Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems

Faculty of Electrical Engineering, Computer Science and Telecommunications
University of Zielona Góra

Lecture Notes in Control and Computer Science
Volume 9

Marek Kowal

Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems

University of Zielona Góra Press, Poland
2005

Marek KOWAL
Institute of Control and Computation Engineering
University of Zielona Góra
ul. Podgórna 50
65-246 Zielona Góra, Poland
e-mail: `M.Kowal@issi.uz.zgora.pl`

Referees:

- Danuta RUTKOWSKA, Częstochowa University of Technology
- Wiesław MOCZULSKI, University of Zielona Góra

The text of this book was prepared based on the author's Ph.D. dissertation

Partially supported by the Ministry of Scientific Research and
Information Technology in Poland

ISBN 83-89712-88-1

Camera-ready copy prepared from the author's LaTeX2$_\varepsilon$ files.
Printed and bound by the University of Zielona Góra Press, Poland.

CONTENTS

Acknowledgments

I would like to express my sincere gratitude and thanks to Professor Józef Korbicz for inspiration, continuous support, and supervision of this work. In addition, I wish to thank all colleagues at the Institute of Control and Computation Engineering, who helped me in many ways while I was preparing this work. I direct special thanks to Marcin Witczak, Marcin Mrugalski and Eugen Arinton for their help and valuable comments.

Notation

Symbols

$\mathbb{R}$	set of real numbers
k	discrete time
$\boldsymbol{p}$	parameter vector
$\hat{\boldsymbol{p}}$	parameter estimate vector
$\boldsymbol{r}$	input vector
$\boldsymbol{x}$	input vector for the neuro-fuzzy network
y	model output
y'	system output
n	number of parameters
N	number of measurements
N_r	number of fuzzy rules
$\mathbb{P}$	admissible set of parameters
$\mathbb{E}$	ellipsoid
$\boldsymbol{\varepsilon}$	disturbances
$\boldsymbol{e}_r$	residual vector

Abbreviations

N-F	Neuro-Fuzzy
LS	Least-Squares method
BEA	Bounded-Error Approach
OBE	Outer Bounding Ellipsoid
LP	Linear in Parameters
NLP	Non-Linear in Parameters

Chapter 1

INTRODUCTION

Nowadays, diagnostics systems are becoming crucial elements of technical and non-technical applications. A wide range of technical ventures have no sense without diagnostic systems due to the necessity of assuring a high level of safety and continuity of work in industrial applications. It is also hard to imagine contemporary medicine without specialist diagnostic equipment that helps doctors to make diagnoses. Industrial systems and medical problems become more and more complicated in the course of years and, consequently, require more sophisticated diagnostic systems. In the case of simple technical systems, human inspection was enough but the increased complexity of inspected systems and the high level of process quality, reliability and safety requirements required the automation of diagnostics in order to make it possible to precisely determine the place, reason and time of the fault. One of the well-known diagnostic methods employs the mathematical model of a system to diagnose it (Chen *et al.*, 1999; Frank and Ding, 1997; Gertler, 1998; Isermann, 1993; Ljung, 1987; Patton *et al.*, 2000; Witczak, 2003).

A model of a system is usually used to generate symptom signals that describe the state of the system, and this step of the diagnosis is named *fault detection*. Fault detection is essential for correct localization and identification of the fault, which are the next steps of fault diagnosis (Gertler, 1998; Kościelny, 2004; Patton *et al.*, 2000; Pau, 1981). Three main types of models can be distinguished: analytical, empirical and heuristical ones (Korbicz and Bidyuk, 1993; Rutkowski and Cpałka, 2002; Kowal and Korbicz, 2003; Kościelny, 2001; Szulim, 2004; Mendes *et al.*, 2002).

Out of those, empirical models appear very useful in the case of diagnostic applications. The design of such models does not require knowledge about physical or mathematical laws that describe the behavior of the system. The empirical model design is based on system identification algorithms (Ljung, 1987; Söderström and Stoica, 1997) using available measurements. The identification procedure is employed to determine the structure of the model and its parameters to copy the behavior of the system as well as possible. It is quite impossible to build an accurate model for real industrial plants using analytical methods due to strong non-linearities and stochastic behavior (Ljung, 1987; Söderström and Stoica, 1997; Korbicz and Bidyuk, 1993; Nahorski and Mańczak, 1983). Therefore, there is a great demand for developing and employing for this task alternative solutions, which will be able to ensure the required accuracy for models of real plants.

In the last decade a particular interest in the analyzed areas has regarded artificial intelligence methods (Korbicz *et al.*, 2004; Patton and Chen, 1999; Patton and Korbicz, 1999; Kościelny, 2001). The attractiveness of such methods in the context of diagnostics results from great capabilities of artificial intelligence methods in general (Korbicz *et al.*, 1994; Rutkowska *et al.*, 1997; Rutkowska and Zadeh, 2000; Rutkowski, 2004b; Osowski, 1996; Tadeusiewicz, 1993). The following important properties of these methods should be stressed: learning from examples, the ability to adapt to a changing environment, non-linear mapping (Landajo *et al.*, 2001), the ability to utilize heuristic knowledge in the design process. One of significant elements of the artificial intelligence area is the neuro-fuzzy (N-F) technique (Rutkowska, 1997; Rutkowska, 2003; Rutkowski, 2004b; Czogała and Łęski, 2000; Korbicz and Kowal, 2001). This method arose through the combination of the specific properties of neural networks and fuzzy logic. N-F techniques are widely used in many areas including diagnostics due to fast and intensive development of different algorithms for structure design and training of N-F networks. One of the main advantages of the N-F approach in the context of diagnostics is the transparency of knowledge coded in the structure of diagnostic systems. It simplifies greatly the analysis of knowledge that determines the operations of the system, which is a very important property, especially in diagnostic applications.

The application of hybrid techniques such as N-F networks to fault diagnosis using a model-based scheme is considered in this work. Practically, there is no design technique that can generate an ideal model and, therefore, it is important to assure for a fault detection system robustness against disturbances. Obviously, this problem concerns N-F models, too. The inaccuracy of modelling is usually described as model uncertainty and the measure of the uncertainty is defined in the form of a confidence interval for the model output.

Robust fault detection under model uncertainty is the main requirement for modern fault detection systems. Robustness in this case is considered as the insensitivity of the fault detection system to model uncertainty. Two main approaches can be distinguished in robust fault detection: an active one and a passive one. Methods like parity relations (Gertler and Kowalczuk, 1997) and observers with an unknown input (Witczak, 2003) represent the active approach. The main idea of these approaches is the special design method, which should eliminate the influence of the unknown input on the residual signal, so that the fault detection system can be robust to disturbances. Unfortunately, the existing techniques can be applied without problems only to linear objects. In the case of non-linear systems the analyzed methods are applicable for a narrow range of systems. Passive methods are based on the adaptive threshold method, which is used to describe model uncertainty (Frank and Ding, 1997; Seliger and Frank, 2000). Unfortunately, there exist some limitations that make such a method difficult to be applied to a wide range of systems. For example, the known methods require detailed information about the type and properties of disturbances that corrupt measurements. Such knowledge is usually unavailable in the case of fault diagnosis of industrial plants. Moreover, the techniques used to determine uncertainty can be applied effectively only to a narrow class of models that are linear in parameters (Mańczak, 1976; Walter and Pronzato, 1997; Rafajłowicz, 1996).

The main objective of this work is to develop new, robust fault detection methods under model uncertainty, where the N-F network is used to implement the model. For this purpose, an effective method of computing the uncertainty of the N-F model and the development of a new adaptive threshold method are the main tasks of research. It is also required to develop a new method for N-F network design that should take into account the uncertainty of the N-F model.

The work is divided into six chapters. Chapter 1 deals with general problems with N-F model design taking into consideration its application to fault detection. Some approaches are described and problems with their application are shown.

Chapter 2 gives an overview of hybrid N-F structures and, especially, N-F networks. Problems with the application of N-F networks to fault detection tasks are considered in detail.

In Chapter 3, reasons for uncertainty in N-F models are presented. Next, the use of the statistical method and the bounded-error approach (BEA) for computing the uncertainty is described. A simplified method that approximates the confidence region of N-F network parameters is also employed to compute the uncertainty due to the complexity of the BEA method and limitations of statistical approaches. A detailed analysis of the presented approaches in the context of fault detection applications to real plants is given in the final part of the chapter.

Chapter 4 presents the N-F model structure selection problem. The common methods for structure selection of the N-F network are summarized and the advantages and disadvantages of these methods are indicated. A new method for the selection of fuzzy rules that employs the BEA method for detecting approximately linear parts of the model characteristic is proposed in the final part of that chapter.

Chapter 5 briefly discusses technical diagnostics. Problems with robust fault detection are considered in detail. A robust fault detection method based on the adaptive threshold technique and model uncertainty is employed using N-F models. Additionally, examples of robust fault detection of an electrical engine and some components of an industrial system from the Lublin Sugar Factory in Poland using the proposed approaches are described and shown in figures and tables.

The last part of the work includes the summary, conclusions and the list of original achievements developed and described in the work.

Chapter 2

NEURO-FUZZY NETWORKS

2.1. Introduction

One of the main areas in the process diagnostic field is research concerning an effective use of artificial intelligence techniques (Korbicz *et al.*, 2004; Kościelny, 2001; Patton and Chen, 1999; Patton and Korbicz, 1999). The interest in this research area in the fault diagnosis context is the effect of potential capabilities of artificial intelligence methods (Duch *et al.*, 2000; Korbicz *et al.*, 1994; Osowski, 1996; Tadeusiewicz, 1993; Rutkowska *et al.*, 1997; Rutkowska and Zadeh, 2000). The main advantages of AI methods are as follows: learning from samples, the ability to adapt to a changing environment, to realize non-linear mappings, and to utilize heuristic knowledge in the design process. N-F networks hold an important position within AI methods (Rutkowska, 1997; Rutkowska, 2002; Kowal and Korbicz, 2003; Czogała and Łęski, 2000).

