability conditions on the basis of considering properties of local state matrices $\mathbf{A}_i$ only, without the necessity of solving global or partitioned sets of inequalities, such as those in (2.11) or (2.16). In [20] conditions of this type were given, based on a direct estimate of the state norm of the system described by (2.10)

$$\begin{aligned} \|x(k+1)\| &= \|\sum_{i=1}^{r} \tilde{w}^i(k) \cdot \mathbf{A}_i x(k)\| \\ &\leq \sum_{i=1}^{r} \tilde{w}^i(k) \|\mathbf{A}_i\| \|x(k)\| \\ &\leq \max_{1 \leq i \leq r} \|\mathbf{A}_i\| \sum_{i=1}^{r} \tilde{w}^i(k) \|x(k)\| \\ &= \max_{1 \leq i \leq r} \|\mathbf{A}_i\| \cdot \|x(k)\| \end{aligned} \tag{2.18}$$

where $\|\mathbf{A}\|$ denotes a norm of the matrix $\mathbf{A}$, induced by the vector norm. Thus satisfaction of the condition

$$\sum_{i=1}^{r} \tilde{w}^i(k) \|\mathbf{A}_i\| < 1 \tag{2.19}$$

or the slightly stronger condition

$$\|\mathbf{A}_i\| < 1, \quad i = 1, 2, ..., r \tag{2.20}$$

ensures asymptotic stability of the dynamic system (2.10). The above conditions turn out to be very conservative in practice, and as such, of not much use.

The example given below, after [133], shows that stability of individual linear subsystems $x(k+1) = \mathbf{A}_i x(k)$ is generally not sufficient for the stability of the overall nonlinear fuzzy system. Thus, in order to ensure stability, additional conditions such as that in Theorem 2.1 or Theorem 2.2, are necessary.

Example 2.4
Let us consider fuzzy sets X_1 and X_2 presented in Fig. 2.19 and a TS fuzzy model consisting of the following two rules

$$\begin{aligned} \mathrm{R}_p^1: &\ \text{IF} \quad x(k-1) \text{ is } X_1 \quad \text{THEN} \quad x^1(k+1) = x(k) - 0.5x(k-1) \\ \mathrm{R}_p^2: &\ \text{IF} \quad x(k-1) \text{ is } X_2 \quad \text{THEN} \quad x^2(k+1) = -x(k) - 0.5x(k-1) \end{aligned}$$

which are equivalent to the rules (2.8) with state matrices

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -0.5 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}$$

corresponding to the state definition

$$[\, x_1(k)\ x_2(k) \,] = [\, x(k)\ x(k-1) \,]$$

Eigenvalues of matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are $0.5 \pm 0.5i$ and $-0.5 \pm 0.5i$, respectively, thus the matrices describe asymptotically stable systems. But, starting the fuzzy system from an initial point $x_1 = 0.9,\ x_2 = -0.7$ we obtain a trajectory shown in Fig. 2.20, definitely indicating non-stability of the fuzzy nonlinear model. Assumptions of Theorem 2.1 can not be satisfied in this case. Indeed, it is easy to check that matrix $\mathbf{A}_1\mathbf{A}_2$,

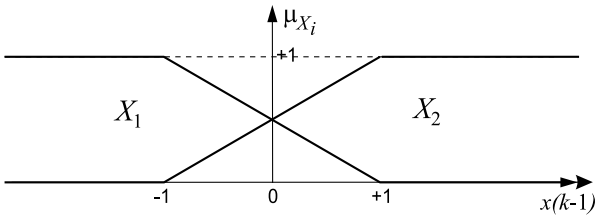

Fig. 2.19. Fuzzy sets of the model, Example 2.4

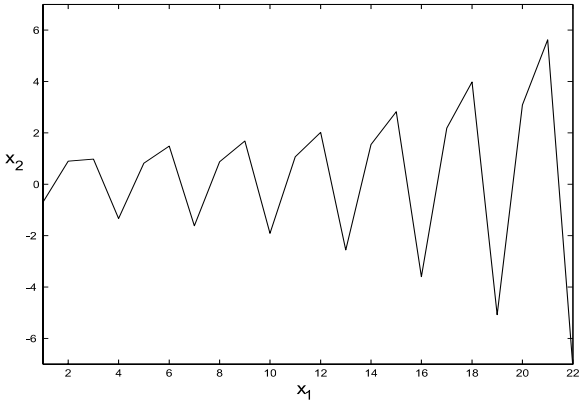

Fig. 2.20. Unstable state trajectory in Example 2.4

$$\mathbf{A}_1\mathbf{A}_2 = \begin{bmatrix} -1.5 & -0.5 \\ -1 & -0.5 \end{bmatrix}$$

has eigenvalues equal to $\lambda_1 = 0.134$, $\lambda_2 = 1.866$, thus it is not a matrix of a stable discrete dynamic system – therefore, the matrix $\mathbf{P}$ satisfying assumptions of Theorem 2.1 does not exist. □

For each local linear dynamic model from the consequents of the rules (2.8), that is for each of the sub-domains of a fuzzy model, a linear state-feedback controller can be designed using a standard method. In this way we obtain the rules of a TS fuzzy controller in the following form

$$\mathrm{R}_c^j: \quad \text{IF} \quad x_1(k) \text{ is } A_1^j \text{ and } x_2(k) \text{ is } A_2^j \text{ and} \cdots \text{ and } x_{n_x}(k) \text{ is } A_{n_x}^j \quad \text{THEN} \quad u^j(k) = -\mathbf{F}_j x(k) \tag{2.21}$$

where $\mathbf{F}_j$ are matrices of state-feedback coefficients, $j = 1, 2, ..., r$. A complete nonlinear TS fuzzy controller will be described by the following relation

$$u(k) = -\sum_{j=1}^{r} \tilde{w}^j(k)\mathbf{F}_j x(k) \tag{2.22}$$

The structure of a control system with such a controller is presented in Fig. 2.21.

When modeling and simulating the feedback control system we can, generally, use for process representation in the closed-loop a nonlinear process model different than the one used for the design of the fuzzy controller. For example, it can be a TS fuzzy model with a different number of partitions for certain state variables x_j (with a different number and shape of fuzzy sets corresponding to those variables), and thus with a different number and parameters of the rule antecedents. If the number of these rules is denoted by ro,

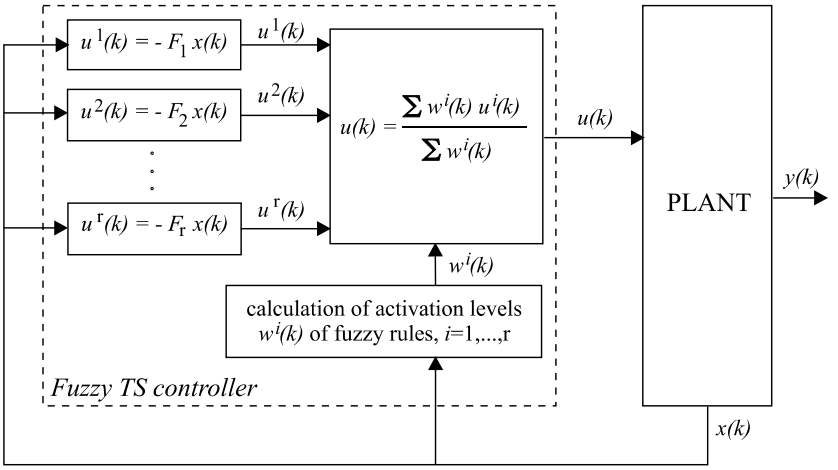

Fig. 2.21. Structure of a control system with the TS fuzzy state-feedback controller

and levels of their activation by $w_o^i(k)$, $i = 1, ..., ro$, then the model output will be given by the following formula, analogous to (2.9)

$$x(k+1) = \frac{\sum_{i=1}^{ro} w_o^i(k)[\mathbf{A}_i x(k) + \mathbf{B}_i u(k)]}{\sum_{l=1}^{ro} w_o^l(k)} = \sum_{i=1}^{ro} \tilde{w}_o^i(k)[\mathbf{A}_i x(k) + \mathbf{B}_i u(k)] \tag{2.23}$$

where $\tilde{w}_o^i(k)$ are normalized rule activation levels. Of course, if the TS fuzzy model used for process modeling in the feedback control system structure is different than the model (2.8) used for the design of the controller, then also its matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ are generally different (although we do not introduce different symbols here, as it does not lead to misunderstanding: when analyzing the closed-loop control system only matrices of the model (2.23) representing the process in the loop and controller matrices $\mathbf{F}_j$ occur in the control system description). Substituting the controller equations (2.22) into (2.23) we obtain the description of the closed-loop system in the following form

$$x(k+1) = \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(k) \tilde{w}^j(k)(\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(k) \tag{2.24}$$

To examine the stability properties of the dynamic system (2.24) Theorem 2.1 can be directly used, however, only matrices $\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j$, $i = 1, 2, ..., ro$, $j = 1, 2, ..., r$, should be taken in place of matrices $\mathbf{A}_i$, $i = 1, 2, ..., r$ in its formulation.

However, if during modeling the closed-loop control system presented in Fig. 2.21 one uses the same TS fuzzy model for the plant description in the loop that was used for the fuzzy controller design (i.e. with the same number

of rules and the same rule antecedents), then the formula describing dynamics of the closed-loop control system will have the following form

$$x(k+1) = \sum_{i=1}^{r} \sum_{j=1}^{r} \tilde{w}^i(k)\tilde{w}^j(k)(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j)x(k) \tag{2.25}$$

Then we can obtain a more convenient, simpler formulation of the stability conditions if we exploit the equality

$$\tilde{w}^i(k)\tilde{w}^j(k) = \tilde{w}^j(k)\tilde{w}^i(k), \quad i,j = 1,2,...,r$$

resulting from the identity of the rule antecedents in the process model and in the controller. Equation (2.25) can then be written in the form

$$x(k+1) = \sum_{i=1}^{r} \tilde{w}^i(k)\tilde{w}^i(k)(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i)x(k) + 2 \sum_{i,j=1,i<j}^{r} \tilde{w}^i(k)\tilde{w}^j(k)\mathbf{D}_{ij}x(k) \tag{2.26}$$

where

$$\mathbf{D}_{ij} = \frac{1}{2}[(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j) + (\mathbf{A}_j - \mathbf{B}_j\mathbf{F}_i)], \quad i<j,\ i,j = 1,2,...,r \tag{2.27}$$

while the symbol $\sum_{i,j=1,i<j}^{r}$ denotes addition of terms with all pairs of indexes i,j such that $i<j$, $i,j = 1,2,...,r$, *e.g.*,

$$\sum_{i,j=1,i<j}^{3} \mathbf{D}_{ij} = \mathbf{D}_{12} + \mathbf{D}_{13} + \mathbf{D}_{23}.$$

Thus, directly from Theorem 2.1 we obtain

Corollary 2.3 *Equilibrium point of the control system (2.25) described by the rules of the process model (2.8) and the controller (2.21) is globally asymptotically stable, if there exists a positive definite matrix* $\mathbf{P}$ *satisfying the following conditions*

$$(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i)^T\mathbf{P}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i) - \mathbf{P} < 0,\ i = 1,...,r \tag{2.28}$$

$$\mathbf{D}_{ij}^T\mathbf{P}\mathbf{D}_{ij} - \mathbf{P} < 0,\ i<j,\ i,j = 1,...,r \tag{2.29}$$

for all pairs (i,j) *except for those for which always (for every sampling instant* k*)* $w^i(k)w^j(k) = 0$. □

In [132] a certain weakening of the above result was achieved, the proof runs similarly as in Theorem 2.1, with the use of the same form of the Lyapunov function.

Theorem 2.4 *Let the maximum number of rules activated at the same time in the control system (2.25) described by the rules of the process model (2.8)*

and of the controller (2.21) be no higher than s, $1 < s \leq r$. *The equilibrium point of this system is globally asymptotically stable if there exist a positive definite matrix* $\mathbf{P}$ *and a positive semi-definite matrix* $\mathbf{Q}$ *satisfying the conditions*

$$(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i)^T\mathbf{P}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i) - \mathbf{P} + (s-1)\mathbf{Q} < 0, \quad i = 1, ..., r \tag{2.30}$$

$$\mathbf{D}_{ij}^T\mathbf{P}\mathbf{D}_{ij} - \mathbf{P} - \mathbf{Q} \leq 0 \quad i < j, \quad i, j = 1, ..., r \tag{2.31}$$

for all pairs (i, j) *except for those for which* $w^i(k)w^j(k) = 0$ *for every sampling instant* k. □

Let us note that conditions (2.30) and (2.31) reduce to (2.28) and (2.29), if $\mathbf{Q} = \mathbf{0}$ is assumed.

Formulations of stability conditions given above rely on a global quadratic Lyapunov function. In many cases, weaker conditions can be obtained when a piecewise-quadratic Lyapunov function is applied instead of the global one, as it was shown in Theorem 2.2. To apply this result to the state-feedback case, the space of the process states partitioned into constant activation cells S_l, $l = 1, ..., L$, must be used, as for the uncontrolled process before. Recalling that $K(l)$ denotes a set containing indexes of all fuzzy rules with non-zero activation level within S_l, the state-feedback controller takes the form (compare with (2.22))

$$u(k) = -\sum_{j \in K(l)} \tilde{w}^j(k)\mathbf{F}_j x(k), \quad x(k) \in S_l, \quad l \in L \tag{2.32}$$

The closed-loop system dynamics can then be described as follows

$$x(k+1) = \sum_{i \in K(l)} \tilde{w}^i(k)[\mathbf{A}_i - \mathbf{B}_i \sum_{j \in K(l)} \tilde{w}^j(k)\mathbf{F}_j]x(k), \quad x(k) \in S_l, \ l \in L \tag{2.33}$$

where $\sum_{j \in K(l)} \tilde{w}^j(k) = 1$. Due to this last equality, (2.33) can be written in the form

$$x(k+1) = \sum_{i \in K(l)} \sum_{j \in K(l)} \tilde{w}^i(k)\tilde{w}^j(k)[\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j]x(k), \quad x(k) \in S_l, \ l \in L \tag{2.34}$$

which is a partitioned form of the overall dynamics description (2.25), corresponding to the partitioning of the whole process domain into L constant activation cells S_l. Theorem 2.2 can now be directly applied to the considered state-feedback case yielding the following result.

Corollary 2.5 *Equilibrium point of the control system (2.34) described by the rules of the process model (2.8) and of the controller (2.21) is globally asymptotically stable, if there exists L symmetric and positive definite matrices* $\mathbf{P}_l$, $l \in L$ *such that the following set of inequalities is satisfied*

$$[\mathbf{A}_i^T - \mathbf{B}_i\mathbf{F}_j]\mathbf{P}_m[\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j] - \mathbf{P}_l < 0, \quad \text{for every } (l,m) \in \Omega, \quad i,j \in K(l) \tag{2.35}$$

where Ω denotes the set of index pairs indicating all possible one-step closed-loop system state transitions between the activation cells S_l, see (2.15). □

The weak point of the above formulation is the necessity to define, in advance, all possible one-step state transitions between all activation cells, *i.e.*, to define the set Ω. It should be emphasized that possible one-step state transitions for the closed-loop control system may be different than those for the uncontrolled process, and obviously the former must be taken into account in the definition of the set Ω in the above corollary. Certainly, underestimation of this set may be dangerous as omitted transitions may be those causing instability. Therefore, overestimation is very likely, leading to an increased, large number of inequalities in (2.35) – and the more inequalities the more constraining is the stability condition.