The N-F network technique was created by combining specific properties of artificial neural networks and fuzzy logic. The intensive development of design algorithms and learning methods results in many applications of N-F networks in different areas including fault diagnosis (Chen *et al.*, 1999; Kowal and Korbicz, 2003; Mendes *et al.*, 2002; Kowal and Korbicz, 2002a; Kowal and Korbicz, 2002b; Kowal, 2001; Calado *et al.*, 2002). N-F networks can be used for system modelling, fault classification, decision support systems, etc. in fault diagnosis applications. The main advantage of N-F techniques in the context of fault diagnosis is the transparency of knowledge coded in the form of fuzzy rules. This simplifies the analysis of rules that determine the behavior of the model, which is important in order to ensure reliable fault detection.

Effective applications of N-F methods to fault diagnosis require tackling the problem of structure selection and tuning the parameters for the N-F model taking into account its uncertainty.

The main aim of this chapter is to present the theoretical background of N-F networks as well problems that are encountered when N-F networks are employed for fault detection. A short introduction to fuzzy systems is presented at the beginning, and then the common N-F structures are presented. Since this work concentrates on the N-F network, this technique is described more deeply than others. Some aspects concerning its use in fault detection are studied in detail.

2.2. Fuzzy systems

2.2.1. Fuzzy models

Fuzzy systems can be seen as logical models, which establish the relationships between variables in the form of fuzzy rules:

$$\textit{IF (antecedent proposition) THEN (consequent proposition).} \tag{2.1}$$

Depending on the particular structure of the consequent proposition, three types of models are distinguished: linguistic, relational and Takagi-Sugeno ones (Babuška, 1998).

2.2.1.1. Linguistic fuzzy model

The linguistic model represents the relationships in the form of the *IF ... THEN ...* fuzzy rule, where both the antecedent and the consequent are fuzzy propositions (Zadeh, 1973; Mamdani, 1977). A general form of the linguistic fuzzy rule is

$$IF\ \boldsymbol{x}\ is\ A\ THEN\ \boldsymbol{y}\ is\ B, \tag{2.2}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ and $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$ are fuzzy variables, which are represented by fuzzy sets $\boldsymbol{x} \in F(\boldsymbol{X})$ and $\boldsymbol{y} \in F(\boldsymbol{Y})$. However, in real applications, the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are usually numerical variables $\boldsymbol{x} \in \boldsymbol{X} \subset R^n$ and $\boldsymbol{y} \in \boldsymbol{Y} \subset R^m$. The fuzzy sets A and B in Eqn. (2.2) represent linguistic terms defined by multivariate membership functions $\mu(\boldsymbol{x}) : \boldsymbol{X} \to [0, 1]$.

An alternative approach considers the input as the vector of single input variables $x_1, x_2, \ldots, x_n$, thus multivariate fuzzy sets are decomposed to the set of univariate fuzzy sets $A_1, A_2, \ldots, A_n$. Such an approach allows representing by fuzzy sets the meaning of linguistic terms, which is more difficult in the case of multidimensional fuzzy sets. The logical operators of conjunction, disjunction and a negation are used in order to construct a compound proposition from univariate fuzzy sets. For instance,

$$IF\ x_1\ is\ A_1\ OR\ x_2\ is\ A_2\ AND\ x_3\ is\ NOT\ A_3 \ldots\ THEN\ \boldsymbol{y}\ is\ \boldsymbol{B}. \tag{2.3}$$

Here, the global antecedent consist of many simple antecedents and the degree of fulfilment of the global antecedent is computed using the logical operators AND, OR, NOT, where the logical operators are defined using an appropriate T-norm, S-norm and negation operator. Moreover, the approach can be simplified if only a conjunction operator is used to compose the antecedent:

$$IF\ x_1\ is\ A_1\ I\ x_2\ is\ A_2\ AND\ \ldots\ AND\ x_n\ is\ A_n\ THEN\ \boldsymbol{y}\ is\ \boldsymbol{B}, \tag{2.4}$$

$$\beta = \mu_{A_1}(x_1) \wedge \mu_{A_2}(x_2) \wedge \ldots \wedge \mu_{A_n}(x_n). \tag{2.5}$$

The expression (2.5) determines the global fulfilment of the rule and is in fact a multidimensional membership function of the multivariate fuzzy set $\mu_A(x)$ created

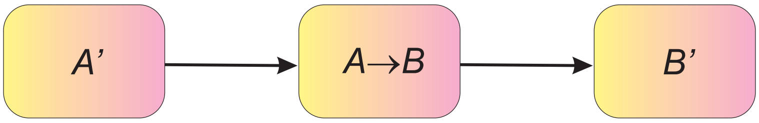

Fig. 2.1 . Generalized *modus-ponens* inference

by the intersection of the univariate sets $A_1, A_2, \ldots, A_n$ on the Cartesian product space $X = X_1 \times X_2 \times \ldots \times X_n$. The shape of the multidimensional membership function depends on the particular operation which is used to define the conjunction between the fuzzy sets.

In previous deliberations only a single *IF ... THEN ...* rule was taken into account but in real problems a set of fuzzy rules is usually employed to describe reality and this set is called the rule base:

$$R_k: \; IF \; x_1 \; is \; A_1^k \; AND \; \ldots \; AND \; x_n \; is \; A_n^k \; THEN \; \boldsymbol{y} \; is \; \boldsymbol{B^k}. \tag{2.6}$$

A single rule in the rule base is responsible for computing the output, but only for a certain region of input values in the whole input space. This region is constrained by the membership function that describes the antecedent of the rule. The partitioning of the input space differs depending on the type of antecedent fuzzy sets. Antecedent multivariate membership functions assure the most general way of partitioning the input space as there is no restriction on the shape of fuzzy sets. Various partitions of the antecedent space can be obtained with multidimensional membership functions generated by intersecting univariate fuzzy sets. The boundaries are restricted to a rectangular grid defined by the fuzzy sets of individual variables. In this case the number of rules needed to cover the entire domain is a product of the input space dimension with the number of fuzzy sets used for each variable. Univariate fuzzy sets assure a strong connection between the fuzzy set and the linguistic term but the approach with multivariate fuzzy sets provides a more effective representation of partitions in the input space.

An inseparable element of fuzzy systems is the fuzzy inference procedure (Driankov *et al.*, 1993; Rutkowska, 1997; Rutkowska *et al.*, 1997; Yager and Filev, 1994). Inference is a process of deriving an output fuzzy set for each fuzzy rule, given the rules and the inputs. In the case of the linguistic fuzzy model the generalized *modus-ponens* inference rule is used. Such a rule may be demonstrated by the scheme shown in Fig. 2.1. The scheme reveals the main difference between the *modus-ponens* inference from the traditional 0, 1 logic and generalized *modus-ponens*, which is expressed in the ability of the generalized method to generate inference results for premises that are slightly different from the antecedent of the rule, thus the conclusion can be a little different than the consequent of the rule.

A fuzzy rule can be regarded as the fuzzy relation $R : (X \times Y) \to [0, 1]$. The fuzzy relation can be considered as the fuzzy implication $A \to B$, by means of the following expression:

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)), \tag{2.7}$$

where the operator I describes the type of fuzzy implication. Given the fuzzy rule in the form of the fuzzy set A', the output fuzzy set B' is derived by the following composition of the premise A' and the relation R:

$$B' = A' \circ R. \tag{2.8}$$

The composition is realized by the $SUP - T$ composition between A' and R. The membership function of the fuzzy set B' is obtained from the following equation:

$$\mu_{B'}(y) = \sup_{x}\{T_{x,y}[\mu_{A'}(x), \mu_R(x,y)]\}. \tag{2.9}$$

The fuzzy linguistic model is simplified in many practical applications. For example, input values may not be fuzzy. The inference procedure in this case is simplified and can be described in three steps. At the beginning, the global fulfilment of the rules is determined (2.5) in order to calculate conclusions using the T-norm operator. The inference procedure is applied to each rule separately and the number of conclusions is equal to the number of rules. In order to achieve a single conclusion, the calculated fuzzy sets are aggregated:

$$B' = \bigcup_{i=1}^{N} B'_i, \tag{2.10}$$

$$\mu_{B'}(y) = \bigcup_{i=1}^{N} \mu_{B'_i}(y). \tag{2.11}$$

For the minimum conjunction operator, the minimum T-norm and aggregation defined as a maximum, the well-known Mamdani scheme of fuzzy inference is achieved, which is also called the $MAX - MIN$ inference:

$$\mu_{B'}(y) = \max_{x} \min_{x,y}(\mu_{A_{i,j}}(x_j), \mu_{B_i}(y)). \tag{2.12}$$

The shown simplification is usually used in real applications where inputs have crisp values. Moreover, the results of inference can be transformed from fuzzy sets into crisp values by a defuzzification algorithm. A lot of different defuzzification algorithms have been proposed, but it should be noted that all these algorithms lead to information loss due to a change of a fuzzy value into a single crisp value.

2.2.1.2. Fuzzy relational model

Fuzzy relational models encode associations between linguistic terms defined in the system's input domain and represented by the fuzzy sets $A_1, A_2, \ldots, A_n$ and linguistic terms defined in the output domain $B_1, B_2, \ldots, B_m$ (Pedrycz, 1984). A fuzzy relation defined in the input-output linguistic space defines the mapping $R : A \to B$, where each A_i is related to each B_j as shown in Fig. 2.2. The fuzzy

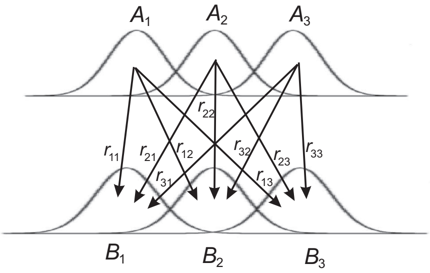

Fig. 2.2 . Relational model

relation can be defined using the table

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \vdots & \vdots & \dots & \vdots \\ r_{n1} & r_{n2} & \dots & r_{mn} \end{pmatrix}_{m \times n} . \tag{2.13}$$

In fact, the table stores the rule base because each row of this table can be translated into a fuzzy IF THEN rule:

$$IF\ x\ is\ A_i\ THEN\ y\ is\ B_1(r_{i1})\ AND\ B_2(r_{i2})\ AND\ \dots\ AND\ B_m(r_{im}). \tag{2.14}$$

Fuzzy inference in such a system is made by the composition of the fuzzy set $X = [\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)]$, which determines the firing levels of the individual sets $A_1, A_2, \dots, A_n$, with the relation R. The result of the composition

$$Y = X \circ R \tag{2.15}$$

gives the fuzzy set $Y = [\mu_{B_1}(y), \mu_{B_2}(y), \dots, \mu_{B_m}(y)]$. The last stage of inference considers the aggregation of conclusions and, finally, the fuzzy set is transformed into a crisp value using the chosen defuzzification method.

2.2.1.3. Takagi-Sugeno model

Practical industrial applications usually have to operate on crisp measurements and should give crisp results. Such requirements can be fulfilled by the fuzzy model proposed in (Takagi and Sugeno, 1985). In the Takagi-Sugeno fuzzy model, rule consequents are crisp functions of model inputs:

$$IF\ \boldsymbol{x}\ is\ A\ THEN\ \boldsymbol{y} = f(\boldsymbol{x}), \tag{2.16}$$

where $\boldsymbol{x} \in R^n$ is the input variable, $\boldsymbol{y} \in R^m$ is the output variable, A is the multivariate fuzzy set, n and m respectively the numbers of input variables and output variables. For the sake of simplicity it is assumed that only one output variable is given, $y \in R$. Like in a fuzzy linguistic model, a multivariate fuzzy set in the antecedents can be decomposed into univariate fuzzy sets connected by logic operators. The Takagi-Sugeno model with univariate fuzzy sets is easier to analyze due to transparent knowledge - univariate fuzzy sets have direct linguistic meaning. Rules with univariate fuzzy sets and conjunction operator are usually used in Takagi-Sugeno models:

$$IF\ x_1\ is\ A_1\ AND\ x_2\ isA_2\ AND\ \ldots\ AND\ x_n\ is\ A_n\ THEN\ y = f(\boldsymbol{x}). \tag{2.17}$$

Consequents, as has been mentioned above, are the functions of inputs and have the same structure for all rules. The only difference between the rules are the values of the parameters of these functions. In real applications, consequents are usually defined in the form of linear functions:

$$y = \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{x} + b, \tag{2.18}$$

where $\boldsymbol{a}$ is the vector of parameters, b is the *bias*.

Generally, the Takagi-Sugeno system consists of many rules, and fuzzy antecedents define fuzzy regions of input values for which the corresponding consequents are active. The global output of the Takagi-Sugeno system is the combination of outputs from all defined rules. The main advantage of using linear functions as consequents of fuzzy rules is the possibility of using for such a Takagi-Sugeno system a lot of theorems developed for linear system, i.e. stability testing, tuning the parameters, input variable selection (Driankov *et al.*, 1993; Matia *et al.*, 2002; Joh *et al.*, 1998).

A fuzzy system that has consequents defined in the form of *biases* is a special case of the Takagi-Sugeno system (Rutkowska, 1997):

$$IF\ \boldsymbol{x}\ is\ A\ THEN\ y = b. \tag{2.19}$$

It is also a special case of a linguistic system where consequents are described by singleton fuzzy sets.

The inference method used in the Takagi-Sugeno system is very similar to the Mamadani inference. First, the fulfilment of rules is computed. If antecedents are defined using a multivariate fuzzy set, the fulfillment is computed directly from the membership function for the given input value x in contrast to a situation where univariate antecedents connected by the logical AND operators are used, and to compute the fulfilment β_i Eqn. 2.5 has to be applied. The next step of inference is required to generate the conclusions y_i for all rules using the values β_i. The global output of the Takagi-Sugeno system is achieved using the aggregation procedure for all conclusions.

2.3. Neuro-fuzzy systems

Neuro-fuzzy systems are a wide class of hybrid systems that combine some elements of artificial neural networks and fuzzy systems (Rutkowska, 1997; Babuška, 1998; Czogała and Łęski, 2000). In the literature a lot of structures have been proposed,

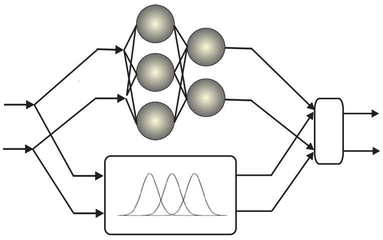

Fig. 2.3 . Parallel combination of a fuzzy system with a neural network

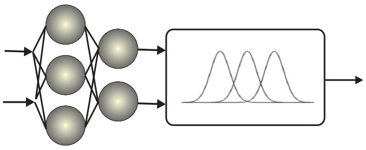

Fig. 2.4 . Cascade combination of a fuzzy system and a neural network

which sometimes are diametrically different (Rutkowska and Zadeh, 2000). One of the presented approaches seems to be easiest to implement because the idea of combination assumes using simultaneously a neural network and a fuzzy system within the framework of a single system. Parallel or cascade structures of neural networks and fuzzy systems are commonly used in Figs. 2.3 and 2.4. The first solution is used to store *apriori* knowledge in the form of fuzzy rules and the neural network is responsible for system adaptation using learning algorithms. In the second approach a neural network or a fuzzy system is responsible for preprocessing the input measurement and then carrying out the main task of the system. Such N-F systems can be applied, e.g. to pattern recognition.

The next hybrid approach considers the use of neural networks in order to compute the parameters of the fuzzy system, which is schematically shown in Fig 2.5. In the learning stage the parameters of the neural network are tuned only, next

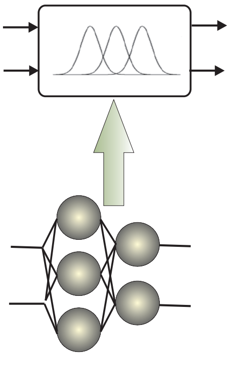

Fig. 2.5 . Neural networks compute parameters for a fuzzy system

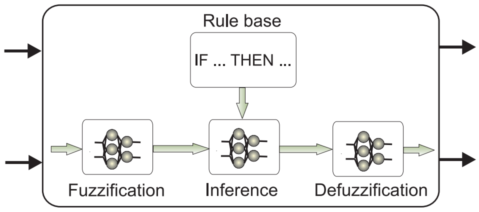

Fig. 2.6 . Neural networks implement some parts of a fuzzy system

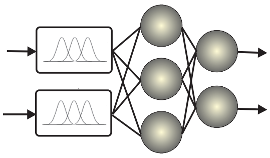

Fig. 2.7 . Neural network process fuzzy values

the parameters are transferred into a fuzzy system and this system is responsible for the main task; however, the fuzzy system does not have learning abilities and for this purpose it relies on the neural network.

The other hybrid approaches assume a closer integration of the neural network with the fuzzy system. Among other things, the realization of some elements of the fuzzy system using neural networks is considered (Fig. 2.6). Different solutions for such an approach are applicable: membership functions can be realized by neural networks, the inference process can be carried out by a neural network, or the defuzzification procedure can be coded in the form of a neural network.

The next approach is based on a modified neural network (Fig. 2.7) which is able to operate using fuzzy numbers, or the weights are fuzzy.

The last approach presents the fuzzy system in the form of a neural network (Fig. 2.8). However, it is not a typical neural network because the individual elements of the network are fuzzy operators that implement the fuzzy inference mechanism. Some elements of the network have parameters, and learning algorithms for neural networks can be used to tune these parameters.

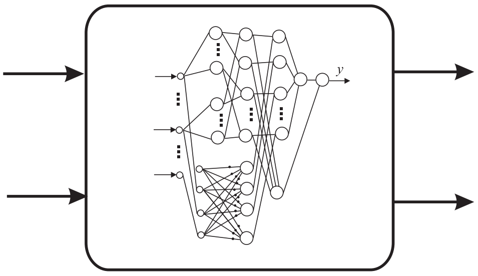

Fig. 2.8 . Neuro-fuzzy network

2.4. Theoretical background of neuro-fuzzy networks

The theory of N-F networks is based on the theory of fuzzy inference (Zadeh, 1973). The difference between fuzzy systems and neuro-fuzzy systems is mainly in the formal representation of the system structure and the methods of tuning the parameters. The neuro-fuzzy system is viewed in the form of a topology similar to the topology of the neural network, there are no equations describing the inference mechanism and there is no list of rules – these elements of the fuzzy system are coded in the form of elements of the network (Rutkowska, 1997; Korbicz and Kowal, 2001). Of course, it is a simple transformation and it does not change the properties of fuzzy inference, it is only another form of realization for the fuzzy system. The crucial difference between fuzzy systems and neuro-fuzzy systems is in the algorithms used to generate the rule base. In the case of N-F systems, learning algorithms for neural networks can be employed to tune the parameters of the rules (Duch *et al.*, 2000; Korbicz *et al.*, 1994; Osowski, 1996; Tadeusiewicz, 1993).

Two types of fuzzy rules are commonly used to describe knowledge in N-F networks:

$$IF\ \boldsymbol{x}\ is\ A\ THEN\ y = \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{x} + b, \tag{2.20}$$

$$IF\ \boldsymbol{x}\ is\ A\ THEN\ y = b. \tag{2.21}$$

These rules correspond to the Takagi-Sugeno fuzzy system (2.20) and to the simplified linguistic fuzzy system (2.21). The system (2.21) is a special case of the Takagi-Sugeno system and in the following discussion only systems with rules described by Eqn. (2.20) are considered.

Data processing in N-F networks similar to fuzzy systems contains three phases: fuzzification, inference, aggregation and defuzzification. The operation of fuzzification is not practically necessary because crisp input values are treated like *singleton* fuzzy sets:

$$\mu(x) = \begin{cases} 1 & \text{if } x = x', \\ 0 & \text{if } x \neq x'. \end{cases} \tag{2.22}$$

However, there exist *non-singleton* defuzzification methods (Rutkowska, 1997; Mouzouris and Mendel, 1997), which are more sophisticated than *singleton* ones, but are rarely used in practice.