The design and analysis presented so far was for closed-loop TS fuzzy control systems with locally designed state-feedback matrices, as presented in the initial part of Section 2.2. This approach seems to be a convincing and natural generalization of the classical state-feedback design for a linear process model at a given equilibrium point. But another, more global oriented approach is also possible. It is based on a design of state-feedback matrices $\mathbf{F}_i$ ensuring stability of the overall state-feedback control system on the basis of a solution of a system of linear matrix inequalities, with respect to both state-feedback and stability related (Lyapunov type) matrices. For instance, with respect to all matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{F}_i$, $i = 1, ..., r$ given in Theorem 2.4 – the system of nonlinear inequalities (2.30)-(2.31) is then reformulated to the form of a system of linear matrix inequalities [132]. In [151] a similar approach is proposed for the design of a state-feedback H_∞ fuzzy controller, based on a piecewise-quadratic Lyapunov function. The sets of linear matrix inequalities can then be effectively solved using available packages, *e.g.*, *LMI Toolbox* of the MATLAB® package could be applied. We shall not present this approach here in detail, as in the author's opinion a more natural, more intuitive approach is the decentralized design methodology of a TS fuzzy controller, based on a number of local feedback designs. Moreover, in the case of an unsatisfactory result of the design, not only values of the feedback coefficients should be corrected, but first of all the number, positioning and shape of membership functions.

In practical applications, an entire vector of state variables may not be available for measurement. In this case the theory of state-feedback control suggests the use of a state observer. This approach can also be applied when designing a control system with a state-feedback TS fuzzy controller, constructing a state observer as a TS fuzzy system with the structure of rules identical to that of the rules of the process model and of the controller – but with rule consequents being classical formulae of linear state observers. Because controllers with feedback from observed state have not yet found po-

pularity in process industries, we shall omit the presentation of the design of fuzzy state observers and stability analysis of the closed-loop control systems with observers, asking the reader to refer to the literature [132].

Example 2.5
Let us reconsider a TS fuzzy model of a process presented in Example 2.4, but with a control variable u added. Fuzzy sets X_1 and X_2 defined by trapezoidal membership functions are presented in Fig. 2.19, while rules of the process model can be presented in the following form

$$\begin{aligned} \mathrm{R}_p^1: &\quad \text{IF} \quad x_2(k) \text{ is } X_1 \quad \text{THEN} \quad x^1(k+1) = \mathbf{A}_1 x(k) + \mathbf{B}u(k) \\ \mathrm{R}_p^2: &\quad \text{IF} \quad x_2(k) \text{ is } X_2 \quad \text{THEN} \quad x^2(k+1) = \mathbf{A}_2 x(k) + \mathbf{B}u(k) \end{aligned}$$

where $x(k) = [x_1(k)\ x_2(k)]^T \in \mathbb{R}^2$, $u(k) \in \mathbb{R}$,

$$\mathbf{A}_1 = \begin{bmatrix} 1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

It was shown in Example 2.4 that the autonomous (uncontrolled) TS fuzzy system can generate, for certain initial conditions, unstable state trajectories. We shall now design, for that system, a TS fuzzy controller with state-feedback ensuring stable operation of the closed-loop control system. The controller rules will have the following form

$$\begin{aligned} \mathrm{R}_c^1: &\quad \text{IF} \quad x_2(k) \text{ is } X_1 \quad \text{THEN} \quad u(k) = -\mathbf{F}_1 x(k) \\ \mathrm{R}_c^2: &\quad \text{IF} \quad x_2(k) \text{ is } X_2 \quad \text{THEN} \quad u(k) = -\mathbf{F}_2 x(k) \end{aligned}$$

It will be most simple to assume identical placement of eigenvalues of state matrices $\mathbf{A}_1 - \mathbf{B}\mathbf{F}_1$ and $\mathbf{A}_2 - \mathbf{B}\mathbf{F}_2$ of individual local feedback systems. Assuming the eigenvalues equal to 0.5 and 0.4 the elements of vectors $\mathbf{F}_1$ and $\mathbf{F}_2$ were calculated (using the *place* function from the *Control System Toolbox* of the MATLAB® package).

$$\mathbf{F}_1 = \begin{bmatrix} 0.1 & -0.3 \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} -1.9 & -0.3 \end{bmatrix}$$

The situation of identical rules in the fuzzy controller and in the process model used for the simulation of the control system will be considered. Thus, to examine the stability it is enough to calculate three matrices, which due to the assumptions should have identical values

$$\begin{aligned} \mathbf{A}_1 - \mathbf{B}\mathbf{F}_1 &= \mathbf{A}_2 - \mathbf{B}\mathbf{F}_2 = \\ &= \mathbf{D}_{12} = \frac{1}{2}[(\mathbf{A}_1 - \mathbf{B}\mathbf{F}_2) + (\mathbf{A}_2 - \mathbf{B}\mathbf{F}_1)] = \begin{bmatrix} 0.9 & -0.2 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

The closed-loop TS fuzzy control system will be, therefore, described by one equation

$$x(k+1) = \mathbf{D}_{12}x(k)$$

and is, of course, stable with assumed eigenvalues of the state matrices equal to 0.5 and 0.4.

For a more elaborate illustration of the stability properties of the closed-loop system we shall repeat the design of the controller assuming a slightly different vector of feedback coefficients in the second subprocess, namely $\mathbf{F}_2 = [-2.1 \ -0.2]$, which corresponds to eigenvalues of the matrix $\mathbf{A}_2 - \mathbf{BF}_2$ equal to 0.5 and 0.6. Now we have

$$\mathbf{A}_1 - \mathbf{BF}_1 = \begin{bmatrix} 0.9 & -0.2 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{A}_2 - \mathbf{BF}_2 = \begin{bmatrix} 1.1 & -0.3 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{D}_{12} = \frac{1}{2}[(\mathbf{A}_1 - \mathbf{BF}_2) + (\mathbf{A}_2 - \mathbf{BF}_1)] = \begin{bmatrix} 1 & -0.25 \\ 1 & 0 \end{bmatrix}$$

A sufficient condition of stability of the closed-loop system is, according to the Corollary 2.3, the existence of a symmetric, positive definite matrix $\mathbf{P}$ satisfying the system of inequalities

$$(\mathbf{A}_1 - \mathbf{BF}_1)^T\mathbf{P}(\mathbf{A}_1 - \mathbf{BF}_1) - \mathbf{P} < \mathbf{0}$$
$$(\mathbf{A}_2 - \mathbf{BF}_2)^T\mathbf{P}(\mathbf{A}_2 - \mathbf{BF}_2) - \mathbf{P} < \mathbf{0}$$
$$(\mathbf{D}_{12})^T\mathbf{P}\mathbf{D}_{12} - \mathbf{P} < \mathbf{0}$$

Using the *LMI Toolbox* of the MATLAB® package it was checked that such a matrix exists, obtaining

$$\mathbf{P} = \begin{bmatrix} 1.9755 & -0.7968 \\ -0.7968 & 0.9459 \end{bmatrix}$$

□

2.2.2 Discrete TS Fuzzy Output-feedback Controllers

In control systems of technological processes linear controllers with feedback from the output are usually used, particularly the well known PID type controllers. Along with a more and more wide application of multilayer control structures with on-line optimization of operating points, there grows a necessity of applications of nonlinear controllers, which are able to operate properly for a wide range of different set-points. Thus, in industrial applications it is important to have a relatively simple nonlinear controller, which would ideally be a nonlinear generalization of the PID controller. The construction of a discrete TS fuzzy output-feedback controller fulfilling the above postulates was proposed in [31], see also [33, 32], where it was called a *multi-regional controller*.

Let us consider a TS fuzzy model of a process which will be used for the design of a fuzzy controller. We shall assume the rules of this model in the following form

$$\begin{aligned} \mathrm{R}_p^i : \mathrm{IF} \quad & y(k) \text{ is } A_0^i \text{ and } y(k-1) \text{ is } A_1^i \text{ and } \cdots \text{ and } y(k-n_R) \text{ is } A_{n_R}^i \\ & \text{and } u(k-1) \text{ is } B_1^i \text{ and } \cdots \text{ and } u(k-m_R) \text{ is } B_{m_R}^i \\ \text{THEN} \quad & y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \cdots + a_{n_A}^i y(k-n_A+1) + \\ & + b_0^i u(k) + b_1^i u(k-1) + \cdots + b_{m_B}^i u(k-m_B) \end{aligned} \tag{2.36}$$

where $i = 1, \ldots, r$ indexes rules and related fuzzy sets (sub-domains) in the domain of the TS fuzzy model (generally multi-dimensional), $y(k)$ is a value of the process output at the k-th sampling instant, $u(k)$ is a value of the process control input at the k-th sampling instant, $A_j^i \in \mathbb{Y}_j$, $B_j^i \in \mathbb{U}_j$, whereas a_j^i and b_j^i are coefficients of functions in the rule consequents, $i = 1, ..., r$. Elements of each of the sets $\mathbb{Y}_j = \{Y_{j1}, ..., Y_{jr_{yj}}\}$ are fuzzy sets covering the domain of the variable $y(k-j)$, $j = 0, \ldots, n_R$, similarly $\mathbb{U}_j = \{U_{j1}, ..., U_{jr_{uj}}\}$ for $u(k-j)$, $j = 1, \ldots, m_R$.

The presented situation is the most general one; we usually have to deal with a far simpler case, when

$$\mathbb{Y}_0 = \cdots = \mathbb{Y}_{n_R} = \mathbb{Y}, \quad \mathbb{Y} = \{Y_1, ..., Y_{r_y}\}$$

i.e., partitions of the domain of each of the current and past output variables, *i.e.*, $y(k)$, $y(k-1), ..., y(k-n_R)$, are the same. Similarly, the partitions of the domain of the control input vector are usually the same, $\mathbb{U}_1 = \cdots = \mathbb{U}_{m_R} = \mathbb{U}$, $\mathbb{U} = \{U_1, ..., U_{r_u}\}$.

Dynamic linear models applied in the consequents of rules (2.36) are the ARX-type models. These models also apply to systems with time delays equal to several, say τ, sampling periods. Then, appropriate coefficients of the rule consequents will only be zero, $b_0^i = b_1^i = \cdots = b_\tau^i = 0$.

The output of a fuzzy model will be calculated according to the general formula (2.5), *i.e.*,

$$y(k+1) = \frac{\sum_{i=1}^{r} w^i(k)\, y^i(k+1)}{\sum_{l=1}^{r} w^l(k)} \tag{2.37}$$

where $w^i(k)$ are activation levels of individual rules (2.36) at sampling instant k,

$$w^i(k) = \prod_{j=0}^{n_R} \mu_{A_j^i}(y(k-j)) \prod_{p=1}^{m_R} \mu_{B_p^i}(u(k-p)) \tag{2.38}$$

For a model of the process described by the rules (2.36) the following *rules of a TS fuzzy controller* are formulated

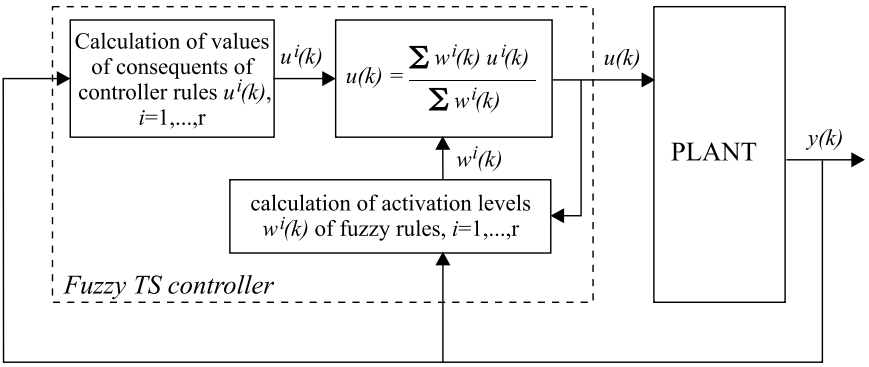

Fig. 2.22. Structure of a control system with the TS fuzzy output-feedback controller

$$\begin{aligned} \mathrm{R}_c^j:\ \text{IF}\ \ & y(k) \text{ is } A_0^j \ \text{ and } \ y(k-1) \text{ is } A_1^j \ \text{ and } \cdots \text{ and } \ y(k-n_R) \text{ is } A_{n_R}^j \\ & \text{and } \ u(k-1) \text{ is } B_1^j \ \text{ and } \cdots \text{ and } \ u(k-m_R) \text{ is } B_{m_R}^j \\ \text{THEN } \ & u^j(k) = c_1^j e(k) + c_2^j e(k-1) + \cdots + c_{n_C}^j e(k-n_C+1) + \\ & + d_1^j u(k-1) + d_2^j u(k-2) + \cdots + d_{m_D}^j u(k-m_D) \end{aligned} \quad (2.39)$$

where $j = 1, \ldots, r$ indexes the rules whose antecedents are identical to those in the model (2.36), $e(k) = y^{sp}(k) - y(k)$ is the control error at sampling instant k, while c_j^j and d_j^j are coefficients of functions in the rule consequents. The form of the discrete control algorithm occurring in the rule consequents (2.39) is fairly general, its special cases are algorithms of *discrete PID controllers* or unconstrained (explicit) versions of *predictive controllers of DMC and GPC type* (described in Chapter 3).

The controller output equation takes a standard form for TS fuzzy structures:

$$u(k) = \frac{\sum_{j=1}^{r} w^j(k)\, u^j(k)}{\sum_{l=1}^{r} w^l(k)} = \sum_{j=1}^{r} \tilde{w}^j(k)\, u^j(k) \quad (2.40)$$

where $w^j(k)$ are activation levels of individual rules (2.39) at k-th sampling instant, defined by formulae (2.38), due to the same rule antecedents as in (2.36). The structure of a control system with the TS fuzzy output-feedback controller described above is presented in Fig. 2.22.

Let us note that in the rule antecedents of the controller (2.39) the condition "$u(k)$ is B_0^j" does not appear, as this would lead to a logical inconsistency. Because a control value $u(k)$ for k-th sampling instant is to be calculated in functional consequents of the rules of the controller, then $u(k)$ cannot occur at k-th sampling instant in the rule antecedents. Thus, the simplest solution is to design the antecedents of the controller rules without the condition

"$u(k)$ is B_0^j", as it was done in (2.39). This is natural, especially for the widely applied discrete control with the extrapolator of a zero order. Since we measure and thus know the value of the process output $y(k)$ just before calculating the control value $u(k)$ at the k-th sampling instant (*i.e.*, for the sampling period beginning at that moment), but the process is still under the influence of the control $u(k-1)$ when $u(k)$ is being computed.