The fuzzy inference process is identical with fuzzy inference in the fuzzy Takagi-Sugeno system, where at the beginning the fulfilment of the rules β_i is calculated. If antecedents are described by univariate fuzzy sets, the global fulfilment of the rules is given by (2.5). Next the SUP-T composition (2.9) is applied to compute conclusions for each rule. It should be noted that the SUP-T composition (2.9) works properly only for fuzzy sets and in the analyzed N-F Takagi-Sugeno system only antecedents are described by fuzzy sets because consequents operate on crisp values. To overcome this problem, crisp values of conclusions are treated like *singletons* and the whole theory of SUP-T is applicable for the Takagi-Sugeno

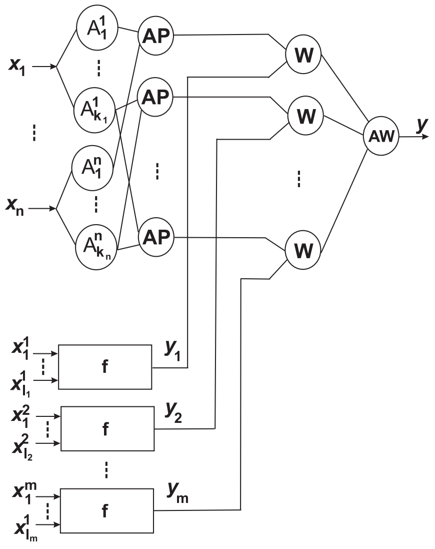

Fig. 2.9 . General structure of the Takagi-Sugeno neuro-fuzzy network

system. The result of inference is the set of *singletons*, and practically the defuzzification procedure is not necessary, only the aggregation algorithm is employed to achieve the global output of the system from the set of conclusions.

The general structure of the Takagi-Sugeno N-F network is shown in Fig. 2.9. Such a network consists of five layers. The first layer contains elements that realize membership functions of fuzzy sets which describe antecedents. The elements of this layer determine the fulfilment of antecedents described by univariate fuzzy sets. The elements of the next layer denoted by AP realize the logical operator AND. The operation realized by these elements is defined by the T-norm. The outputs of the layer give information about the fulfillment of rules. The next layer implements the inference mechanism using the T-norm or fuzzy implication operators. The fourth layer works paralelly to the other layers and its task is to determine crisp values of consequents by computing the value of the function $\boldsymbol{f}$. Generally, the set of input variables defined for the first layer is different than the set of input variables defined for the fourth layer. The last layer of the Takagi-Sugeno N-F network contains one element that is responsible for the aggregation procedure. The specific structure of the N-F network strongly depends on the type of operators used. The typical structure of the Takagi-Sugeno N-F network is shown in Fig. 2.10, where univariate fuzzy sets are used, fuzzy operators are defined as an algebraic product, aggregation is realized using the height method and consequents are defined in the form of linear equations. In practical applications it is often required to model non-linear dynamic systems:

$$y'(k) = f(y'(k-1), \ldots, y'(k-n_y-1), u(k), \ldots, u(k-n_u)), \tag{2.23}$$

where, for the sake of simplicity, only one input variable u and one output variable

y' is considered, f is the unknown function, k is the discrete time, n_y and n_u determine the order of dynamics. It is possible to build different dynamic N-F networks for such a system (Nelles, 2001):

- NARX (Nonlinear AutoRegressive with eXogenous input) model,
- finite impulse response model,
- output error model.

In the case of the NARX model, the N-F network realizes the following mappings:

$$y(k) = F(y'(k-1), \ldots, y'(k-n_y-1), u(k), \ldots, u(k-n_u)), \tag{2.24}$$

where F indicates mappings realized by the N-F network, and $y(k)$ is the predicted value of the real system output $y'(k)$. Such an N-F dynamic model requires the following fuzzy rules:

$$\begin{aligned} &\mathit{IF}\ u(k)\ \mathit{is}\ A_1\ \mathit{AND}\ \ldots\ \mathit{AND} u(k-n_u)\ \mathit{is}\ A_{n_u}\ \mathit{AND} \\ &y'(k-1)\ \mathit{is}\ A_{n_u+1}\ \mathit{AND}\ \ldots\ \mathit{AND}\ y'(k-n_y-1)\ \mathit{is}\ A_{n_u+n_y} \\ &\qquad\qquad\qquad \mathit{THEN}\ y(k) = \boldsymbol{r}^T(k)\boldsymbol{p}, \end{aligned} \tag{2.25}$$

where $\boldsymbol{p}$ is the vector of consequent parameters and $\boldsymbol{r}(k) = [u(k), \ldots, u(k-n_u), y'(k-1), \ldots, y'(k-n_y-1)]$ is the regression vector.

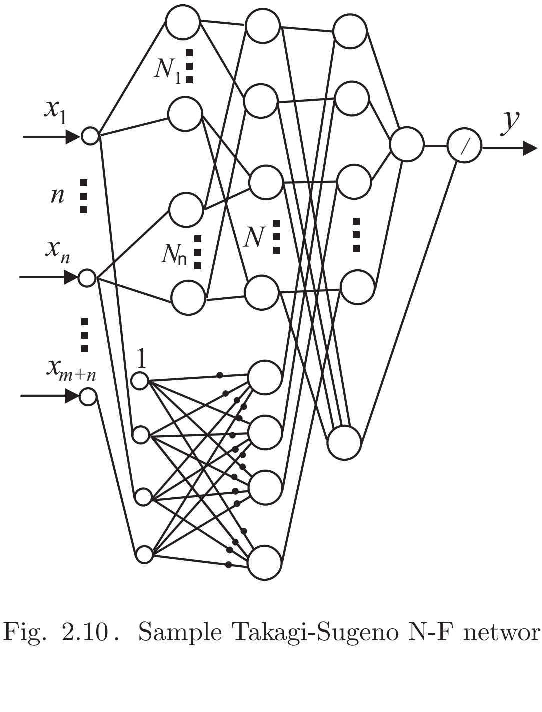

Fig. 2.10. Sample Takagi-Sugeno N-F network

The N-F network in the form of the finite input response model is described by the following equation:

$$y(k) = F(u(k), \dots, u(k-n_u)), \tag{2.26}$$

and the model consists of fuzzy rules:

$$\begin{aligned} &\mathit{IF}\ u(k)\ \mathit{is}\ A_1\ I\ \dots\ \mathit{AND}\ u(k-n_u)\ \mathit{jest}\ A_{n_u} \\ &\qquad\qquad\qquad \mathit{THEN}\ y(k) = \boldsymbol{u}^T(k)\boldsymbol{p}, \end{aligned} \tag{2.27}$$

where $\boldsymbol{u}(k) = [u(k), \dots, u(k-n_u)]$. The main advantage of such a model is that its stability is guaranteed; however, it is usually required to use a great value for n_u to ensure an exact model.

The dynamic output error model has a structure similar to that of the NARX model, but real values of the output variable $y'(k)$ are replaced with output values computed by the model $y(k)$:

$$y(k) = F(y(k-1), \dots, y(k-n_y-1), u(k), \dots, u(k-n_u)). \tag{2.28}$$

Depending on the way the output values are transmitted to the inputs of the N-F network, two ways of dynamic implementations can be distinguished. The first one takes the output value from the global output of the N-F model as shown in Fig. 2.11. In this case fuzzy rules have a structure similar to the rule described by the formula 2.11, only the real output values $y'(k)$ are replaced with $y(k)$. The second

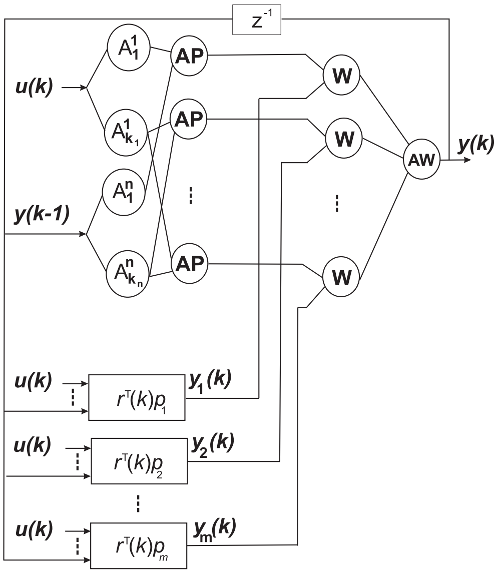

Fig. 2.11 . Dynamic Takagi-Sugeno N-F network

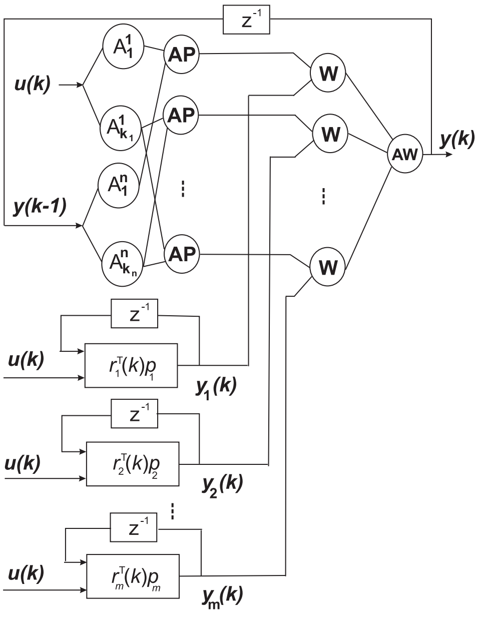

Fig. 2.12 . Dynamic Takagi-Sugeno N-F network with internal dynamics

approach assumes that each rule transmits its delayed local output $y_i(k)$ to its inputs. A sample structure is shown in Fig. 2.12.

All dynamic N-F models presented in this chapter are input-output models, but it possible to build an alternative dynamic N-F model in the state space. Let us assume that a non-linear system is described by the following state space equations:

$$\begin{aligned} x(k+1) &= g(x(k), u(k)), \\ y(k) &= h(x(k)), \end{aligned} \tag{2.29}$$

and its equivalent in the form of an N-F network is defined by the following fuzzy rules:

$$\mathit{IF}\ \boldsymbol{u}(k)\ \mathit{is}\ D_1\ \mathit{AND}\ \boldsymbol{x}(k)\ \mathit{is}\ D_2\ \mathit{THEN}\ \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ y(k) = \boldsymbol{C}\boldsymbol{x}(k) \end{cases} .(2.30)$$

The dynamic fuzzy rules mentioned above have a general form, which can be changed or simplified in practical applications.

2.5. Neuro-fuzzy networks in fault diagnosis

In general, diagnostics are responsible for monitoring characteristic variables which describe the state of the diagnosed system and the detection of states that are abnormal. Input variables, output variables and state variables are usually monitored. The simplest fault diagnosis method consist in observing selected variables in order to check if they exceed the defined thresholds. Unfortunately, such a simple solution is not able to detect faults in complex systems, when an abnormal state is manifested in values of many variables but does not necessarily exceed the defined thresholds. The detection of such faults requires knowledge about physical laws that describe the dependencies between variables in the form of mathematical models. However, it is usually quite impossible to extract such knowledge in the case of real systems. An alternative approach is based on hardware redundancy, which uses a few copies of the same plant for a single task. This solution is usually impractical due to expenses connected with additional equipment. In this case an attractive solution can be analytical redundancy. The main idea of this approach consist in building a model of the diagnosed system and performing fault detection by a comparison of the output signal of the system with the output signal of the model in order to generate residuals. The residual signal indicates the state of the system and is usually computed as a difference between the outputs of the system and the model. In this case the fault-free mode is indicated by residuals equal to 0, and any departure from this rule indicates an abnormal work of the system. Of course, this approach is effective if an ideal model is given and modelling errors do not corrupt the residual signal. A suitable procedure of residual evaluation and then classification is usually applied to identify, localize the fault and to detect the reasons for it.

A model-based fault detection technique is the subject of intensive research in the area of diagnostics due to many important properties in the context of fault detection:

- the ability to detect small-scale faults,
- the ability to detect faults in different working points of the system,
- the solution is relatively cheap because sophisticated equipment is not required; suitable software and computer are usually enough,
- the installation of the fault diagnosis system does not usually require an intervention in the existing system; usually the installed sensors can be useful for data acquisition.

The idea of model-based fault diagnosis is widely used in many industrial applications. However, there are also a lot of cases where a suitable and accurate model cannot be obtained. This usually happens because real systems are complex, have a lot of input variables, and input-output mappings are non-linear. Additionally, systems are usually dynamic. In such cases physical models are not known or are known only partially, thus such an approximation is not precise enough to guarantee the appropriate accuracy. The only available knowledge about the system is given in the form of measurements which describe the behavior of the input and output variables. This leads directly to an identification problem because such methods are able to ensure the required accuracy of the model for diagnostic tasks. The choice of the identification method depends strongly on the model structure. Two main approaches can be distinguished among dynamic system modelling problems: prediction and simulation. In diagnostic applications simulation is used because prediction can hide faults. The simplest approach uses linear models, i.e. input-output models or state space models (Nahorski and Mańczak, 1983; Söderström and Stoica, 1997).

The general description of the input-output model is given by the following equation:

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}\varepsilon(k), \tag{2.31}$$

where q is a delay operator and $\frac{B(q)}{F(q)A(q)}$ and $\frac{C(q)}{D(q)A(q)}$ is a transfer function for the input variable $u(k)$, and $\varepsilon(k)$ represents disturbances. Making different assumptions about the transfer functions $A(q), B(q), C(q), D(q), F(q)$, different structures of the model can be obtained:

- the ARX model (Auto Regresive with eXogenous input), where $C(q) = 1, D(q) = 1$ and $F(q) = 1$,
- the ARMAX model (Auto Regresive Moving Average with eXogenous input), where $D(q) = 1$ and $F(q) = 1$,
- the ARARX model (Auto Regresive Auto Regresive with eXogenous input), where $C(q) = 1$ and $F(q) = 1$,
- the ARARMX model(Auto Regresive Auto Regresive Moving Average with eXogenous input), where $F(q) = 1$,
- the output error model, where $A(q) = 1, C(q) = 1$ i $D(q) = 1$,
- the Box Jenkins model, where $A(q) = 1$.

The right choice of the model structure requires some knowledge about the structure of the system and the type of disturbances that corrupt measurements.

An alternative approach assumes building the model described by the state space equations

$$x(k+1) = Ax(k) + Bu(k) + w(k), \tag{2.32}$$

$$y(k+1) = Cx(k+1) + v(k+1). \tag{2.33}$$

Although the theory of parameter estimation and stability analysis is well developed for linear systems, their usage is limited due to common non-linearities in real systems. The described methods are able to linearize the non-linear characteristic around the working point of the system. Unfortunately, such approaches usually do not assure the required accuracy, which in the case of diagnostic applications has a great influence on their efficiency. Better accuracy can be obtained if non-linear models are applied. As linear approaches, two types of non-linear models can be distinguished: input-output models and state space models.

The classical input-output approach is Kolmogorov-Gabor polynomials. Unfortunately, their usage is limited by the lack of effective methods of choosing the order of dynamics, fast growth of the number of tuned parameters when the number of input variables increases, and wrong interpolation and extrapolation properties. Hammerstein-Wiener models are another technique for modelling non-linear dynamic systems. This approach decomposes the non-linear system into a static non-linear part and a dynamic linear piece (Janczak, 2004). Although the approach simplifies stability analysis or the choice of the dynamic order, the usage of this method is limited to a narrow group of real systems.

In the case of non-linear state space models there are no general design methods. Usually, a rough estimate of physical description is required, which is not always given.

All problems with the described classical methods in the field of non-linear modelling inspire research in artificial intelligence areas. The domain of artificial intelligence consists of, along others, artificial intelligence, fuzzy systems and neuro-fuzzy systems (Duch *et al.*, 2000; Korbicz *et al.*, 1994; Osowski, 1996; Rutkowska *et al.*, 1997).

This work focuses on N-F networks and their usage in fault detection tasks. The interest in this field results from the abilities of the method for knowledge representation using fuzzy IF ... THEN .... rules. This is a characteristic property of fuzzy systems, but the N-F method is based on the neural network technique, too. It is possible to tune the parameters of fuzzy rules using algorithms known for neural networks. These properties make the N-F approach very useful for fault diagnosis, i.e. fault detection using the N-F model. The design procedure for the N-F model can be based on qualitative knowledge, i.e. in the form of physical dependencies, expert knowledge or qualitative knowledge in the form of measurements.

Fuzzy representation of knowledge makes it more legible, thus N-F models are often called *grey boxes* in contrast to neural models, which are called *black boxes* with respect to non-transparent knowledge in the form of weights. Transparent knowledge is very important in the case of fault diagnosis applications because they have to be preceded by a detailed analysis of the behavior of the diagnostic system. The task of knowledge analysis can be much easier if the knowledge is transparent.

2.6. Summary

A lot of hybrid, N-F approaches have been proposed but, the work focuses on N-F networks. The integration of neural networks and fuzzy systems in the case of N-F networks is based on the representation of the fuzzy system in the form of network topology, which resembles the neural network. Such a representation of the fuzzy system allows applying for its learning algorithms developed for neural networks. In such an approach fuzzy knowledge representation is kept. It has to be noticed that some other N-F methods do not have a transparent representation of knowledge, thus structure selection and the parameter estimation procedure are more complex and such models can be less accurate.

Additionally, the chapter gives an overview of three main types of fuzzy models: linguistic, relational and Takagi-Sugeno ones. The first and the second method are based on fuzzy rules where both the antecedent and consequent are fuzzy, thus such models are suitable to represent qualitative knowledge obtained from human experts using the natural language and linguistic terms. However, these properties of fuzzy systems make the formal, analytical analysis more difficult. Knowledge representation in the Takagi-Sugeno model is less transparent, but a lot of formal techniques can be employed to analyze it (Joh *et al.*, 1998; Matia *et al.*, 2002; Hadjili and Wertz, 2002). The work focuses on Takagi-Sugeno systems because they are more suitable for problems where knowledge about the system is given in the form of measurements, and such situations are common in the case of industrial plants. Additionally, it is assumed that consequents in fuzzy rules are crisp linear functions of N-F model inputs. This allows extending a lot of research and design methods formulated for linear models to Takagi-Sugeno N-F models.

Despite many advantages coming from the use of Takagi-Sugeno N-F networks for fault diagnostic purposes, some problems arise and they should be solved in order to permit an effective usage of Takagi-Sugeno N-F networks. For example, there is no automatic structure design for the Takagi-Sugeno N-F network, learning algorithms known for neural networks generate non-transparent knowledge, knowledge acquisition methods originating from fuzzy systems do not assure the required accuracy of modelling.

The analyzed fault detection method requires precise modelling of the behavior of the system but, as has been mentioned before, there is always a model-reality mismatch, which cannot be avoided due to disturbances or model inaccuracy. Therefore, it is very important to estimate the uncertainty of the model to consider it in the residual evaluation procedure. Model uncertainty is usually simulated in the form of thresholds defined for residuals. Of course, smaller model uncertainty generates smaller uncertainty interval for residuals, and thus faults can be detected at an early stage, which is a very important property for all fault detection systems. Nevertheless, wrong selection of thresholds for residuals so that they are not adequate for model uncertainty can disturb the work of the fault detection system, thus the system will generate a lot of false alarms. Therefore, an effective method of the estimation of N-F model uncertainty is required in order to successfully detect faults. Such a requirement is specific for fault diagnosis applications. Another important requirement states that knowledge coded in the N-F model structure should be transparent. This requires developing special identification procedures and parameter estimation algorithms for N-F techniques. The solution to these problems should allow employing N-F models for fault detection even in real industrial applications.