For simulative or analytical research of the closed-loop control system with the fuzzy controller described by (2.39), presented in Fig. 2.22, it is possible to use a process model different that given by the rules (2.36) and used for the construction of the controller. In particular, it can be a slightly different TS fuzzy model with the number of rules ro (generally $ro \neq r$) and the rules in the form

$$\begin{aligned} \mathrm{R}_{po}^i:\ \text{IF}\ \ & y(k) \text{ is } A_0^i \ \text{ and } \ y(k-1) \text{ is } A_1^i \ \text{ and } \cdots \text{ and } \ y(k-n_O) \text{ is } A_{n_O}^i \\ & \text{and } \ u(k) \text{ is } B_0^i \ \text{ and } \ u(k-1) \text{ is } B_1^i \ \text{ and } \cdots \text{ and } \ u(k-m_O) \text{ is } B_{m_O}^i \\ \text{THEN } \ & y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \cdots + a_{n_A}^i y(k-n_A+1) + \\ & + b_0^i u(k) + b_1^i u(k-1) + \cdots + b_{m_B}^i u(k-m_B) \end{aligned} \tag{2.41}$$

where $i = 1, ..., ro$, while the parameters n_A and m_B of the rules (2.36) and (2.41) can be of different values. We did not introduce different descriptions for those parameters, in order not to complicate the notation; it does not lead to misunderstandings as in the closed-loop control system the description of the process model (2.41) only is present. The activation levels of the rules (2.41) will be denoted by $w_o^i(k)$, and their normalized values by $\tilde{w}_o^i(k)$,

$$y(k+1) = \frac{\sum_{i=1}^{ro} w_o^i(k)\, y^i(k+1)}{\sum_{l=1}^{ro} w_o^l(k)} = \sum_{i=1}^{ro} \tilde{w}_o^i(k)\, y^i(k+1) \tag{2.42}$$

Let us note that in the rule antecedents of the process model (2.41) the condition "$u(k)$ is B_0^i" can be present, in general. It does not lead to logical inconsistency here, as in the consequents of these rules the process output value for the next ($(k+1)$-th) sampling instant is calculated.

However, if it is desired that in the rule antecedents of the controller the condition "$u(k)$ is B_0^j" should be present, then it is possible provided modified consequents in the controller rules are designed. Namely, in order to eliminate the logical inconsistency, it is necessary to apply in the rule consequents control algorithms calculating the control for the next sampling period (see [161, 32]), *i.e.*, to use the controller rules of the following form

$$\begin{aligned} \mathrm{R}_c^j:\ \text{IF}\ \ & y(k) \text{ is } A_0^j \ \text{ and } \ y(k-1) \text{ is } A_1^j \ \text{ and } \cdots \text{ and } \ y(k-n_R) \text{ is } A_{n_R}^j \\ & \text{and } \ u(k) \text{ is } B_0^j \ \text{ and } \ u(k-1) \text{ is } B_1^j \ \text{ and } \cdots \text{ and } \ u(k-m_R) \text{ is } B_{m_R}^j \\ \text{THEN } \ & u^j(k+1) = c_1^j e(k) + c_2^j e(k-1) + \cdots + c_{n_C}^j e(k-n_C+1) + \\ & + d_1^j u(k) + d_2^j u(k-1) + d_3^j u(k-2) + \cdots + d_{m_D}^j u(k-m_D+1) \end{aligned} \tag{2.43}$$

It should be added that conditions containing the process control input variables are often not present at all in the antecedents of the process model rules. Therefore, consideration of the problem of occurrence of the condition "$u(k)$ is B_0^i" in the rule antecedents is of no significance for these applications. The key variables of the rule antecedents of a TS fuzzy model are usually the process output variables $y(k), y(k-1), \ldots$. Therefore, for an important class of problems it is enough to use a process model (and thus a controller) in which conditions of the antecedents are defined for the output variables only, *i.e.*, to use the following rules

$$\begin{aligned} \mathrm{R}_p^i: \ \text{IF} \ \ & y(k) \text{ is } A_0^i \ \text{ and } \ y(k-1) \text{ is } A_1^i \ \text{ and } \cdots \text{ and } \ y(k-n_R) \text{ is } A_{n_R}^i \\ \text{THEN} \ \ & y^i(k+1) = a_1^i y(k) + a_2^i y(k-1) + \cdots + a_{n_A}^i y(k-n_A+1) + \\ & + b_0^i u(k) + b_1^i u(k-1) + \cdots + b_{m_B}^i u(k-m_B) \end{aligned} \tag{2.44}$$

Let us now consider the stability of a closed-loop control system with a TS fuzzy output-feedback controller and a process model in the form of a TS fuzzy system. Let us begin from the general case when the model used for describing the process in the feedback loop, although also a TS fuzzy one, is however different than the one used for the design of the controller. This generally results in a different number and different antecedents of process and controller fuzzy rules. Let us assume the process model with the rules (2.41) and the controller with the rules (2.39). To shorten the notation, in all the formulae presented next we shall denote levels of activation of the process and controller rules by w_o^i, w^j instead of $w_o^i(k)$, $w^j(k)$ (dropping the time variable k), and consequently, the normalized levels of activation, see (2.6), by $\tilde{w}_o^i$, $\tilde{w}^j$. For the simplicity and clarity of the following algebraic expressions we shall also assume that

$$\begin{aligned} n_A &= n_C = n \\ m_B &= m_D = m \end{aligned} \tag{2.45}$$

which should be understood as follows:

- $n = \max\{n_A, n_C\}$ and the missing coefficients a_p^i (if $n_A < n_C$) or c_p^i (if $n_A > n_C$) are zeros; similarly
- $m = \max\{m_B, m_D\}$ and the missing coefficients b_p^i (if $m_B < m_D$) or d_p^i (if $m_B > m_D$) are zeros.

Substituting dependencies describing local process models into (2.42) we have

$$y(k+1) = \sum_{i=1}^{ro} \tilde{w}_o^i \left\{ \sum_{p=1}^{n_A} a_p^i y(k-p+1) + \sum_{q=0}^{m_B} b_q^i u(k-q) \right\} \tag{2.46}$$

Analogously, inserting equations of local controllers (2.39) into (2.40) we obtain

$$u(k) = \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j e(k-t+1) + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\}$$
$$= \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j y^{sp}(k-t+1) - \sum_{t=1}^{n_C} c_t^j y(k-t+1) + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\} \tag{2.47}$$

Now inserting this dependence into (2.46) and taking into account that $\sum_{j=1}^{r} \tilde{w}^j = 1$, we obtain the equation describing the closed-loop control system

$$y(k+1) = \sum_{i=1}^{ro} \tilde{w}_o^i \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{p=1}^{n_A} a_p^i y(k-p+1) + \sum_{q=1}^{m_B} b_q^i u(k-q) \right\} +$$
$$+ \sum_{i=1}^{ro} \tilde{w}_o^i b_0^i \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{t=1}^{n_C} c_t^j y^{sp}(k-t+1) - \sum_{t=1}^{n_C} c_t^j y(k-t+1) + \right.$$
$$\left. + \sum_{s=1}^{m_D} d_s^j u(k-s) \right\}$$

Applying now the assumption (2.45) which allows for a consistent and clear notation we can, after proper grouping of the terms, present the obtained equation in the following form

$$y(k+1) = \sum_{i=1}^{r} \tilde{w}_o^i \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{p=1}^{n} (a_p^i - b_0^i c_p^j) y(k-p+1) + \right.$$
$$\left. + \sum_{q=1}^{m} (b_q^i + b_0^i d_q^j) u(k-q) + \sum_{t=1}^{n} b_0^i c_t^j y^{sp}(k-t+1) \right\} \tag{2.48}$$

Let us now define a state vector $x(k) \in \mathbb{R}^{n+m-1}$ describing the dynamics given by (2.48) and (2.47) as follows

$$x(k) = [y(k)\ y(k-1)\ \cdots\ y(k-n+1)\ u(k-1)\ u(k-2)\ \cdots\ u(k-m)]^T \tag{2.49}$$

Furthermore, in order to have the following expressions concise, let us define the parameters

$$ac_p^{i,j} = a_p^i - b_0^i c_p^j, \quad p = 1, ..., n$$
$$bd_q^{i,j} = b_q^i + b_0^i d_q^j, \quad q = 1, ..., m$$

Then, taking into account that (2.47) is equivalent to the equation

$$u(k) = \sum_{i=1}^{ro} \tilde{w}_o^i \sum_{j=1}^{r} \tilde{w}^j \left\{ \sum_{t=1}^{n} c_t^j [y^{sp}(k-t+1) - y(k-t+1)] + \right.$$
$$\left. + \sum_{s=1}^{m} d_s^j u(k-s) \right\}$$

the description of the closed-loop control system can be presented in the form

$$x\left(k+1\right)=\sum_{i=1}^{ro}\sum_{j=1}^{r}\tilde{w}_o^i\tilde{w}^j\mathbf{A}_{ij}x(k)+\sum_{i=1}^{ro}\sum_{j=1}^{r}\tilde{w}_o^i\tilde{w}^j\sum_{t=1}^{n}\mathbf{E}_{ij}\tilde{y}^{sp}(k) \tag{2.50}$$

where

$$\mathbf{A}_{ij}=\begin{bmatrix} ac_1^{i,j} & ac_2^{i,j} & \cdots & ac_{n-1}^{i,j} & ac_n^{i,j} & bd_1^{i,j} & bd_2^{i,j} & \cdots & bd_{m-1}^{i,j} & bd_m^{i,j} \\ 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -c_1^j & -c_2^j & \cdots & -c_{n-1}^j & -c_n^j & d_1^j & d_2^j & \cdots & d_{m-1}^j & d_m^j \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{2.51}$$

$$\mathbf{E}_{ij}=\begin{bmatrix} b_0^ic_1^j & b_0^ic_2^j & \cdots & b_0^ic_{n-1}^j & b_0^ic_n^j \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ c_1^j & c_2^j & \cdots & c_{n-1}^j & c_n^j \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{2.52}$$

and $\tilde{y}^{sp}(k)$ is defined as

$$\tilde{y}^{sp}(k)=[y^{sp}(k)\;\; y^{sp}(k-1)\;\; \cdots\;\; y^{sp}(k-n+1)]^T \tag{2.53}$$

Dynamics of the autonomous system is described by the equation

$$x\left(k+1\right)=\sum_{i=1}^{ro}\sum_{j=1}^{r}\tilde{w}_o^i(k)\tilde{w}^j(k)\mathbf{A}_{ij}x(k) \tag{2.54}$$

Applying now to (2.54) the Lyapunov theorem we can obtain a result analogous to Theorem 2.1, which can also be treated as a direct application of this theorem, provided all matrices $\mathbf{A}_{ij}$ with indexes for which $w_o^i(k)w^j(k)=0$ are always excluded:

Corollary 2.6 *The equilibrium point of a TS fuzzy system described by (2.54) is globally asymptotically stable, if there exists a symmetric positive definite matrix* $\mathbf{P}$ *such that for each matrix* $\mathbf{A}_{ij}$ *the following equation is fulfilled*

$$\mathbf{A}_{ij}^T \mathbf{P} \mathbf{A}_{ij} - \mathbf{P} < \mathbf{0} \tag{2.55}$$

for all pairs (i,j), $i = 1,2,...,ro$, $j = 1,2,...,r$ *except for those for which always* $w_o^i(k)w^j(k) = 0$. □

If we assume that the number of rules and the rule antecedents in a model used for the description of the process in a closed-loop control system and in the controller description are identical, then the following equalities are true: $ro = r$, $\tilde{w}_o^i(k) = \tilde{w}^i(k)$, $i = 1,...,r$ and $\tilde{w}^i(k)\tilde{w}^j(k) = \tilde{w}^j(k)\tilde{w}^i(k)$, for every value of $i,j = 1,2,...,r$. Using this last equality it is possible to diminish the number of matrices $\mathbf{A}_{ij}$ occurring in (2.55), proceeding analogously as in the previous section. Namely, the dynamics (2.54) can be then presented in the form

$$x(k+1) = \left[\sum_{i=1}^{r} \tilde{w}^i(k)\tilde{w}^i(k)\mathbf{A}_{ii} + 2 \sum_{i,j=1,\, i<j}^{r} \tilde{w}^i(k)\tilde{w}^j(k)\overline{\mathbf{A}_{ij}} \right] x(k) \tag{2.56}$$

where

$$\overline{\mathbf{A}_{ij}} = \frac{\mathbf{A}_{ij} + \mathbf{A}_{ji}}{2}, \quad i < j,\ i,j = 1,2,...,r$$

while $\sum_{i,j=1,\, i<j}^{r}$ denotes addition of all terms with pairs of indexes i,j such that $i < j$. In stability conditions (2.55) it is now enough to consider, instead of matrices $\mathbf{A}_{ij}$, $i,j = 1,2,...,r$, only the matrices $\mathbf{A}_{ii}$, $i = 1,...,r$ and $\overline{\mathbf{A}_{ij}}$, $i < j$, $i,j = 1,2,...,r$. Conditions (2.55) can in this case be made slightly weaker, by application of Theorem 2.4 to the dynamic system (2.56).

Corollary 2.7 *Let the maximum number of rules activated at the same sampling instant be no higher than* s, $1 < s \leq r$. *Then the equilibrium point of the dynamic system described by (2.56) is globally asymptotically stable, if there exist a positive definite matrix* $\mathbf{P}$ *and a positive semi-definite matrix* $\mathbf{Q}$, *satisfying the following conditions*

$$\begin{aligned} {\mathbf{A}_{ii}}^T \mathbf{P} \mathbf{A}_{ii} - \mathbf{P} + (s-1)\mathbf{Q} < 0, \quad i = 1,2,...,r \\ \overline{\mathbf{A}_{ij}}^T \mathbf{P} \overline{\mathbf{A}_{ij}} - \mathbf{P} - \mathbf{Q} \leq 0, \quad i < j,\ i,j = 1,2,...,r \end{aligned}$$

for all pairs (i,j) *except those for which* $w^i(k)w^j(k) = 0$ *for every value of* k. □

The performed stability analysis does not directly include the case of a control system with a controller defined by rules (2.43), whose antecedents additionally contain the condition "$u(k)$ is B_0^j", and thus consequents are defined by slightly different formulae – compare (2.39) with (2.43). A stability

analysis of such a control system can be conducted in a way identical as was done above, obtaining analogous results – only matrices $\mathbf{A}_{ij}$ and $\mathbf{E}_{ij}$ will be defined by slightly different formulae. These formulae can be obtained directly from those already mentioned, only a modification of the coefficients of the controller rule consequents is necessary, from the form

$$u^j(k+1) = c_1^j e(k) + c_2^j e(k-1) + \cdots + c_{n_C}^j e(k-n_C+1) + \\ + d_1^j u(k) + d_2^j u(k-1) + \cdots + d_{m_D}^j u(k-m_D+1)$$

to the form

$$u^j(k) = \bar{c}_1^j e(k) + \bar{c}_2^j e(k-1) + \cdots + \bar{c}_{\bar{n}_C}^j e(k-\bar{n}_C+1) + \\ + d_1^j u(k-1) + d_2^j u(k-3) + \cdots + d_{m_D}^j u(k-m_D)$$

where $\bar{n}_C = n_C + 1$, $\bar{c}_1^j = 0$, $\bar{c}_p^j = c_{p-1}^i$, $p = 2, ..., n_C + 1$. Moreover, $w^j(k-1)$ should be, consequently, inserted to all formulae in place of $w^j(k)$, $j = 1, ..., r$.

Example 2.6
Let us consider a process model described by the following fuzzy rules in the form (2.44)

$$\mathrm{R}_p^1: \quad \text{IF} \quad y(k) \text{ is } Y_1 \quad \text{THEN} \quad y^1(k+1) = 0.7y(k) + 0.8u(k) \tag{2.57}$$

$$\mathrm{R}_p^2: \quad \text{IF} \quad y(k) \text{ is } Y_2 \quad \text{THEN} \quad y^2(k+1) = 0.3y(k) + 0.2u(k) \tag{2.58}$$

Fig. 2.23 (a) presents fuzzy sets Y_1 and Y_2, while Fig. 2.23 (b) presents step responses of local processes as described by consequents of the rules (2.57) and (2.58). Comparison of these responses shows that the considered process, despite its simple structure, is strongly nonlinear – nonlinear is in its statics

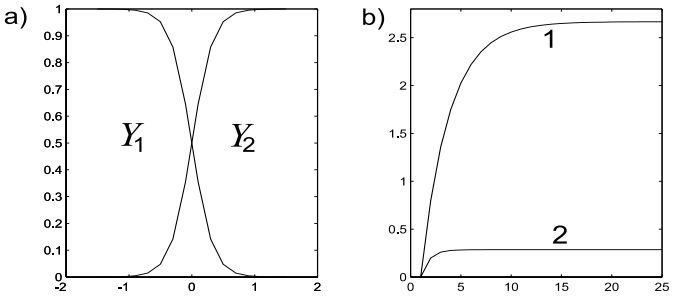

Fig. 2.23. a) sigmoidal membership functions, b) step responses, of local processes in Example 2.6

(gain) as well as in dynamics (time constant). On the other hand, it is not easily controlled, due to its time constant being relatively small when compared to the sampling period.

Discrete PI controllers were designed for process models from the rule consequents (2.57) and (2.58), in a version with integration implemented by the method of trapezoids, *i.e.*, in the following form

$$u(k) = u(k-1) + k_p(1 + \frac{T_p}{2T_I})e(k) + k_p(-1 + \frac{T_p}{2T_I})e(k-1)$$

where k_p is the controller gain and T_I is its integral time. Values of these parameters were selected by Ziegler-Nichols rules (see *e.g.*, [3]), followed by slight corrections, resulting in

$$k_p^1 = 0.9, \;\; T_I^1 = 3$$
$$k_p^2 = 2.0, \;\; T_I^2 = 1$$

Individual PI controllers operate very well in vicinities of their operating points, but poorly within the whole domain of the output variable. Figure 2.24 presents trajectories in the control system with a nonlinear process and, subsequently, with the first PI controller designed for the operating point $y = -1$ and with the second PI controller designed for the operating point $y = +1$. The first controller is definitely too slow in the range of positive values of y.

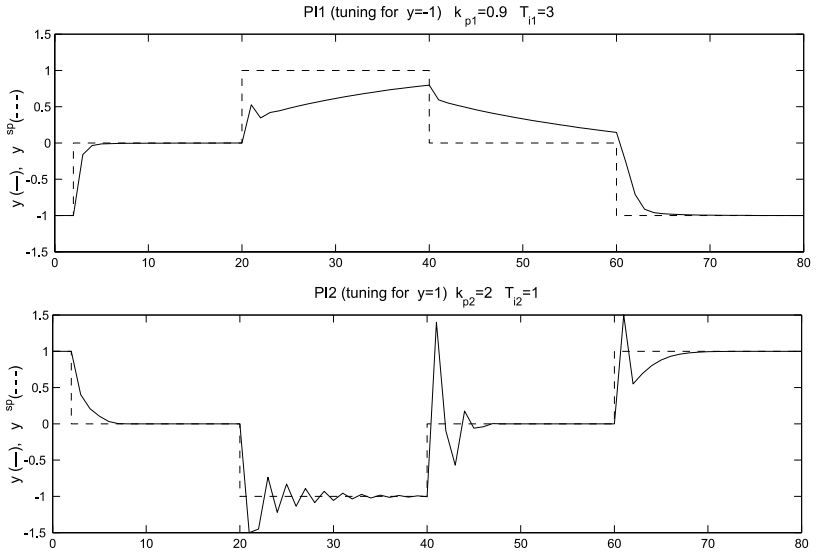

Fig. 2.24. Controlled variable trajectories in control systems with a nonlinear fuzzy process and the PI controller designed for the operating point $y = -1$ (upper figure) and for the operating point $y = +1$ (lower figure)

The second, in turn, is too aggressive, its trajectories are too oscillating not only for negative values of the controlled variable y, but also in the vicinity of its zero value. Moreover, Fig. 2.24 presents a trajectory which is too aggressive even in the vicinity of its own operating point $y = +1$, if a step change of the set-point from zero is followed. Therefore, before going on to the construction of a TS fuzzy controller the gain of the second controller was decreased to $k_p^2 = 1.5$.

Assuming antecedents such as in the process model and consequents in the form of the discussed local PI controllers we obtain the following rules of a TS fuzzy controller

$$\mathrm{R}_c^1\text{: IF } \; y(k) \text{ is } Y_1 \; \text{ THEN } \; u^1(k) = u(k-1) + 1.05e(k) - 0.75e(k-1) \tag{2.59}$$

$$\mathrm{R}_c^2\text{: IF } \; y(k) \text{ is } Y_2 \; \text{ THEN } \; u^2(k) = u(k-1) + 2.25e(k) - 0.75e(k-1) \tag{2.60}$$

Figure 2.25 presents trajectories of the process output and the control input in the closed-loop system with the designed TS fuzzy controller, for a fairly wide range of changes of the set-point, while Fig. 2.26 presents trajectories of the output variable in a slightly narrower, but critical range of changes of the set-point between -1 and 1. In the lower part of Fig. 2.26 a trajectory for the case $k_p^2 = 2$ is presented, namely without the previously mentioned damping of the gain k_p^2. As could be expected, increasing gain coefficient k_p^2 slightly

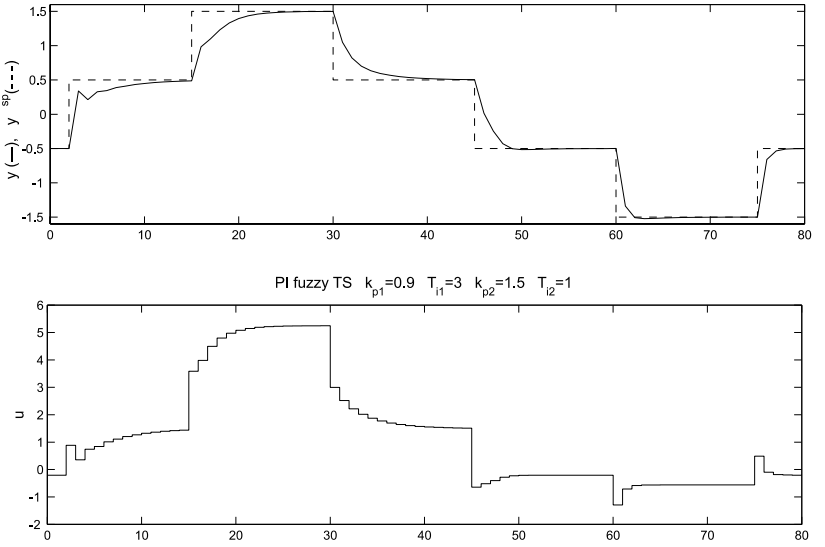

Fig. 2.25. Trajectories of the process output and input variables in the control system with the TS fuzzy controller

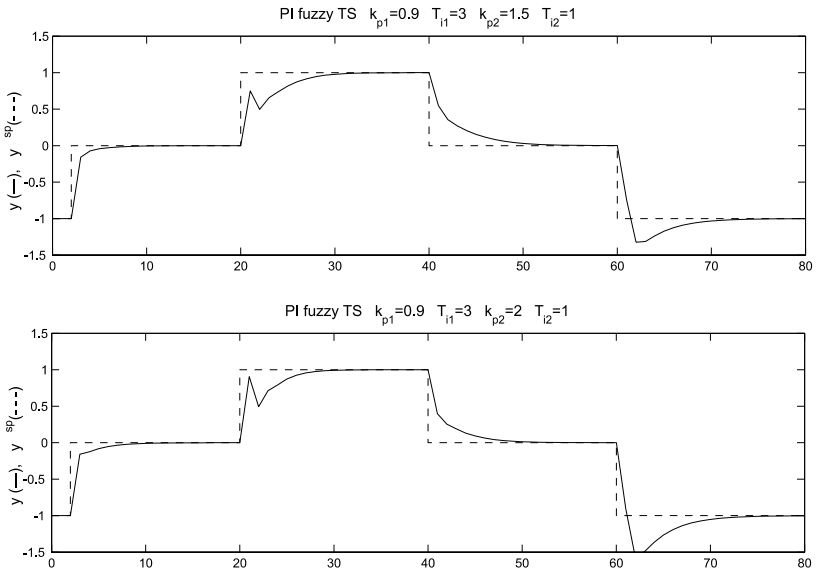

Fig. 2.26. Trajectories of the output variable in the control system with the PI fuzzy controller: with $k_p^2 = 1.5$ (upper trajectory), $k_p^2 = 2$ (lower trajectory)

speeds up the operation of a control system in the range of positive output values. Unfortunately, at the same time the overshoot increases with changes of the set-point in the direction of negative values, which can clearly be seen for a set-point step from zero to -1. Therefore, it justifies the damping of k_p^2 carried out previously. In Chapter 3 we shall present that the described conflict can be significantly reduced by employing a nonlinear predictive controller, with faster operation for positive output values and a simultaneous reduction of the overshoot for the negative values.

To perform stability analysis of the obtained nonlinear fuzzy control system (with $k_p^2 = 1.5$) we shall use Corollary 2.7, resulting from Theorem 2.4. In order to do this, matrices $\mathbf{A}_{ij}$ (2.51) must be evaluated. The extended state vector in the considered example is: $x(k) = [y(k)\ y(k-1)\ u(k-1)]^T$, where dimensions $n = 2$ and $m = 1$ result from the form of the process and controller rule consequents. Thus, the matrices A_{ij} are of dimension 3, and of the following general form

$$\mathbf{A}_{ij} = \begin{bmatrix} a_1^i - b_0^i c_1^j & a_2^i - b_0^i c_2^j & b_1^i + b_0^i d_1^j \\ 1 & 0 & 0 \\ -c_1^j & -c_2^j & d_1^j \end{bmatrix}, \quad i, j = 1, 2$$

Assuming, for a theoretical verification of stability conditions, the process modeled by (2.57), (2.58), *i.e.*, the same rule antecedents in the process and in the controller, it is enough to take into account matrices $\mathbf{A}_{11}$, $\mathbf{A}_{22}$ and $\overline{\mathbf{A}_{12}}$.

They have the following form

$$\mathbf{A}_{11} = \begin{bmatrix} -0.14 & 0.6 & 0.8 \\ 1 & 0 & 0 \\ -1.05 & 0.75 & 1 \end{bmatrix}$$

$$\mathbf{A}_{22} = \begin{bmatrix} -0.15 & 0.15 & 0.2 \\ 1 & 0 & 0 \\ -2.25 & 0.75 & 1 \end{bmatrix}$$

$$\overline{\mathbf{A}_{12}} = \begin{bmatrix} -0.5050 & 0.3750 & 0.5 \\ 1 & 0 & 0 \\ -1.65 & 0.75 & 1 \end{bmatrix}$$

Using the *LMI Toolbox* of the MATLAB® package it was checked that a system of linear matrix inequalities

$$\begin{aligned} (\mathbf{A}_{11})^T \mathbf{P} \mathbf{A}_{11} - \mathbf{P} &< 0 \\ (\mathbf{A}_{22})^T \mathbf{P} \mathbf{A}_{22} - \mathbf{P} &< 0 \\ (\overline{\mathbf{A}_{12}})^T \mathbf{P} \overline{\mathbf{A}_{12}} - \mathbf{P} &< 0 \\ -\mathbf{P} &< 0 \end{aligned}$$

has a solution

$$\mathbf{P} = 10^3 \begin{bmatrix} 5.4140 & -2.8931 & -3.0451 \\ -2.8931 & 2.2731 & 2.1858 \\ -3.0451 & 2.1858 & 2.5867 \end{bmatrix}$$

Thus, it follows from Corollary 2.7 (with $\mathbf{Q} = \mathbf{0}$) that sufficient conditions of stability of the considered nonlinear control system are satisfied. The conducted analysis applies to stability of a nominal system, there only remains to examine the robustness of the designed control system. It was checked that the system operates correctly even under quite significant deviations of its parameters. □

As mentioned previously, the form of the discrete TS fuzzy controller considered in this chapter also covers cases of predictive controllers of DMC and GPC type (in explicit, unconstrained versions). In Chapter 3 a design of nonlinear predictive controllers for the process from the example just shown above will be presented, including a GPC type controller.

2.3 Continuous-time TS Fuzzy Control

Until now, a discrete-time description of a process, and consequently, of controllers were assumed in our considerations. In practice, a discrete-time description is applied mainly to a situation when the sampling period in a control

system is not radically shorter than the dominant time constants of the process. For very small sampling periods we obtain, e.g.,, discrete step responses with a very large number of significant elements or difference equations of a high order, which unnecessarily makes the process of the design of control systems more difficult. On the other hand, as a result of the development of the microprocessor technology, controllers with a large computational power are widely available, which enables the use of small sampling periods even with more complex control algorithms. Thus, it is more common to use small sampling periods in direct control loops. Especially, because this approach enables a direct transfer from analog to digital technology – without the need to change the settings of the controllers.

In a situation when the sampling period in a control loop is small enough in relation to process dynamics, a fully authorized approach is to design a continuous-time control system, and then a digital realization of the obtained continuous-time controller by one of the emulation methods, see *e.g.*, [44, 52]. For that reason, in this section we shall be concerned with questions of the design and analysis of continuous (*i.e.*, continuous-time) TS fuzzy controllers. First of all, local control algorithms of such a controller will be defined as continuous. We should also mention that a verification of a control system by simulations can be conducted by formulating its description in a continuous time domain (*e.g.*, using the Simulink® program), or better yet, in a hybrid environment with a continuous realization of the process modeled by a set of differential equations integrated using an appropriate small step size and discrete-time controller realization.

Structures of continuous-time TS fuzzy algorithms are analogous to those for the discrete-time presented in the previous section. Therefore, considering design methods for continuous controllers and stability conditions of the obtained control systems we shall do it briefly, concentrating mainly on formulations and features distinguishing the two cases. Particularly, the first part of Section 2.2 was written in a general way, and thus refers to process modeling and to the design of TS fuzzy control algorithms, both in discrete and continuous versions.

2.3.1 Continuous TS Fuzzy State-feedback Controllers

Let us consider a dynamic process with the following state and control vectors

$$x(t) = [x_1(t)\ x_2(t)\ \cdots\ x_{n_x}(t)]^T$$

$$u(t) = [u_1(t)\ u_2(t)\ \cdots\ u_{n_u}(t)]^T$$

described by a TS fuzzy model with rules in the following form

$$\mathrm{R}_p^i: \quad \text{IF} \quad x_1(t) \text{ is } A_1^i \text{ and } x_2(t) \text{ is } A_2^i \text{ and } \cdots \text{ and } x_{n_x}(t) \text{ is } A_{n_x}^i$$
$$\text{THEN} \quad \dot{x}^i(t) = \mathbf{A}_i x(t) + \mathbf{B}_i u(t) \qquad (2.61)$$

where $A_j^i \in \mathbb{X}_j = \{X_{j1}, ..., X_{jr_j}\}$ are fuzzy sets of components x_j of the state vector x (see (2.4)), $j = 1, \ldots, n_x$, while $\mathbf{A}_i$ and $\mathbf{B}_i$ are state and control matrices of local linear models designed for individual fuzzy sub-domains of the model domain, $i = 1, ..., r$. When using a grid partition of the domain of the state variables we obtain a fuzzy system with a number of sub-domains (and rules) equal to $r = r_1 r_2 \cdots r_{n_x} = \prod_{j=1}^{n_x} r_j$.