Chapter 3

UNCERTAINTY OF THE NEURO-FUZZY MODEL

3.1. Introduction

The reliability of diagnosis is a very important requirement for diagnostic tools. This requirement in the case of a model-based fault detection scheme is satisfied if a sufficiently precise model of the system is available. An accurate model permits a precise simulation of the system output. Thus residuals are approximately 0 for nominal conditions, and non-zero residuals indicate the faulty working mode of the system. Therefore, the fault detection system design procedure should take into account the fact that residuals should be independent of the input signals of the model. They should be sensitive to faults only. The fulfilment of these conditions permits a very fast fault detection, practically without any time delay. Unfortunately, taking into account practical conditions it must be claimed that such an ideal scenario is not possible in the case of real diagnostic applications. It results from the fact that our knowledge about the diagnosed system is not complete, thus the ideal model cannot be built, i.e. the model structure is not adequate to the system structure, the order of dynamics is not known, input variables are not determined, etc. Another problem arises from the fact that measurements used for model design and fault detection are always disturbed, thus the input and output variables cannot be treated as deterministic variables. The above-mentioned problems should be considered during model design and then the residual evaluation procedure in order to minimize the effects of disturbances and model uncertainty.

The problems raised are very important in the fault detection context because model accuracy has a fundamental influence on the reliability of the fault diagnosis system. The fact of model imperfection and the existence of disturbances cannot be ignored in order to perform fault detection tasks correctly. The correct work of the fault detection system in the case of industrial applications influences the safety of human operators and economic efficiency of the company. Thus unreliable fault detection systems, which generate false alarms or leave some faults undetected, are useless in the industrial environment. The avoidance of this problem requires ideal models of processes, but it is practically impossible to obtain such models (Chryssolouris *et al.*, 1996; Milanese and Novara, 2004; Papadopoulos *et al.*, 2001). Another approach assumes the acceptance of model imperfection and the existence of disturbances, and examines the influence of these facts on fault detection. The simulation of disturbances in a residual signal is usually performed by the interval defined around 0, which determines fault-free operations of the process. Thresholds defined for such intervals should be tuned properly to avoid false alarms and to assure fast fault detection.

The present chapter deals with the problem of N-F model imprecision and the presence of disturbances in measurements in the form of model uncertainty. Methods of computing the confidence interval for output variables of the N-F model are developed. Methods used to compute the confidence interval for linear systems are presented and then, under some assumptions, they are adapted for Takagi-Sugeno N-F networks. Statistical methods for model uncertainty determination are presented in the first section of the chapter. Unfortunately, such methods cannot be widespread due to strong assumptions concerning knowledge about the type of disturbances present in measurements. Next, the Bounded Error Approach (BEA), which is usually used to estimate the parameters of linear systems, is adapted to determine the confidence interval for the output of the N-F model. The proposed method is compared with the statistical approach. The Outer Bounding Ellipsoid (OBE) method is presented at the end of the chapter to tackle the problem of computational complexity of the BEA method.

3.2. Statistical approach for computing neuro-fuzzy network uncertainty

Given the linear model

$$y(k) = \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}, \tag{3.1}$$

where $y(k) \in \mathbb{R}$ is the model output (for the sake of simplicity only one output variable is considered), $\boldsymbol{r}(k) \in \mathbb{R}^n$ is the input vector, $\hat{\boldsymbol{p}} \in \mathbb{R}^n$ is a parameter vector. In order to estimate the parameters of the model (3.1), it is assumed that the output of the process is given in the form of the following equation:

$$y'(k) = \boldsymbol{r}^T(k)\boldsymbol{p} + \varepsilon(k), \tag{3.2}$$

where $\boldsymbol{p} \in \mathbb{R}^n$ is the vector of system parameters, $\varepsilon(k) \in \mathbb{R}$ represents disturbances. Equation (3.2) can be converted to a form which contains all measurements for $k = 1 \dots N$ in order to simplify further deliberations:

$$\boldsymbol{y}' = \boldsymbol{R}\boldsymbol{p} + \boldsymbol{\varepsilon}, \tag{3.3}$$

where $\boldsymbol{R}$ is an input matrix which consists of input vectors $\boldsymbol{r}(k)$ for $k = 1, \dots, N$. The next section shows a method that can be used to estimate the parameters of the model defined by Eqn. (3.3).

3.2.1. Least-squares method – static models

The Least-Squares (LS) method is a very popular method of parameter estimation for linear models. The method assumes that model quality can be evaluated in the form of the Sum Squares Error (SSE):

$$J = \boldsymbol{e}^T\boldsymbol{e}, \tag{3.4}$$

where

$$\boldsymbol{e} = \boldsymbol{y}' - \boldsymbol{y}. \tag{3.5}$$

The minimal value of the SSE (3.4) is obtained by computing the derivative of the function $J(\hat{\boldsymbol{p}})$, where the derivation variable is defined as a vector of the parameters $\hat{\boldsymbol{p}}$:

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}} = -2\boldsymbol{R}^T\boldsymbol{y} + 2\boldsymbol{R}^T\boldsymbol{R}\hat{\boldsymbol{p}}. \tag{3.6}$$

The result of this operation is a system of equations, which is used to determine the optimal parameters $\hat{\boldsymbol{p}}$ according to the following equation:

$$\hat{\boldsymbol{p}} = (\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T\boldsymbol{y}'. \tag{3.7}$$

Some assumptions concerning the type of disturbances must be established in order to obtain real values of the parameters $\hat{\boldsymbol{p}}$. Disturbances that corrupt measurements are represented in the form of random variables $\boldsymbol{\varepsilon} = [\varepsilon(1), \varepsilon(2), \ldots, \varepsilon(n)]^T$. The influence of the disturbances on the values of the parameters can be shown by determining the expectation for the parameter vector:

$$\begin{aligned} E[\hat{\boldsymbol{p}}] =& E[(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T\boldsymbol{y}'] = \\ =& E[(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{p} + \boldsymbol{\varepsilon})] = \\ =& \boldsymbol{p} + (\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T E[\boldsymbol{\varepsilon}]. \end{aligned} \tag{3.8}$$

This dependence clearly shows that an unbiased estimator of the real parameters $\boldsymbol{p}$ can be obtained only when the expectation of the disturbances is 0, $E[\boldsymbol{\varepsilon}] = \boldsymbol{0}$; assuming additionally that random variables used to describe disturbances are not correlated,

$$cov[\boldsymbol{\varepsilon}] = \boldsymbol{I}\sigma^2, \tag{3.9}$$

where σ^2 is the variance of each element of the vector $\boldsymbol{\varepsilon}$. For the sake of simplicity it is asssumed that each element of this vector has the same variance. The unbiased estimator of the parameters determined by the LS method is an optimum estimator and its covariance matrix takes the following form:

$$cov[\hat{\boldsymbol{p}}] = E[[\hat{\boldsymbol{p}} - E[\hat{\boldsymbol{p}}]][\hat{\boldsymbol{p}} - E[\hat{\boldsymbol{p}}]^T] = (\boldsymbol{R}^T\boldsymbol{R})^{-1}\sigma^2. \tag{3.10}$$

The estimates described by Eqn. (3.7) taking into account the above-mentioned assumptions are the best linear estimates of the parameters. From the point of view of diagnostics it is important to determine the uncertainty of such a model. Uncertainty is usually measured by the confidence interval for the output of the model. The confidence interval includes the real value of the output with defined probability. Unfortunately, in order to determine the confidence interval, more assumptions on disturbances must be established. The approach considered requires normally distributed disturbances, thus the model output

$$y(k) = \boldsymbol{r}^T(k)\hat{\boldsymbol{p}} \tag{3.11}$$

is a normally distributed random variable. The expectation for such a variable is given in the following form:

$$E[y(k)] = E[\boldsymbol{r}^T(k)\hat{\boldsymbol{p}}] = \boldsymbol{r}^T(k)E[\hat{\boldsymbol{p}}] = \boldsymbol{r}^T(k)\boldsymbol{p}, \tag{3.12}$$

and a variance

$$\begin{aligned} var[y(k)] =& E[(y(k) - E[y(k)])(y(k) - E[y(k)])^T] = \\ =& E[\boldsymbol{r}^T(k)(\hat{\boldsymbol{p}} - \boldsymbol{p})(\hat{\boldsymbol{p}} - \boldsymbol{p})^T\boldsymbol{r}(k)] = \\ =& \boldsymbol{r}^T(k)E[(\hat{\boldsymbol{p}} - \boldsymbol{p})(\hat{\boldsymbol{p}} - \boldsymbol{p})^T]\boldsymbol{r}(k). \end{aligned} \tag{3.13}$$

Taking into consideration the fact that

$$E[(\hat{\boldsymbol{p}} - \boldsymbol{p})(\hat{\boldsymbol{p}} - \boldsymbol{p})^T] = (\boldsymbol{R}^T\boldsymbol{R})^{-1}\sigma^2, \tag{3.14}$$

the final version of Eqn. (3.13) can be shown in the following form:

$$\sigma_y^2 = var[y(k)] = \boldsymbol{r}^T(k)(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{r}(k)\sigma^2. \tag{3.15}$$

The output variable $y(k)$ can be viewed as a normally distributed $N(0,1)$ random variable

$$U(k) = \frac{y(k) - E[y(k)]}{\sigma_y}. \tag{3.16}$$

Because the variance σ_y^2 in real applications is usually unknown, it can be replaced with its unbiased estimator:

$$\hat{\sigma}_y^2 = \frac{1}{N-n-1}\boldsymbol{e}^T\boldsymbol{e}. \tag{3.17}$$

The standard deviation of disturbances in Eqn. (3.16) is replaced with the random variable $\hat{\sigma}_y^2$ with chi-square distribution and $(N-n-1)$ degrees of freedom. Thus a new random variable t with t-Student distribution and $(N-n-1)$ degrees of freedom is obtained:

$$t = \frac{y(k) - \boldsymbol{r}^T(k)\boldsymbol{p}}{\sqrt{\boldsymbol{r}^T(k)(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{r}(k)\hat{\sigma}_y^2}}. \tag{3.18}$$

The confidence interval for the random variable t with the level of confidence $(1-\alpha)$ can be determined from the inequality

$$P(-t_{\alpha,N-n-1} < t_{N-n-1} < t_{\alpha,N-n-1}) = 1 - \alpha. \tag{3.19}$$

For given α, the equivalent $t_{\alpha,N-n-1}$ can be determined and the inequality (3.19) can be transformed into the form

$$-t_{\alpha,N-n-1} < \frac{y(k) - \boldsymbol{r}^T(k)\boldsymbol{p}}{\sqrt{\boldsymbol{r}^T(k)(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{r}(k)\hat{\sigma}_y^2}} < t_{\alpha,N-n-1}, \tag{3.20}$$

$$y(k) - t_{\alpha,N-n-1}\sqrt{\boldsymbol{r}^T(k)(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{r}(k)\hat{\sigma}_y^2} < \boldsymbol{r}^T(k)\boldsymbol{p} <$$
$$y(k) + t_{\alpha,N-n-1}\sqrt{\boldsymbol{r}^T(k)(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{r}(k)\hat{\sigma}_y^2}. \tag{3.21}$$

The presented method cannot be directly applied to Takagi-Sugeno N-F networks due to non-linearities present in such system. However, if some assumptions concerning the structure of the N-F model are established, a modified method can be proposed.