For given values $x(t)$ and $u(t)$ the output of the fuzzy process model is calculated according to the standard formula, *i.e.*,

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} w^i(t)[\mathbf{A}_i x(t) + \mathbf{B}_i u(t)]}{\sum_{l=1}^{r} w^l(t)} \tag{2.62}$$

where $w^i(t)$ are activation levels of individual rules (2.61),

$$w^i(t) = \prod_{j=1}^{n_x} \mu_{A_j^i}(x_j(t)).$$

We have taken a natural assumption that always $\sum_{i=1}^{r} w^i(k) > 0$, *i.e.*, for each value of the state variables from their domain the model is well defined – at least one rule is activated.

The model of an autonomous process (with $u(t) \equiv 0$) can be written in the form

$$\dot{x}(t) = \sum_{i=1}^{r} \tilde{w}^i(t) \mathbf{A}_i x(t) \tag{2.63}$$

where $\tilde{w}^i(t)$ denote normalized activation levels of the rules. For such a dynamic model, the following sufficient condition of asymptotic stability follows directly from the Lyapunov theorem.

Theorem 2.8 *The equilibrium point of the dynamic system (2.63) is globally asymptotically stable if there exists a symmetric positive definite matrix* $\mathbf{P}$ *such that for the state matrix* $\mathbf{A}_i$ *of each local model the following inequality is satisfied*

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i < \mathbf{0}, \quad i = 1, 2, ..., r \tag{2.64}$$

Proof. It can be conducted similarly as that of Theorem 2.1. We define a scalar function

$$V(x(t)) = x^T(t) \mathbf{P} x(t) \tag{2.65}$$

where $\mathbf{P}$ is a symmetric positive definite matrix. The function (2.65) from definition satisfies all prerequisites to be a Lyapunov function except for negativity of its derivative along trajectories of (2.63). The latter will be proven now. Namely,

$$\dot{V}(x(t)) = \dot{x}^T(t) \mathbf{P} x(t) + x^T(t) \mathbf{P} \dot{x}(t)$$

$$= x^T(t) \left[\sum_{i=1}^{r} \tilde{w}^i(t) \mathbf{A}_i^T \mathbf{P} + \sum_{i=1}^{r} \tilde{w}^i(t) \mathbf{P} \mathbf{A}_i \right] x(t)$$

$$= \sum_{i=1}^{r} \tilde{w}^i(t) x^T(t) \left[\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i \right] x(t)$$

Because

$$\tilde{w}^i(t) \geq 0 \quad \text{for every } i = 1, 2, ..., r$$

$$\sum_{i=1}^{r} \tilde{w}^i(t) > 0$$

then $\dot{V}(x(t)) < 0$ for $x(t) \neq 0$, if only (2.64) is satisfied. Thus, the function $V(x(t))$ is a Lyapunov function for the nonlinear dynamic system (2.63). □

It was shown in [141] that *a necessary condition for existence of a matrix* $\mathbf{P}$ *satisfying (2.64) is that each of the sums* $\mathbf{A}_i + \mathbf{A}_j$, $i, j = 1, 2, ..., r$ *(actually, each sum of matrices from among* $\mathbf{A}_i$, $i = 1, 2, ..., r$*) is a matrix of an asymptotically stable system* (all its eigenvalues have negative real parts). This results directly from addition of inequalities (2.64) for $i = i$ and $i = j$. Namely, if $\mathbf{P}$ is positive definite and $(\mathbf{A}_i + \mathbf{A}_j)^T \mathbf{P} + \mathbf{P}(\mathbf{A}_i + \mathbf{A}_j) < \mathbf{0}$, then $\mathbf{A}_i + \mathbf{A}_j$ is an asymptotically stable matrix. Thus, finding that any of the sums $\mathbf{A}_i + \mathbf{A}_j$ is not such a matrix indicates non-existence of the matrix $\mathbf{P}$ satisfying conditions of Theorem 2.8.

The existence of one common positive definite matrix $\mathbf{P}$ satisfying inequalities (2.64) is a sufficient condition of stability. However, necessary and sufficient conditions are as yet unknown. Conditions (2.64) are of global nature, they do not take into account fuzzy structure of the system (location and shapes of membership functions). That is why attempts were undertaken to derive weaker conditions by the use of a more structured Lyapunov function. The most natural construction would be the following

$$V(x(t)) = x^T(t) \mathbf{P}(x(t)) x(t) = x^T(t) [\sum_{i=1}^{r} \tilde{w}^i(t) \mathbf{P}_i] x(t) \tag{2.66}$$

where $\mathbf{P}_i$ are symmetric positive definite matrices. This function has nice properties, it is continuous, differentiable and positive definite. Unfortunately, to the best knowledge of the author the attempts to derive clear and reasonable conditions assuring its derivative is negative along the trajectory of the system (2.63) have not been successful. The reason is the dependence of $\mathbf{P}(x(t))$ on $x(t)$, which results in an additional term (stemming from the time derivative of $\mathbf{P}(x(t))$) in the expression for $\dot{V}(x(t))$, which makes it difficult to obtain constructive conditions assuring this derivative is negative along the system trajectories, see [23, 21, 10]. So far, the obtained conditions are either not constructive enough (a priori assumption that a norm of the derivative of

the state vector is appropriately bounded [21])), or are formulated in a too complex form, of unclear usefulness at actual stage of knowledge [10]. Therefore, they shall not be quoted here, referring an interested reader to the cited papers.

A theoretically more successful, although leading to a constrained and significantly more complex construction of the Lyapunov function, occurred to be the approach using a piecewise-quadratic Lyapunov function of the form

$$V(x(t)) = x^T(t)\mathbf{P}_l x(t), \quad x(t) \in S_l, \quad l = 1, ..., L \tag{2.67}$$

where S_l, $l = 1, ..., L$ constitute a partition of the process state-space into non-overlapping subsets of constant activation of all fuzzy rules (each rule is active or not within a subset) – called *constant activation cells* and constructed precisely in the same way as defined and explained in Section 2.2.1 for a discrete-time case.

However, the difficulty in a design of the function (2.67) as a Lyapunov function is the requirement of continuity when crossing boundaries of neighboring cells – the matrices $\mathbf{P}_l$ should be constructed in a way assuring this continuity. In [60] a design of such a continuous function was proposed, based on structuring matrices $\mathbf{P}_l$ as follows

$$\mathbf{P}_l = \mathbf{F}_l^T \mathbf{T} \mathbf{F}_l, \quad l = 1, ..., L \tag{2.68}$$

where $\mathbf{T}$ is a symmetric matrix and matrices $\mathbf{F}_l$ satisfy the following conditions

$$\mathbf{F}_l x = \mathbf{F}_m x \quad \text{for} \quad x \in S_l \cap S_m, \quad l, m \in L \tag{2.69}$$

In [60], see also [116], a way of construction of the matrices $\mathbf{F}_l$ for the case of trapezoidal membership functions was given, where the matrices are fully determined by the shape (by the corner points) of the membership functions. The stability of the autonomous fuzzy system

$$\dot{x}(t) = \sum_{i \in K(l)} \tilde{w}^i(t)[\mathbf{A}_i x(t)], \quad x(t) \in S(l), \quad l \in L \tag{2.70}$$

can then be proven provided a matrix $\mathbf{T}$ can be found such that matrices (2.68) are positive definite and the condition

$$\mathbf{A}_i^T \mathbf{P}_l + \mathbf{P}_l \mathbf{A}_i < \mathbf{0}, \quad i \in K(l), \quad l = 1, ..., L \tag{2.71}$$

holds, where $K(l)$ denotes a set containing indexes of all fuzzy rules with nonzero activation levels within S_l, $l = 1, ..., L$. For a detailed design and analysis the reader is referred to [60], as the design is difficult, leading to large sets of linear matrix inequalities (in fact, condition (2.71) in an augmented, more complex but slightly weakened form was there formulated). Moreover, a practical importance of the discussed stability conditions and their relation to the global result given by Theorem 2.8 seems to be not sufficiently understood yet (note that a global matrix T is to be designed as well).

In [22] a completely different formulation of sufficient stability conditions for a dynamic system (2.63) was given.

Theorem 2.9 *The equilibrium point of a fuzzy dynamic system (2.63) is globally asymptotically stable if*

$$\lambda_{\max}(\mathbf{A}_i + \mathbf{A}_i^T) < 0, \quad i = 1, 2, ..., r \tag{2.72}$$

where $\lambda_{\max}(\mathbf{A}_i + \mathbf{A}_i^T)$ *denotes the maximal eigenvalue of a symmetric matrix* $\mathbf{A}_i + \mathbf{A}_i^T$. □

The above theorem is quoted here without proof (see. [22]), as the given stability condition (2.72), although of a quite different nature than the one from Theorem 2.8 and formulated in terms of local matrices, is difficult to interpret and generally turns out to be very conservative, thus of not much use.

For each local linear dynamic model from the rule consequents (2.61), *i.e.*, for each sub-domain of a TS fuzzy model, a linear state-feedback controller can be designed, using a classical method. This way the rules of a TS fuzzy controller are obtained, in the form

$$\begin{aligned} \mathrm{R}_c^j: \quad \text{IF} \quad & x_1(t) \text{ is } A_1^j \text{ and } x_2(t) \text{ is } A_2^j \text{ and } \cdots \text{ and } x_{n_x}(t) \text{ is } A_{n_x}^j \\ & \text{THEN} \qquad u^j(t) = -\mathbf{F}_j x(t) \end{aligned} \tag{2.73}$$

where $\mathbf{F}_i$ are matrices of state-feedback coefficients, $j = 1, 2, ..., r$. A nonlinear TS fuzzy controller is described by

$$u(t) = -\sum_{j=1}^{r} \tilde{w}^j(t) \mathbf{F}_j x(t) \tag{2.74}$$

where normalized levels of rule activation are the same as in the process model (2.61) used for the design of the controller.

To represent a process in the closed-loop control system, it is generally possible to use a TS fuzzy model different than the one used for the design of the fuzzy controller, *i.e.*, with a different number and/or shape of the rules (analogously as it was discussed in Section 2.2.1). The number of rules of this model is denoted by ro, and their activation levels by $w_o^i(k)$, $i = 1, ..., ro$. Substituting now the description of the controller (2.74) into the fuzzy process model (2.62) – but with ro rules with activation levels $w_o^i(k)$ – we obtain a formula describing the closed-loop control system

$$\begin{aligned} \dot{x}(t) &= \frac{\sum_{i=1}^{ro} \sum_{j=1}^{r} w_o^i(t) w^j(t) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(t)}{\sum_{l=1}^{ro} \sum_{p=1}^{r} w_o^l(t) w^p(t)} \\ &= \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) (\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_j) x(t) \end{aligned} \tag{2.75}$$

Of course, if a model used to represent the process in the closed-loop control system is different than the model (2.61)-(2.62) which was used for the design

of the controller, then its matrices are, generally, also different. However, in the control system description (2.75) only the model with matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ representing the process in the closed loop occurs.

In order to examine stability of a dynamic system described by (2.75) it is possible to directly apply Theorem 2.8 or Theorem 2.9, only matrices $\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j$, $i = 1, ..., ro$, $j = 1, ..., r$ instead of the matrices $\mathbf{A}_i$, $i = 1, ..., r$ must be considered. However, if the number of rules and its antecedents used to describe a process in the closed-loop control system are the same as in the fuzzy description of the controller (thus $ro = r$ and $w_o^i(k) = w^i(k)$), then (2.75) can be simplified – in an analogous way as it was done in Section 2.2.1 with the formula (2.25) for systems with discrete time. Namely, applying equalities $w^i(k)w^j(k) = w^j(k)w^i(k)$, $i, j = 1, 2, ..., r$, (2.75) can be transformed to the form

$$\dot{x}(t) = \sum_{i=1}^{r} \tilde{w}^i(t)\tilde{w}^i(t)(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i)x(t) + 2 \sum_{i,j=1,i<j}^{r} \tilde{w}^i(t)\tilde{w}^j(t)\mathbf{D}_{ij}x(t) \quad (2.76)$$

where

$$\mathbf{D}_{ij} = \frac{1}{2}[(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j) + (\mathbf{A}_j - \mathbf{B}_j\mathbf{F}_i)], \quad i < j,\ i, j = 1, 2, ..., r \quad (2.77)$$

Sufficient stability conditions can now be formulated in the following form, see [132].

Theorem 2.10 *Let the maximum number of rules activated simultaneously in the control system (2.76) described by the rules of the process model (2.61) and controller (2.73) be no higher than s, $1 < s \leq r$. Then the equilibrium point of this system is globally asymptotically stable, if there exist a positive definite matrix $\mathbf{P}$ and a positive semi-definite matrix $\mathbf{Q}$ satisfying the following conditions*

$$(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i)^T\mathbf{P} + \mathbf{P}(\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i) + (s-1)\mathbf{Q} < 0, \quad i = 1, 2, ..., r$$
$$\mathbf{D}_{ij}^T\mathbf{P} + \mathbf{P}\mathbf{D}_{ij} - \mathbf{Q} < 0, \quad i < j,\ i, j = 1, 2, ..., r$$

except for pairs (i, j) for which always $w^i(t)w^j(t) = 0$. □

In particular, for $\mathbf{Q} = \mathbf{0}$ we obtain a slightly sharper formulation of the stability conditions which can be used for initial stability verification. In [140] a further, rather slight weakening of the assumptions of Theorem 2.10 was obtained, using the same Lyapunov function $V(x(t)) = x^T(t)\mathbf{P}x(t)$. The obtained conditions in a form of a set of LMIs (Linear Matrix Inequalities) are however much more complex.

It is also possible to try to use Theorem 2.9 for the case of a closed-loop control system, then we obtain

Corollary 2.11 *The equilibrium point of a dynamic system (2.76) is globally asymptotically stable if*

$$\lambda_{\max}([\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j] + [\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_j]^T) < 0, \quad i, j = 1, 2, ..., r$$

except for pairs (i, j) *for which always* $w^i(t)w^j(t) = 0$. □

If it is necessary to reconstruct the state vector, it is possible to use a nonlinear state observer in the form of a TS fuzzy system with a rule structure and antecedents identical to those of the process model and the controller, but with functional consequents of the rules which are classical state observer equations, see [132].

Example 2.7
The task of stabilization of an inverted pendulum attached to a cart, presented in Fig. 2.27, will be considered. Equations describing process dynamics are as follows (see *e.g.*, [106, 141])

$$\dot{x}_1 = x_2 \tag{2.78a}$$

$$\dot{x}_2 = \frac{g \sin x_1 - aml(x_2)^2 \sin(2x_1)/2 - a\cos(x_1)u}{4l/3 - aml\cos^2 x_1} \tag{2.78b}$$

where x_1 denotes the angle between actual position of the pendulum arm and vertical direction ($0\,[rad]$ corresponds to the vertical position of the pendulum), hence $\dot{x}_1 = x_2$ denotes angular velocity. Masses of the pendulum and the cart are denoted by M and m, respectively, $2l$ is its length and $a = 1/(M+m)$. A force imposed on the cart is the control variable u, $g = 9.81\,[m/sec^2]$ is the gravity acceleration. The following numerical values are assumed: m = $2\,[kg]$, M = $8\,[kg]$, 2l = $1\,[m]$ (as in [150, 141], where the control of the presented process is also investigated).

It is known that stabilization of the pendulum in the vertical position by a linear controller works correctly for a limited range of angular deviations. We shall design a simple TS fuzzy controller consisting of only two rules, operating more efficiently and for a wider range of deviations. The first operating region will be around the zero position, denoted by X_1 and defined by a two-sided sigmoidal membership function in the following form

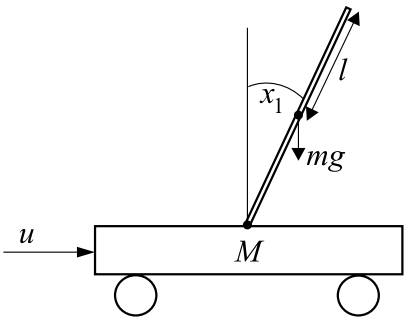

Fig. 2.27. Inverted pendulum on a cart

$$\text{sigmf1}(x) = \frac{1}{1+\exp[-15(x_1-\pi/8)]} - \frac{1}{1+\exp[-15(x_1+\pi/8)]} \tag{2.79}$$

and the second region will be around the positions $x_1 = \pm\pi/4$, denoted by X_2 and defined by the membership function

$$\text{sigmf2}(x) = 1 - \text{sigmf1}(x) \tag{2.80}$$

In order to design a fuzzy controller we shall make an approximation of the nonlinear description (2.78a)-(2.78b) by a TS fuzzy model with the rules

$$\text{R}_p^1: \text{ IF } \; x_1(t) \text{ is } X_1 \quad \text{THEN} \quad \dot{x}^1(t) = \mathbf{A}_1 x(t) + \mathbf{B}_1 u(t) \tag{2.81a}$$

$$\text{R}_p^2: \text{ IF } \; x_1(t) \text{ is } X_2 \quad \text{THEN} \quad \dot{x}^2(t) = \mathbf{A}_2 x(t) + \mathbf{B}_2 u(t) \tag{2.81b}$$

where matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ will now be designed.

First linear model will be designed for the first region around the equilibrium point [0 0]. Performing a standard linearization we obtain

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ \frac{g}{4l/3-aml} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 17.3118 & 0 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{-a}{4l/3-aml} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.1765 \end{bmatrix}$$

However, performing a standard linearization (based on Taylor series expansion) of the original nonlinear model (2.78a)-(2.78b) at a non-zero point $[\pi/4,\ 0]$, we would obtain an affine model, not a linear one which is used for construction of a local state-feedback matrix. A possible solution is to use a slightly different way of linearization, as proposed in [141]. We shall proceed along these lines, using the general structure of our nonlinear model, namely

$$\dot{x}(t) = \mathbf{f}(x(t)) + \mathbf{G}(x(t))u(t)$$

It is required that the linear model $\dot{x}(t) = \mathbf{A}_2 x(t) + \mathbf{B}_2 u(t)$ approximates a nonlinear description, in the vicinity of the point $x_0 = [\pi/4 \;\; 0]$, in such a way that

$$\mathbf{f}(x) + \mathbf{G}(x)u \approx \mathbf{A}_2 x + \mathbf{B}_2 u$$

and

$$\mathbf{f}(x_0) + \mathbf{G}(x_0)u = \mathbf{A}_2 x_0 + \mathbf{B}_2 u, \; u \in \mathbb{R}$$

It follows directly from these conditions that

$$\mathbf{B}_2 = \mathbf{G}(x_0)$$

Thus, what remains to be fulfilled is

$$\mathbf{f}(x) \approx \mathbf{A}_2 x \;\text{ and }\; \mathbf{f}(x_0) = \mathbf{A}_2 x_0$$

Let us denote by $\mathbf{a}_{2j}$ rows of the matrices $\mathbf{A}_2$, $j = 1, 2$. It was shown in [141] that to satisfy the conditions formulated above the rows $\mathbf{a}_{2j}$ should result from the solution of the following optimization problem

$$\min_{\mathbf{a}_{2j}} \{0.5 \|\nabla f_j(x_0) - \mathbf{a}_{2j}\|_2^2\}$$
$$\text{subj. to: } \mathbf{a}_{2j}^T x_0 = f_j(x_0)$$

where $\mathbf{f} = [f_1 \ f_2]^T$. An analytical solution of the formulated quadratic optimization problem is given by the formula

$$\mathbf{a}_{2j} = \nabla f_j(x_0) + \frac{f_j(x_0) - x_0^T \nabla f_j(x_0)}{\|x_0\|^2} x_0, \quad x_0 \neq 0$$

Using the above reasoning we obtain the following matrices $\mathbf{A}_2$ and $\mathbf{B}_2$:

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ \frac{4g \sin(\pi/4)}{\pi[4l/3 - aml \cos^2(\pi/4)]} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 14.3223 & 0 \end{bmatrix}$$
$$\mathbf{B}_2 = \begin{bmatrix} 0 \\ \frac{-a \cos(\pi/4)}{4l/3 - aml \cos^2(\pi/4)} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.1147 \end{bmatrix}$$

A TS fuzzy controller will be described by the rules

$$\mathrm{R}_c^1: \quad \text{IF} \quad x_1(t) \text{ is } X_1 \quad \text{THEN} \quad u^1(t) = -\mathbf{F}_1 x(t)$$
$$\mathrm{R}_c^2: \quad \text{IF} \quad x_1(t) \text{ is } X_2 \quad \text{THEN} \quad u^2(t) = -\mathbf{F}_2 x(t)$$

and the final inference formula

$$u(t) = w^1(t)\, u^1(t) + w^2(t)\, u^2(t)$$

where $w^1(t)$ and $w^2(t)$ are activation levels of the rules (let us note that it follows from the construction of membership functions (2.79) and (2.80) that $w^1(t) + w^2(t) = 1$). Matrices $\mathbf{F}_1$ and $\mathbf{F}_2$ will be obtained in a standard way, assuming appropriate placement of eigenvalues of matrices $\mathbf{A}_i - \mathbf{B}_i \mathbf{F}_i$, $i = 1, 2$. If we require that for both linear models these values should be -2 and -2.5, then using the command *place* from the *Control Systems Toolbox* of the MATLAB® package, we obtain

$$\mathbf{F}_1 = [-126.4125 \ -25.4958]$$
$$\mathbf{F}_2 = [-168.4595 \ -39.2328]$$

For simulations of the closed-loop control system the original nonlinear process model (2.78a)-(2.78b) was used and the designed TS fuzzy controller, which generated the control signal according to

$$u(t) = -w^1(t)\, \mathbf{F}_1\, x(t) - w^2(t)\, \mathbf{F}_2\, x(t)$$

Selected trajectories of state variables $x(t)$ for initial conditions $x_1(0)$ equal to 0.25, 0.5 and 1.0 $[rad]$ and $x_2(0) = 0\,[rad/sec]$ are presented in Fig. 2.28. For a comparison, Fig. 2.29 presents trajectories in the control system with the fuzzy controller and a linear controller designed for the first area ($u(t) = -\mathbf{F}_1 x(t)$), for initial state $x_1(0) = 0.5$, $x_2(0) = 0$. Generally, the further the pendulum position from a vertical one the faster the operation of the nonlinear fuzzy controller. Moreover, it regulates the pendulum to a vertical position from a wider range of initial states. It still operates starting from $x_1(0) = 1.0$, while the linear controller is already unstable for $x_1(0) = 0.85$.

In order to check the quality of fuzzy modeling of the considered strongly nonlinear process, simulations were also conducted for the control system with the process represented by the TS fuzzy model given by (2.81a)-(2.81b). Trajectories obtained in this control system are shown by dashed lines in Fig. 2.30, where trajectories corresponding to the control system with the original nonlinear process are also shown (solid lines). In the range of smaller angular deviations the coincidence of these two types of trajectories is very good (the trajectories for $x_1(0) = 0.25$ are almost undistinguishable).

The above analysis is interesting since we can apply theoretical stability criteria when both the process and the controller are described by fuzzy models in the control system loop. Applying Theorem 2.10 with $\mathbf{Q} = \mathbf{0}$, it can be concluded that the sufficient stability condition is satisfied, with the matrix

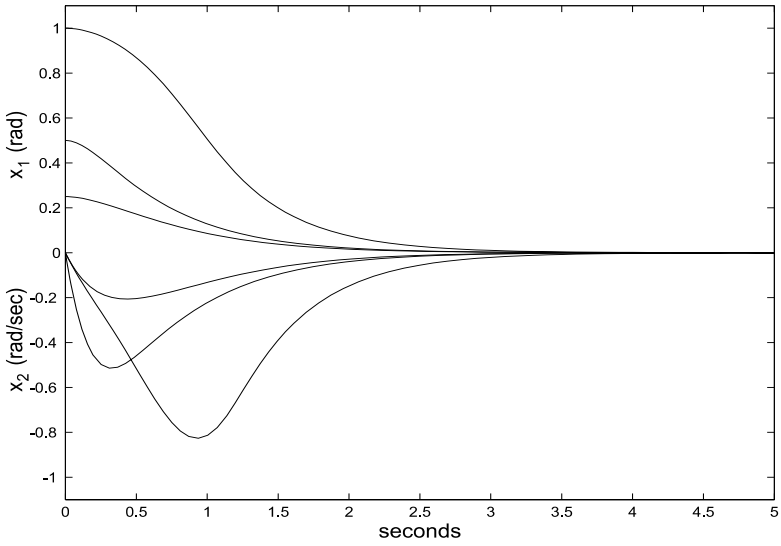

Fig. 2.28. Trajectories in the control system with the TS fuzzy controller, for various initial conditions

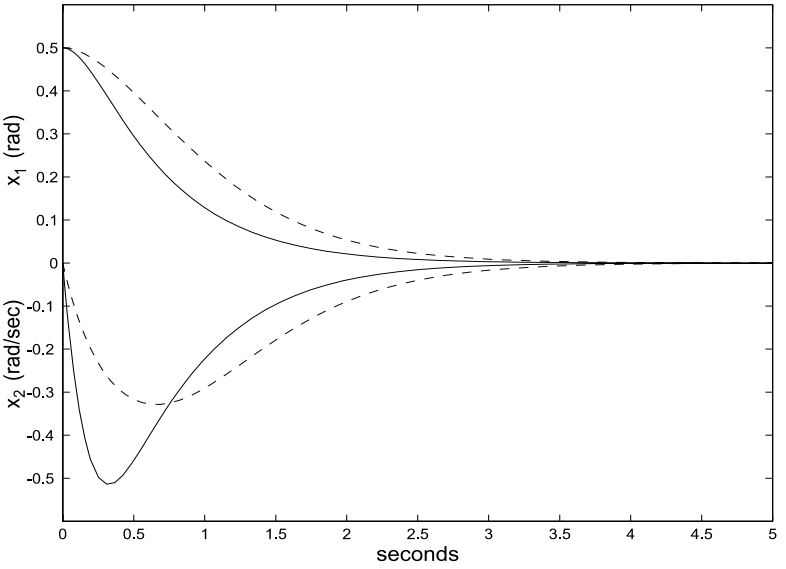

Fig. 2.29. Trajectories in the control system with the TS fuzzy controller (solid line) and with the linear controller (dashed line)

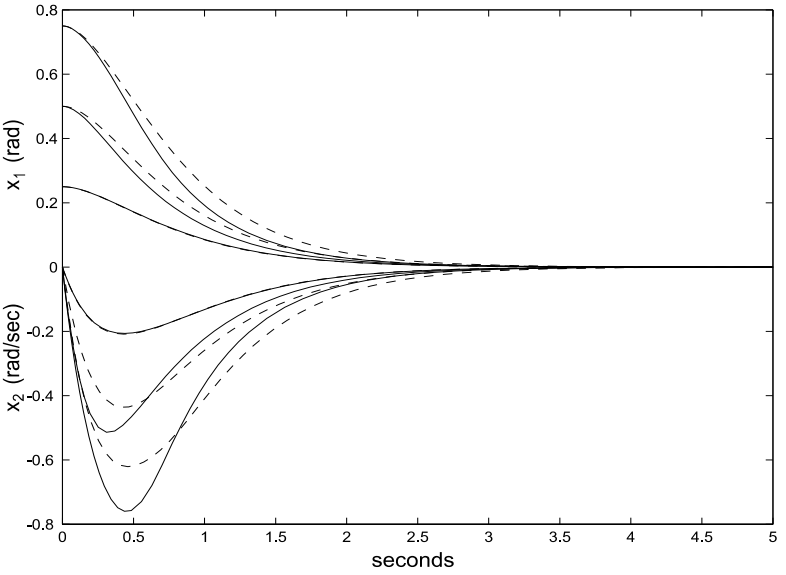

Fig. 2.30. Trajectories in control systems with the TS fuzzy controller and the original nonlinear process model (solid lines), and with the TS fuzzy process model (dashed lines)

$$\mathbf{P} = \begin{bmatrix} 1.2529 & 0.1025 \\ 0.1025 & 0.1461 \end{bmatrix}$$

It was also investigated that the control system is stable with different (perturbed) state-feedback matrices $\mathbf{F}_i$, corresponding to eigenvalues of the matrices $\mathbf{A}_i - \mathbf{B}_i\mathbf{F}_i$ from the range -1 to -4.

If credibility of this sort of analysis should be increased, it would be necessary to construct a more precise TS fuzzy model, with a larger number of rules.

Applying Corollary 2.11 to the stability analysis of the considered control system does not lead to a constructive result, unfortunately – the matrices inspected have positive eigenvalues. This confirms the author's opinion about a conservative character of assumptions of Theorem 2.9 and about limited and unclear, so far, range of usefulness of this theorem. □

2.3.2 Continuous TS Fuzzy Output-feedback Controllers

In practice of industrial process control the most important role is played by controllers with feedback from process outputs and with settings based on an approximate process model, as a state vector is usually not available for current measurements and we rarely have a precise model of the process dynamics at our disposal. Especially, in direct control loops PID controllers are still dominant. As mentioned in the introduction to this chapter, in situations of frequent changes of operating points and significant nonlinearities, the PID linear controllers do not always ensure appropriate quality of the control. One of the most effective solutions then is to apply a TS fuzzy control with feedback from measured process outputs. In Section 2.2.2 problems of the design and analysis of discrete-time TS fuzzy output-feedback controllers were presented. In this section we shall deal with continuous TS fuzzy output-feedback controllers, designed for a continuous-time process description. The obtained continuous control algorithm is then transformed into a discrete one by one of the emulation methods and implemented in a digital controller, with the sampling period sufficiently small in relation to the dynamics of the controlled process.

The considerations will be limited to the case of a PID controller which is the most important one in practice. The methodology of the design will be, however, of a general type, applicable to any linear dynamic control algorithm.