3.2.2. Uncertainty of the Takagi Sugeno neuro-fuzzy model – statistical approach

The output of the Takagi-Sugeno N-F model can be viewed as a combination of the outputs of partial models represented in the form of the consequents of fuzzy rules:

$$y(k) = \frac{\sum_{i=1}^{N_r}\mu_i(k)y_i(k)}{\sum_{i=1}^{N_r}\mu_i(k)}, \tag{3.22}$$

where $\mu_i(k)$ is the degree of fulfilment of the i-th rule and k is discrete time. Applying the following substitution:

$$\phi_i(k) = \frac{\mu_i(k)}{\sum_{j=1}^{N_r}\mu_j(k)}, \tag{3.23}$$

Eqn. (3.22) takes the form

$$y(k) = \phi_1(k)y_1(k) + \phi_2(k)y_2(k) + \ldots + \phi_{N_r}(k)y_{N_r}(k). \tag{3.24}$$

In order to apply the methods shown in Section 3.2, it is assumed that the system is linear in parameters (LP) and is given in the following form:

$$y'(k) = \boldsymbol{x}^T(k)\boldsymbol{p} + \varepsilon(k), \tag{3.25}$$

where $\boldsymbol{x}(k)$ is an input vector. Next, it has to be assumed that the disturbances $\varepsilon(k)$ are described by a random variable with expectation equal to 0 to allow determining unbiased LS estimators of the parameters $\boldsymbol{p}$. Additionally, it has to be assumed that the disturbances are normally distributed in order to compute the confidence interval for the model output. Such assumptions allow computing the confidence interval for the following model:

$$y(k) = \boldsymbol{x}^T(k)\hat{\boldsymbol{p}}. \tag{3.26}$$

The Takagi-Sugeno N-F model is non-linear in parameters (NLP), so it must be converted to an LP model under some assumptions. Let the functions $\phi_i(k)$ defined for fuzzy rules be given. Partial models represented by consequents have input variables represented by the following vectors: $\boldsymbol{r}_1(k), \boldsymbol{r}_2(k), \ldots, \boldsymbol{r}_{N_r}(k)$, so Eqn. (3.24) can be rewritten in the form

$$y(k) = \phi_1(k)\boldsymbol{r}_1^T(k)\hat{\boldsymbol{p}}_1 + \phi_2(k)\boldsymbol{r}_2^T(k)\hat{\boldsymbol{p}}_2 + \ldots + \phi_{N_r}(k)\boldsymbol{r}_{N_r}^T(k)\hat{\boldsymbol{p}}_{N_r}. \tag{3.27}$$

Substituting

$$\boldsymbol{x}_i(k) = \phi_i(k)\boldsymbol{r}_i(k), \tag{3.28}$$

the following expression is achieved:

$$y(k) = \boldsymbol{x}_1^T(k)\hat{\boldsymbol{p}}_1 + \boldsymbol{x}_2^T(k)\hat{\boldsymbol{p}}_2 + \ldots + \boldsymbol{x}_{N_r}^T(k)\hat{\boldsymbol{p}}_{N_r}, \tag{3.29}$$

which can be transformed into the following form:

$$y(k) = \boldsymbol{x}^T(k)\hat{\boldsymbol{p}}, \tag{3.30}$$

where

$$\boldsymbol{x}(k) = \begin{bmatrix} \boldsymbol{x}_1(k) \\ \boldsymbol{x}_2(k) \\ \vdots \\ \boldsymbol{x}_{N_r}(k) \end{bmatrix}, \hat{\boldsymbol{p}} = \begin{bmatrix} \hat{\boldsymbol{p}}_1 \\ \hat{\boldsymbol{p}}_2 \\ \vdots \\ \hat{\boldsymbol{p}}_{N_r} \end{bmatrix}.$$

The vectors $\boldsymbol{r}_i(k) \in \mathbb{R}^n$ describe the inputs of partial models and the vector $\boldsymbol{x}(k) \in \mathbb{R}^{nN_r}$ describes the same inputs but in the form of a single vector. It has to be mentioned that the columns of the matrix $\boldsymbol{X}$ cannot be linearly dependent because otherwise the estimates of the parameters using the LS algorithm cannot be computed. The confidence interval for such a modified Takagi-Sugeno model can be evaluated using Eqn. (3.21), substituting the input vector $\boldsymbol{r}(k)$ with the modified input vector $\boldsymbol{x}(k)$ and replacing the matrix $\boldsymbol{r}(k)$ with the matrix $\boldsymbol{X}$.

The effectiveness of the presented method in fault diagnosis applications is strongly limited by detailed assumptions established for disturbances. In practical problems disturbances are usually not normally distributed and knowledge about their distribution is not available. Thus the method can be used for a narrow class of objects only. The use of the method with incompletely fulfilled assumptions leads to wrong results, which are useless for fault diagnosis systems. The problem is illustrated by an example. The identification of the LP static system using the LS method is shown and next the confidence interval for the output of the model is calculated.

Example 3.1

We are given a system described by the following equation:

$$y'(k) = e^{-\frac{(u(k)-0.5)^2}{2}}p_1u(k) + e^{-\frac{(u(k)+0.5)^2}{2}}p_2u(k) + \varepsilon(k), \tag{3.31}$$

where $\boldsymbol{p} = [2 \quad 1]^T$ and $\varepsilon \in U(-0.3, 0.7)$. The input signal contains 200 samples randomly generated from the range $\boldsymbol{X}$ and 200 test samples generated using the

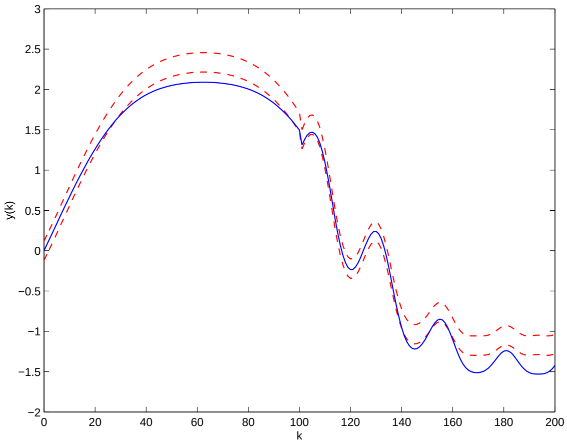

Fig. 3.1. Ideal system response (–) and the confidence interval for the output of the system (- -)

following signal:

$$u(k) = \begin{cases} sin(2\pi(k/250)) & \text{for } k = 1, \ldots, 100 \\ 0.8sin(2\pi(k/250)) + 0.2sin(2\pi(k/25)) & \text{for } k = 101, \ldots, 200. \end{cases}$$

The parameters of the model were estimated using the LS method. The following estimates were obtained: $\hat{\boldsymbol{p}} = [2.46 \quad 0.63]^T$. The confidence interval for the output signal was computed using Eqn. $\hat{\boldsymbol{p}} = [2.46 \quad 0.63]^T$ and assuming the confidence level $\alpha = 0.01$. The results are shown in Fig. 3.1, where the ideal output of the model $y(k)$ and the confidence interval are presented. The results show that the confidence interval does not include the ideal response of the model because the assumptions concerning disturbances are not satisfied. This example proves that the LS method and the statistical approach cannot be used for computing the confidence interval if the required assumptions are not satisfied. Such limitations concern also N-F models, thus it is necessary to develop an alternative method of estimating the parameters and a new method of computing the confidence interval. The new methods should not require a detailed knowledge about disturbances.

3.2.3. Dynamic neuro-fuzzy model and uncertainty

Previous discussions concerning model uncertainty did not take into account the fact that real systems are usually dynamic. This fact introduces a lot of modifications to the problems shown in the previous sections. Dynamics are introduced into the N-F model by applying partial linear dynamic models described by difference equations:

$$\begin{aligned} y(k) = &- a_1 y(k-1) - a_2 y(k-2) - \ldots - a_{n_a} y(k-n_a) + \\ &+ \boldsymbol{b}_0^T \boldsymbol{u}(k) + \boldsymbol{b}_1^T \boldsymbol{u}(k-1) + \ldots + + \boldsymbol{b}_{n_b}^T \boldsymbol{u}(k-n_b). \end{aligned} \tag{3.32}$$

Analogously to (3.1), the equation that describes the model is simplified and the input vector and the vector of parameters are given in the following form:

$$\boldsymbol{r}(k) = [-y(k-1), \ldots, -y(k-n_a), \boldsymbol{u}(k), \boldsymbol{u}(k-1), \ldots, \boldsymbol{u}(k-n_b)], \tag{3.33}$$

$$\hat{\boldsymbol{p}} = [a_1, \ldots, a_{n_a}, \boldsymbol{b}_0, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n_b}]. \tag{3.34}$$

Like in the static case, the estimates of parameters can be computed using Eqn. (3.7). The output values of the system must be treated like random variables due to random disturbances that corrupt the output signal. The matrix that stores the input vectors,

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}^T(1) \\ \boldsymbol{r}^T(2) \\ \vdots \\ \boldsymbol{r}^T(N) \end{bmatrix},$$

is correlated through the output signal $y'(k)$ with all previous disturbances $\varepsilon(k)$ according to (3.32). Although the assumption $E[\boldsymbol{\varepsilon}] = 0$ is satisfied, $\boldsymbol{R}$ and $\boldsymbol{\varepsilon}$ are correlated and in the general case

$$E[(\boldsymbol{R}^T\boldsymbol{R})^{-1}\boldsymbol{R}^T\boldsymbol{\varepsilon}] \neq \boldsymbol{0}, \tag{3.35}$$

thus

$$E[\hat{\boldsymbol{p}}] \neq \boldsymbol{p}. \tag{3.36}$$

Model uncertainty can be evaluated if unbiased estimates of parameters are available. In the case considered, such estimates cannot be determined, thus the analysed methods of computing model uncertainty are inappropriate for dynamic models. There exist some methods that allow determining unbiased estimates for dynamic systems $\boldsymbol{\varepsilon}$, but knowledge about the correlation between $\boldsymbol{X}$ and $\boldsymbol{\varepsilon}$ is required, so the usage of this method is strongly limited.