The basic question when starting a design of a continuous TS fuzzy controller, analogously as it was in the case of a discrete controller, is the choice of variables for antecedents of the controller rules and a design of fuzzy sets for these variables. That is, a partitioning of domains of these variables into mutually overlapping fuzzy sets defining altogether overlapping sub-domains (multivariable fuzzy sets) of the whole process domain. This should be done in such a way that in each of these sub-domains the process can be controlled by an appropriately selected linear controller, designed for an operating point

central for the sub-domain. Process operating points are related to certain values of process states, therefore the variables occurring in the rule antecedents of a TS fuzzy model will be, first of all, state variables describing nonlinear behavior of the process. These will be mainly the controlled variable and its derivatives. The variables of the rule antecedents will be denoted by $x_1(t), ..., x_n(t)$, and fuzzy sets describing the partitioning of the domain of each of them will be denoted by A_j^i, $j = 1, ..., n_x$.

Having defined the fuzzy sub-domains of the process operational space and its central points (process operating points), in each of them a local continuous PID controller is tuned, *e.g.*, using one of the known, standard tuning methods. In this way a *continuous TS fuzzy PID* (FPID) controller has been obtained, defined by the rules

$$\begin{aligned} \mathrm{R}_c^j: \quad \text{IF} \quad & x_1(t) \text{ is } A_1^j \text{ and } x_2(t) \text{ is } A_2^j \text{ and } \cdots \text{ and } x_{n_x}(t) \text{ is } A_{n_x}^j \\ \text{THEN} \quad & u^j(t) = k_P^j \left[e(t) + \frac{1}{T_I^j} \int_0^t e(\tau) d\tau + T_D^j \frac{de(t)}{dt} \right] \end{aligned} \tag{2.82}$$

where $j = 1, \ldots, r$ indexes rules (and at the same time local operating sub-domains), $e(t) = y^{sp}(t) - y(t)$ is the control error at time t, whereas settings of the local PID controllers: gain k_P^j, integral time constant T_I^j and derivative time constant T_D^j are coefficients of the functions of the rule consequents.

The output signal of the controller takes a standard form, for TS fuzzy structures,

$$u(t) = \frac{\sum_{j=1}^r w^j(t)\, u^j(t)}{\sum_{l=1}^r w^l(t)} = \sum_{j=1}^r \tilde{w}^j(t)\, u^j(t) \tag{2.83}$$

where $w^j(t)$ are activation levels of individual rules (2.82) at time t,

$$w^j(t) = \prod_{p=1}^{n_x} \mu_{A_p^j}(x_p(t)) \tag{2.84}$$

and $\tilde{w}^j(t)$ are their normalized values.

Implementing (2.82) and (2.83) in a simulation environment, such as Simulink®, we obtain a realization of a continuous TS fuzzy PID controller. In order to perform simulations of the entire system it is also necessary to have a nonlinear model of the dynamic process itself, formulated in a form accepted by the employed simulation environment. If we want to support the simulation experiments by a theoretical analysis of stability of the control system, then we currently have methods of such an analysis using process and controller models formulated in the form of state equations. State equations describe the dynamics directly in the time domain, a domain which is natural for formulation of nonlinear fuzzy models and for analysis of stability of their equilibrium points. The basic tool for this analysis is the Lyapunov method, see *e.g.*, [61, 148, 132, 141, 21, 140], but almost all publications concern

systems with controllers with linear state-feedback (only in [23] an output-feedback controller is considered). In the sequel, we shall present an analysis of stability of a continuous control system with a TS fuzzy controller, limiting the derivation of appropriate equations to the case of a PI type controller.

To describe the dynamics of a closed-loop control system it is necessary to have a TS fuzzy process model, using a model with the same rule antecedents as in controller rules (2.82) will simplify the analysis. Obviously, it can also be a fuzzy model with a different number of rules and different form of their antecedents, in general. Similarly as in the description of control systems in previous sections, we shall denote the number of process model rules by $ro \neq r$. A functional consequent of each rule is a local linear process model given in the form of the following state equations

$$\dot{x}^i(t) = \mathbf{A}_i x(t) + \mathbf{B}_i u(t)$$
$$y^i(t) = \mathbf{C} x^i(t)$$

$i = 1, 2, ..., ro$. Denoting normalized activation levels of the process model rules by $\tilde{w}_o^i(k)$ we obtain equations of the entire nonlinear process model in the form

$$\dot{x}(t) = \sum_{i=1}^{ro} \tilde{w}_o^i(t)[\mathbf{A}_i x(t) + \mathbf{B}_i u(t)] \tag{2.85}$$
$$y(t) = \mathbf{C} x(t) \tag{2.86}$$

We shall add to these equations dynamics of the integrator of the control error

$$\dot{q}(t) = y^{sp} - \mathbf{C} x(t)$$

The rules of a PI controller have the following form

$$\mathrm{R}_c^j: \text{ IF } \quad x_1(t) \text{ is } A_1^j \text{ and } x_2(t) \text{ is } A_2^j \text{ and } \cdots \text{ and } x_{n_x}(t) \text{ is } A_{n_x}^j$$
$$\text{THEN } \quad u^j(t) = k_P^j e(t) + \frac{k_P^j}{T_I^j} q(t) \tag{2.87}$$

$j = 1, 2, ..., r$, where $e(t) = y^{sp}(t) - \mathbf{C} x(t)$. The controller output is given by

$$\begin{aligned} u(t) &= \sum_{j=1}^{r} \tilde{w}^j(t)[k_P^j e(t) + \frac{k_P^j}{T_I^j} q(t)] \\ &= -\sum_{j=1}^{r} \tilde{w}^j(t) k_P^j \mathbf{C} x(t) + \sum_{j=1}^{r} \tilde{w}^j(t) \frac{k_P^j}{T_I^j} q(t) + \sum_{j=1}^{r} \tilde{w}^j(t) k_P^j y^{sp}(t) \end{aligned} \tag{2.88}$$

Substituting this equation into (2.85) we obtain

$$\dot{x}(t) = \sum_{i=1}^{ro} \tilde{w}_o^i(t) \left\{ [\mathbf{A}_i - \mathbf{B}_i \sum_{j=1}^{r} \tilde{w}^j(t) k_P^j \mathbf{C}] x(t) \right.$$
$$\left. + \mathbf{B}_i \sum_{j=1}^{r} \tilde{w}^j(t) \frac{k_P^j}{T_I^j} q(t) + \mathbf{B}_i \sum_{j=1}^{r} \tilde{w}^j(t) k_P^j y^{sp}(t) \right\}$$
$$= \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ [\mathbf{A}_i - \mathbf{B}_i k_P^j \mathbf{C}] x(t) + \mathbf{B}_i \frac{k_P^j}{T_I^j} q(t) + \mathbf{B}_i k_P^j y^{sp}(t) \right\}$$

If we introduce now the following extended state vector

$$v(t) = [x(t)^T \; q(t)]^T \tag{2.89}$$

then, using the equalities $\sum_{i=1}^{ro} \tilde{w}_o^i(t) = \sum_{j=1}^{r} \tilde{w}^j(t) = 1$, we can write equations of the dynamics of the closed-loop control system in the following form

$$\dot{v}(t) = \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ \begin{bmatrix} \mathbf{A}_i - \mathbf{B}_i k_P^j \mathbf{C} & \mathbf{B}_i \frac{k_P^j}{T_I^j} \\ -\mathbf{C} & 0 \end{bmatrix} v(t) + \begin{bmatrix} \mathbf{B}_i k_P^j \\ 1 \end{bmatrix} y^{sp}(t) \right\}$$
$$= \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) \left\{ [\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}] v(t) \right\} + \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) \tilde{\mathbf{B}}_{ij} y^{sp}(t)$$

where

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix}, \; \mathbf{G}_{ij} = \begin{bmatrix} \mathbf{B}_i k_P^j \mathbf{C} & -\mathbf{B}_i \frac{k_P^j}{T_I^j} \\ \mathbf{0} & 0 \end{bmatrix}, \; \tilde{\mathbf{B}}_{ij} = \begin{bmatrix} \mathbf{B}_i k_P^j \\ 1 \end{bmatrix}$$

Without loos of generality we can assume that $y^{sp}(t) = 0$. Then equations of the dynamics of our system take the following form

$$\dot{v}(t) = \sum_{i=1}^{ro} \sum_{j=1}^{r} \tilde{w}_o^i(t) \tilde{w}^j(t) [\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}] v(t) \tag{2.90}$$

which is identical to the dynamics of the state-feedback control system considered in the previous section described by (2.75), where matrices $\mathbf{A}_i$ and $\mathbf{B}_i \mathbf{F}_j$ in (2.75) correspond to matrices $\tilde{\mathbf{A}}_i$ and $\mathbf{G}_{ij}$ in (2.90). To examine stability of the dynamic system described by (2.90) it is possible to apply directly Theorem 2.8 or Theorem 2.9, only matrices $\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}$, $i = 1, ..., ro$, $j = 1, ..., r$ should be considered instead of matrices $\mathbf{A}_i$, $i = 1, ..., r$.

Moreover, if we assume that the number and antecedents of the rules in the process model and in the controller are the same, then defining the matrix

$$\tilde{\mathbf{D}}_{ij} = \frac{1}{2}[(\tilde{\mathbf{A}}_i - \mathbf{G}_{ij}) + (\tilde{\mathbf{A}}_j - \mathbf{G}_{ji})], \quad i < j, \; i, j = 1, 2, ..., r \tag{2.91}$$

analogous to the matrix $\mathbf{D}_{ij}$ (2.77), we obtain (2.90) in a transformed form

$$\dot{v}(t) = \sum_{i=1}^{r} w^i(t)w^i(t)(\tilde{\mathbf{A}}_i - \mathbf{G}_{ii})v(t) + 2\sum_{i,j=1,i<j}^{r} w^i(t)w^j(t)\tilde{\mathbf{D}}_{ij}v(t) \quad (2.92)$$

corresponding to (2.76). Thus, to analyze stability of the dynamic system (2.92) it is possible to apply directly Theorem 2.10 (also Corollary 2.11), only matrices $\mathbf{A}_i$, $\mathbf{B}_i\mathbf{F}_i$ and $\mathbf{D}_{ij}$ occurring there should be replaced with matrices $\tilde{\mathbf{A}}_i$, $\mathbf{G}_{ii}$ and $\tilde{\mathbf{D}}_{ij}$.

The method of an analysis of a continuous control system with an output-feedback controller was presented above by way of the example of a PI controller. In a similar way, we can treat another cases with linear dynamic output-feedback controllers.

Let us note that due to linearity of the rule consequents (2.82) the dependence (2.83) can be written in the following form:

$$\begin{aligned} u(t) &= \sum_{j=1}^{r} \tilde{w}^j(t) \cdot k_P^j \left[e(t) + \frac{1}{T_I^j}\int_0^t e(\tau)d\tau + T_D^j \frac{de(t)}{dt} \right] \\ &= \sum_{j=1}^{r} \tilde{w}^j(t)k_P^j \cdot e(t) + \sum_{j=1}^{r} \tilde{w}^j(t)\frac{k_P^j}{T_I^j} \cdot \int_0^t e(\tau)d\tau + \sum_{j=1}^{r} \tilde{w}^j(t)k_P^j T_D^j \cdot \frac{de(t)}{dt} \end{aligned} \quad (2.93)$$

Therefore, the considered fuzzy controller can be treated as a nonlinear PID controller, with the parameters varying in a nonlinear way, continuously adapting to the changes in the process operating point, because $\tilde{w}^j(t) = \tilde{w}^j(x(t))$, see (2.84). This feature, together with the relative simplicity and clarity of the design method, has been decisive in the success of TS fuzzy controllers and especially of TS fuzzy PID (TS-FPID) controllers. The above interpretation indicates also that in the case of an identical structure of linear consequents in all rules of a TS fuzzy controller, as in the considered case of the TS-FPID controller, on-line changing its settings can be treated as applying a kind of *gain scheduling* technique, see *e.g.*, [2]. However, it is performed in a completely different way: not by using a nonlinear function approximating an inverse of the process nonlinearity, but by using a more general and simpler mechanism of fuzzy reasoning.

Example 2.8

A nonlinear control of pH value of a mixture in a continuous-flow reactor presented in Fig. 2.31 will be considered.

Dynamics of the reactor is described by the following equations:

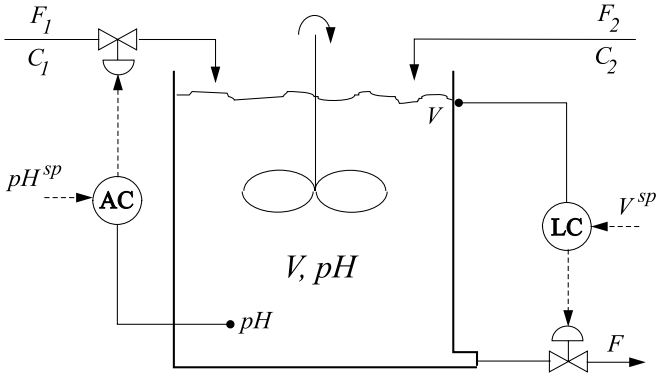

Fig. 2.31. Structure of pH and level control in a continuous-stirred tank reactor, Example 2.8

$$\frac{dV\xi}{dt} = F_1C_1 - (F_1 + F_2)\xi \tag{2.94}$$

$$\frac{dV\psi}{dt} = F_2C_2 - (F_1 + F_2)\psi \tag{2.95}$$

$$\frac{dV}{dt} = F_1 + F_2 - F \tag{2.96}$$

$$\left[H^+\right]^3 + (K_a + \psi)\left[H^+\right]^2 + (K_a(\psi - \xi) - K_w)\left[H^+\right] - K_aK_w = 0 \tag{2.97}$$

where

$\xi \sim [HAC] + [AC^-]$,

$\psi \sim [Na^+]$,

$pH = -\log_{10}[H^+]$,

$C_1 = 0.32\,[\frac{mol}{l}]$ – acid concentration in the inflow F_1,

$C_2 = 0.05005\,[\frac{mol}{l}]$ – acid concentration in the inflow F_2,

$V = 1000\,[l]$ – volume of the mixture,

K_a i K_w – equilibrium constants of acid and water,

$K_a = 1.8 \times 10^{-5}$, $K_w = 1.0 \times 10^{-14}$,

and the following nominal values of the inflow rates were assumed:

$F_1(0) = 81\,[\frac{l}{min}]$, $F_2(0) = 512\,[\frac{l}{min}]$.

There are two feedback control loops in the system, see Fig 2.31:

- Control of the volume V (level control), where the outflow rate F is the manipulated variable,
- Concentration of pH, where the inflow rate F_1 is the manipulated variable,

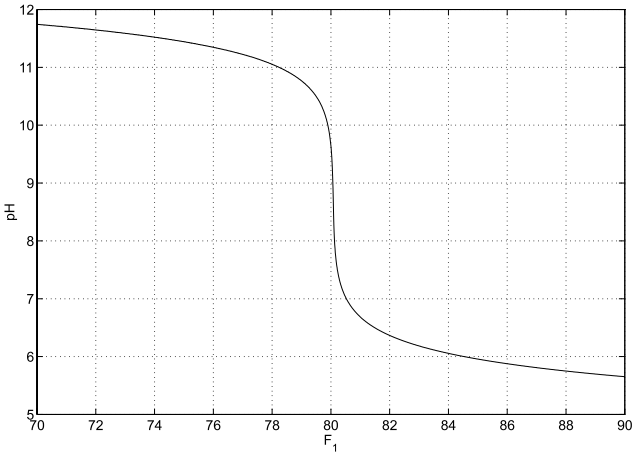

Fig. 2.32. Static dependence of pH versus F_1

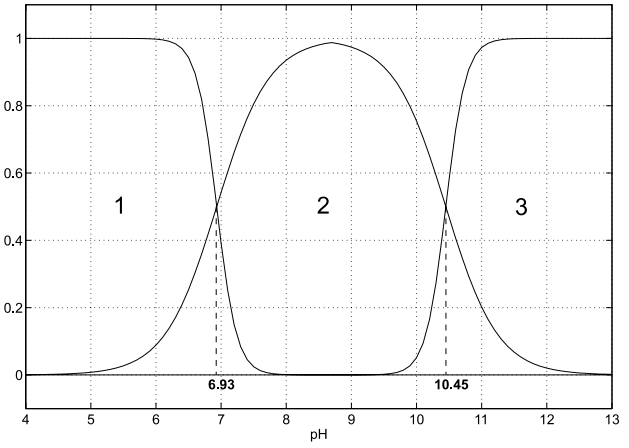

Fig. 2.33. Membership functions of the fuzzy pH controller

while the inflow rate F_2 is a disturbance (uncontrolled process input).

The first feedback control loop is standard and a typical linear controller easily ensures a sufficient control quality. However, controlling the pH value turns out to be very difficult, which results from a strongly nonlinear static dependence of pH concentration versus the inflow F_1, as it is shown in Fig. 2.32. Moreover, the operating point of interest ($pH = 8.7$) is located in the area of strongest nonlinearity, which makes it impossible to select one PI controller for the entire domain of the control. A controller operating correctly in the

area of high process gain ($pH = 7.5 \div 10$) operates unacceptably slowly in both areas of smaller gain, *i.e.*, for small and large values of pH. On the other hand, a controller operating better (faster) in these areas becomes unstable in the middle area of high gain. Therefore, a nonlinear TS fuzzy controller was designed. The only variable present in the antecedents of the controller rules is the pH value. A partition of the pH domain into three fuzzy sets was assumed, with sigmoidal membership functions presented in Fig. 2.33. The PI controllers were designed for each of the fuzzy sets, with settings given in Table 2.1.

Table 2.1. Parameters of local PI controllers

controller$j$	k_p^j	T_I^j
$j = 1$	-20	1.57
$j = 2$	-1	1.7
$j = 3$	-20.2	1.55

The controller algorithm was applied in the form (2.93), *i.e.*,

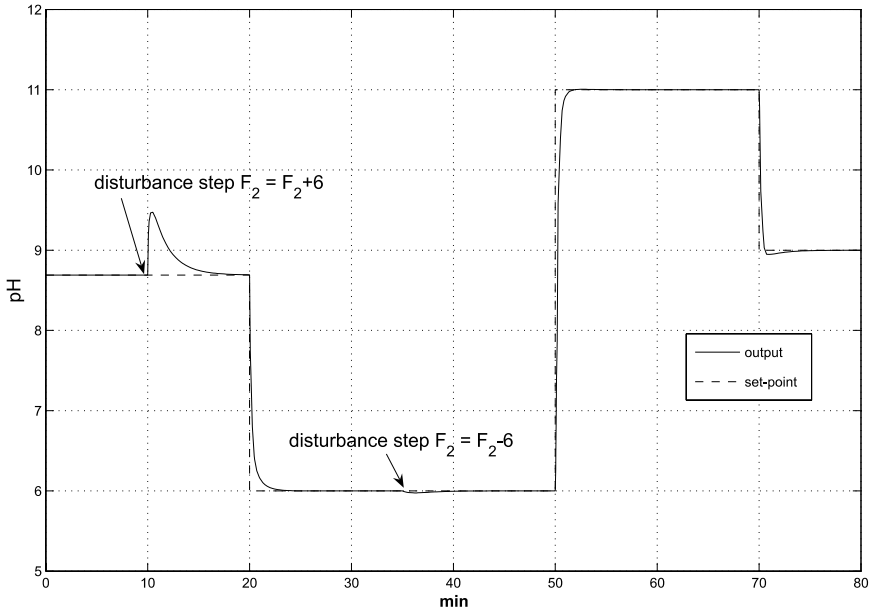

Fig. 2.34. Output trajectory in the pH control system with the TS fuzzy PI controller

$$u(t) = \sum_{j=1}^{3} \tilde{w}^j(t) k_P^j \cdot e(t) + \sum_{j=1}^{3} \tilde{w}^j(t) \frac{k_P^j}{T_I^j} \cdot \int_0^t e(\tau) d\tau + \sum_{j=1}^{3} \tilde{w}^j(t) k_P^j T_D^j \cdot \frac{de(t)}{dt}$$

where

$$e(t) = pH(t)^{sp} - pH(t)$$

and $\tilde{w}^j(t)$ are normalized activation levels of the rules with antecedents in the form : "$pH(t)$ is pH_j", where pH_j, $j = 1, 2, 3$ are linguistic values "small", "medium" and "high" of the linguistic variable "pH value", corresponding to three fuzzy sets described by the membership functions shown in Fig. 2.33.

A chosen result of the control system simulation, corresponding to step changes of the set-point and of the disturbance (inflow rate F_2) is presented in Fig. 2.34. This is the result of a simulation with a continuous process and a discretized controller, with the sampling period $T_p = 0.02\,\text{min}$. □

2.4 Feedforward Compensation, Automatic Tuning

Measured Disturbance and Feedforward Compensation

A designer of control systems should always obey the well-known general principle that *the influence of measured disturbances*, *i.e.*, uncontrolled but measured inputs, *should be compensated in a feedforward structure* – if only influence of disturbances on the controlled variables is significant and dynamics of the process enables effective realization of the feedforward path (roughly, influence of the manipulated input on the controlled output should not be slower then influence of the disturbance). Feedforward control is a control in an open-loop structure, therefore it is faster and more effective than in the closed-loop structure provided its mentioned applicability conditions are satisfied. Then, for control in the feedback loop, there remains reduction of influences of unmeasured disturbances and errors caused by inaccuracies in the process models used for the design of the feedback controller and feedforward compensator. The feedback control structure with measured disturbance compensation in the feedforward path is presented in Fig. 2.35.

The feedforward compensation of disturbances operates independently of the closed-loop feedback control, it does not influence the closed-loop stability conditions. The control signal u consists of a feedback controller signal u_c and feedforward compensator signal u_f. The design of a compensator algorithm is independent of the design of a feedback controller algorithm. Nonlinear TS fuzzy controllers considered in this chapter, both in their discrete and continuous versions, generate the control signal u_c. Control structures with these controllers can be easily supplemented by modules realizing compensation of the measured disturbances, as shown in Fig. 2.35. The design of a compensator algorithm is based on generation of a signal u_f compensating the influence of the disturbance z on the process output y. If G and G_z denote transfer functions of the control path and of the disturbance path, respectively, *i.e.*,

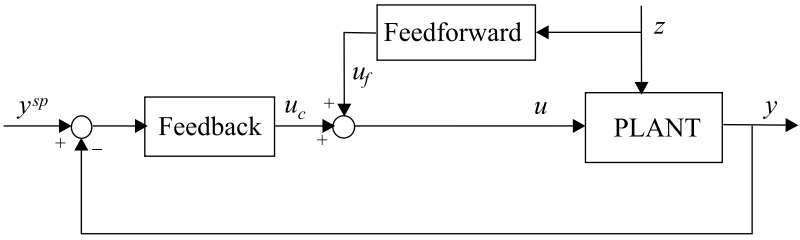

Fig. 2.35. Feedback-feedforward control structure with measured disturbance compensation in the feedforward path

$$y = Gu + G_z z$$

then the output signal of an ideal compensator should be equal to

$$u_f = -G^{-1} G_z z \tag{2.98}$$

Certainly, a difficulty in a the design of a compensator results from the fact that the inverse of the transfer function G should be stable and the transfer function of the compensator must be physically implementable. Most important condition for a successful compensation is that time delay in the control path should not exceed the one in the disturbance path and that the transfer function of the control path must not have unstable zeros, see *e.g.*, [45, 3, 52]. Moreover, accuracy of the process model used for compensator design is important for quality of compensation – but this aspect is not critical for the structure from Fig. 2.35, as inaccuracies of compensation can be reduced or eliminated in the feedback loop.

To design a compensator it is necessary to have a process model. In this chapter nonlinear processes described by TS fuzzy models were considered. In the presence of measured disturbances significantly influencing the process behavior, the TS fuzzy models will also depend on these variables. The disturbances will therefore be present in the formulations of rule antecedents and consequents. Let us consider a case of discrete-time models described by difference equations. Denoting the value of a disturbance measured at k-th sampling instant by $z(k)$, the rules of a *TS fuzzy model used for the design of a fuzzy controller with disturbance compensation* can be written in the following very general form (compare with (2.36)):

$$\begin{aligned} \mathrm{R}_p^i: \quad \text{IF} \quad & y(k) \text{ is } A_0^i \text{ and } y(k-1) \text{ is } A_1^i \text{ and } \cdots \text{ and } y(k-n_R) \text{ is } A_{n_R}^i \\ & \text{and } z(k) \text{ is } Z_0^i \text{ and } z(k-1) \text{ is } Z_1^i \text{ and } \cdots \text{ and } z(k-p_R) \text{ is } Z_{p_R}^i \\ & \text{and } u(k-1) \text{ is } B_1^i \text{ and } \cdots \text{ and } u(k-m_R) \text{ is } B_{m_R}^i \end{aligned}$$

$$\begin{aligned}\text{THEN}\quad y^i(k+1) = {} & a_1^i y(k) + a_2^i y(k-1) + \cdots + a_{n_A}^i y(k-n_A+1) + \\ & + b_0^i u(k) + b_1^i u(k-1) + \cdots + b_{m_B}^i u(k-m_B) + \\ & + f_0^i z(k) + f_1^i z(k-1) + \cdots + f_{p_F}^i z(k-p_F) \end{aligned} \tag{2.99}$$

where $i = 1, \ldots, r$ indexes the rules (and corresponding fuzzy sets) in the domain of a TS model, $y(k)$, $z(k)$ and $u(k)$ are values of output, disturbance and control input at k-th sampling instant, $A_j^i \in \mathbb{Y}_j$, $B_j^i \in \mathbb{U}_j$, $Z_j^i \in \mathbb{Z}_j$, and a_j^i, b_j^i and f_j^i are coefficients of functions in the rule consequents. Elements of each set $\mathbb{Y}_j = \{Y_{j1}, ..., Y_{jr_{yj}}\}$ are fuzzy sets covering the domain of the variable $y(k-j)$, $j = 0, \ldots, n_R$, analogously $\mathbb{U}_j = \{U_{j1}, ..., U_{jr_{uj}}\}$ for $u(k-j)$, $j = 1, \ldots, m_R$ and $\mathbb{Z}_j = \{Z_{j1}, ..., Z_{jr_{zj}}\}$ (see comments after (2.36)). The form of the rule antecedent (2.99) looks rather complicated due to the assumed generality. However, in many applications there are only a few conditions there, in particular delayed variables are often not present, if there are process outputs y then there are no process inputs u, *etc.*

The output of a fuzzy model is calculated according to the general formula

$$y(k+1) = \frac{\sum_{i=1}^{r} [w^i(k) y^i(k+1)]}{\sum_{l=1}^{r} w^l(k)} \tag{2.100}$$

wherte $w^i(k)$ are activation levels of individual rules (2.99) at sampling instant k,

$$w^i(k) = \prod_{j=0}^{n_R} \mu_{A_j^i}(y(k-j)) \prod_{j=0}^{p_R} \mu_{Z_j^i}(z(k-j)) \prod_{j=1}^{m_R} \mu_{B_j^i}(u(k-j)) \tag{2.101}$$

For each linear model in the rule consequents (2.99) both a feedback controller algorithm as well as a feedforward compensator algorithm are designed. Design of a controller was considered in Section 2.2.2 – the only difference caused by the presence of a measured disturbance can be an enlarged number of rules, due to a nonlinear dependence on an additional disturbance variable. Every local compensator algorithm is designed according to the general law of feedforward compensation, shortly recalled in (2.98). For the considered fuzzy modeling method it will be in the form of a difference equation. This way, we obtain the following rules of a *discrete, nonlinear TS fuzzy compensator*

$$\begin{aligned}\text{R}_f^i:\quad \text{IF}\quad & y(k) \text{ is } A_0^i \text{ and } y(k-1) \text{ is } A_1^i \text{ and } \cdots \text{ and } y(k-n_R) \text{ is } A_{n_R}^i \\ & \text{and } z(k) \text{ is } Z_0^i \text{ and } z(k-1) \text{ is } Z_1^i \text{ and } \cdots \text{ and } z(k-p_R) \text{ is } Z_{p_R}^i \\ & \text{and } u(k-1) \text{ is } B_1^i \text{ and } \cdots \text{ and } u(k-m_R) \text{ is } B_{m_R}^i \end{aligned}$$

$$\begin{aligned}\text{THEN}\quad u_f^i(k) = {} & g_1^i u_f(k-1) + g_2^i u_f(k-2) + \cdots + g_{m_G}^i u_f(k-m_G) + \\ & + h_0^i z(k) + h_1^i z(k-1) + \cdots + h_{p_H}^i z(k-p_H) \end{aligned} \tag{2.102}$$

where $i = 1, \ldots, r$ indexes rules, while g_j^i and h_j^i are coefficients of functions of the rule consequents describing local compensators. The nonlinear compensator output signal assumes a standard form for TS fuzzy structures

$$u_f(k) = \sum_{i=1}^{r} \tilde{w}^i(k) u_f^i(k) \tag{2.103}$$

where $\tilde{w}^i(k)$ are normalized activation levels of the rules (2.102). Compensation of the disturbance by each of the r local compensators is linear, but due to the fuzzy reasoning the entire TS fuzzy compensator defined by (2.102) and (2.103) is of course *nonlinear*.

Automatic Tuning of a TS-FPID Controller

In modern industrial PID controllers it is now the standard to have a software for automatic tuning (auto-tuning, self-tuning) installed, operating on the basis of a certain, previously programmed active experiment, usually a relay control phase, a pulse response or a multi-frequency response. By activating this software option the user automatically obtains settings calculated by the controller at the current operating point of the process, the user does not have to know the algorithm or assumed process model used for doing that.

If one applies the mentioned self-tuning of linear controllers, then in the first stage of the design of a TS fuzzy controller it will not be necessary to have local process models needed for a design of the rule consequents of a fuzzy controller. It will, however, be necessary to perform fuzzy partitioning and allocate positions of operating points in appropriately chosen centers of the obtained fuzzy sets (sub-domains). Then, self-tuning will be performed at each of these points.

In a case of a confidence in the settings of the PID controllers obtained by the automatic self-tuning procedure, sufficiently dense fuzzy partitioning (in relation to the process nonlinearity) and confidence in the reliability of the controller software implementing fuzzy reasoning, it is possible to test the TS-FPID controller straight on a real application. Such a solution was implemented in certain fuzzy controllers, *e.g.*, it was the case in EFTRONIK XF controller manufactured by PNEFAL company [114]. The user of this controller defines the number and position of operating points (up to ten), at each of them a self-tuning is activated by the user and on the basis of obtained parameters of local PID controllers a TS-FPID controller is created automatically, in its final, discrete form. Certainly, an important design parameter is also the controller sampling period.

http://www.springer.com/978-1-84628-634-6

Advanced Control of Industrial Processes
Structures and Algorithms
Tatjewski, P.
2007, XIX, 332 p., Hardcover
ISBN: 978-1-84628-634-6