An alternative approach for computing the parameters of dynamic models is the instrumental variable (IV) method (Eykhoff, 1980; Nahorski and Mańczak, 1983; Söderström and Stoica, 1997; Ljung, 1987). The method requires extra measurements in the form of instrumental variable matrices $\boldsymbol{Z}$, which have the same size as the matrix $\boldsymbol{R}$. Data in the matrix $\boldsymbol{Z}$ must be asymptotically correlated with data in the matrix $\boldsymbol{R}$ and asymptotically uncorrelated with disturbances:

$$P \lim_{N\to\infty} \left(\frac{1}{N}\boldsymbol{Z}^T\boldsymbol{\varepsilon}\right) = \boldsymbol{0}. \tag{3.37}$$

Moreover, a non-singular matrix must exist:

$$P \lim_{N\to\infty} \left(\frac{1}{N}\boldsymbol{Z}^T\boldsymbol{R}\right) = \boldsymbol{R}_{zx}. \tag{3.38}$$

The estimates of parameters have the following form:

$$\hat{\boldsymbol{p}} = (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T\boldsymbol{y}'. \tag{3.39}$$

Additionally, the estimates are consistent and so asymptotically unbiased. This can be proved in the following way:

$$\begin{aligned}\hat{\boldsymbol{p}} &= (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T\boldsymbol{y}' = \\ &= (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T(\boldsymbol{R}\boldsymbol{p}+\boldsymbol{\varepsilon}) = \\ &= (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T\boldsymbol{R}\boldsymbol{p} + (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T\boldsymbol{\varepsilon} = \\ &= \boldsymbol{p} + (\boldsymbol{Z}^T\boldsymbol{R})^{-1}\boldsymbol{Z}^T\boldsymbol{\varepsilon} = \\ &= \boldsymbol{p} + \left(\frac{1}{N}\boldsymbol{Z}^T\boldsymbol{R}\right)^{-1}\left(\frac{1}{N}\boldsymbol{Z}^T\boldsymbol{\varepsilon}\right).\end{aligned} \tag{3.40}$$

Taking into consideration (3.37) and (3.38) it can finally be shown that the estimates are consistent:

$$P\lim_{N\to\infty}\hat{\boldsymbol{p}} = \boldsymbol{p} + \boldsymbol{R}_{zx}^{-1}\boldsymbol{0} = \boldsymbol{p}. \tag{3.41}$$

The unbiased estimates of parameters let us determine the confidence interval analogously to the approaches raised in the previous sections. Nevertheless, it must be noted that success in this case strongly depends on a suitable selection of the auxiliary matrix $\boldsymbol{Z}$ (Söderström and Stoica, 1997; Nahorski and Mańczak, 1983). Another problem arises from the fact that the IV method generates significantly biased estimates if not enough measurements are used for calculations.

It is clear now that the dynamic Takagi-Sugeno N-F network may cause complications in the estimation procedure and problems with the calculation of the confidence interval if the above methods will be used for this purpose. Therefore, the rest of this work focuses on alternative methods, which are able to determine the confidence interval without a detailed knowledge about disturbances.

3.3. Bounded-error method

As was shown in the example 3.1, the LS method and the statistical approach for calculating the confidence interval cannot be used for this purpose if disturbances are not normally distributed or the expectation of disturbances is not 0. This work deals with the use of N-F models in fault detection in real industrial applications, so there is a high probability that the required assumptions may not be satisfied. Therefore, the presented methods have only a theoretical sense and real applications require alternative solutions. The BEA method does not assume what kind of disturbances corrupt measurements, and it can be applied to parameter estimation and confidence interval determination (Walter and Pronzato, 1997; Milanese *et al.*, 1996; Walter and Piet-Lahanier, 1990). The method requires the maximum values of disturbances to be given; however, there exist some methods that can estimate even these values (Piet-Lahanier and Walter, 1994; Maksarov and Norton, 1996).

Let us consider the system

$$y'(k) = \boldsymbol{r}^T(k)\boldsymbol{p} + \varepsilon(k), \tag{3.42}$$

where the errors $\varepsilon(k)$ are bounded by the following conditions:

$$\varepsilon^{min}(k) \leqslant \varepsilon(k) \leqslant \varepsilon^{max}(k). \tag{3.43}$$

The lower bound $\varepsilon^{min}(k)$ and the upper bound $\varepsilon^{max}(k)$ are known *apriori* (i.e. from the technical specification of the plant) and $\varepsilon^{min}(k) \neq \varepsilon^{max}(k)$. The main aim of the method is to determine the area in the parameter space which is compatible with all accessible measurements and with the defined bounds. The disturbances can be viewed in the following form:

$$\varepsilon(k) = y'(k) - \boldsymbol{r}^T(k)\boldsymbol{p}. \tag{3.44}$$

The searched area can be represented by the following set of parameters, which is consistent with the measurements and bounds:

$$\mathbb{P} = \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \varepsilon^{max}(k) \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant y'(k) - \varepsilon^{min}(k), k = 1, \ldots, N\}. \tag{3.45}$$

The set $\mathbb{P}$ is created by the intersection of N sets

$$\mathbb{P} = \bigcap_{k=1}^{N} \mathbb{U}(k), \tag{3.46}$$

where each set $\mathbb{U}(k)$ is formed by two parallel hyperplanes:

$$\begin{aligned} \mathbb{H}^+ &= \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \boldsymbol{r}^T(k)\boldsymbol{p} = \varepsilon^{max}(k)\}, \\ \mathbb{H}^- &= \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \boldsymbol{r}^T(k)\boldsymbol{p} = \varepsilon^{min}(k)\}. \end{aligned} \tag{3.47}$$

Each point inside the set $\mathbb{P}$ defines the vector of model parameters and all sets of parameters determine the group of models consistent with the measurements and bounds. This means that instead of one model, a set of models with different parameters is given and the output signal is represented in the form of an interval which contains all possible model responses. Real applications usually require a single output value, thus one set of parameters must be chosen. The most common approach chooses the geometrical center of the area $\mathbb{P}$ as the set of parameters that is used to calculate the output of the model. This sample procedure is shown in Fig. 3.2. If the maximum and minimum values of the parameters are known,

$$p_i^{min} = \arg \min_{p \in \mathbb{P}} p_i, \tag{3.48}$$

$$p_i^{max} = \arg \max_{p \in \mathbb{P}} p_i, \tag{3.49}$$

the estimates of the parameters can be computed using the following formula:

$$p_i = \frac{p_i^{min} + p_i^{max}}{2}, \quad i = 1, \ldots, N. \tag{3.50}$$

The minimum and maximum values for the following parameters are determined using the linear programming technique (Milanese *et al.*, 1996). The computation

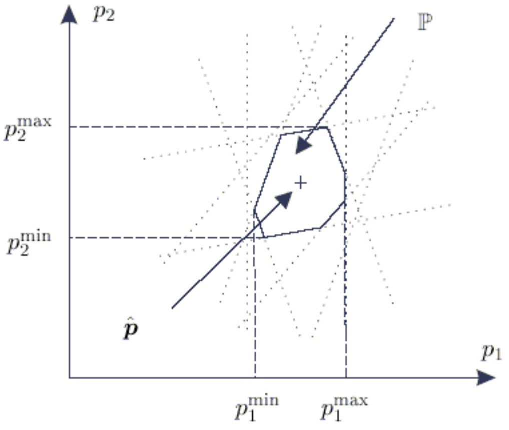

Fig. 3.2 . Sample set of parameters $\mathbb{P}$

cost for this procedure can be high and can grow very fast if the number of parameters n increases or the number of measurements N increases. The relationship between the number of faces that bounds the set $\mathbb{P}$ and the number of measurements, and the number of vertices in the polytope that bounds $\mathbb{P}$ is given by the following formulae:

$$w(n_s, N) = \begin{cases} \sum_{j=1}^{m} \frac{n_s}{n_s - j} \binom{n_s - j}{j} \binom{j}{j-N} & \text{for} \quad N = 2m \\ \sum_{j=1}^{m} \frac{N+1}{n_s - j} \binom{n_s - j}{j} \binom{j}{j-N} & \text{for} \quad N = 2m + 1 \end{cases} .$$

In order to compare the BEA and the LS method, a sample problem is examined.

Example 3.2

The identification of the LP static system is made using the BEA and the LS algorithm. The system is described as follows:

$$y'(k) = p_1 \sin(r_2(k)) + p_2 r(k) + \varepsilon(k), \tag{3.51}$$

where $\boldsymbol{p} = [1.7 \quad 0.6]^T$, $\varepsilon(k) = \mathbb{U}(-0.3, 0.7)$. The estimation procedure is based on 200 samples, which were generated using a random input signal. The BEA method also requires lower and upper bounds, thus we must be given $\varepsilon^{max} = 0.8$ and $\varepsilon^{min} = -0.4$. Both methods determine the values of the parameters and compute confidence regions for the parameters. In the case of the LS method, the F-test was employed to calculate the confidence region for the parameters (Mańczak, 1976; Rafajłowicz, 1996):

$$(\hat{\boldsymbol{p}} - \boldsymbol{p})^T \boldsymbol{R}^T \boldsymbol{R} (\hat{\boldsymbol{p}} - \boldsymbol{p}) \leqslant (n+1) \hat{\sigma}_y^2 F_{\alpha, N-n-1}^{n+1}, \tag{3.52}$$

where $F^{n+1}_{\alpha,N-n-1}$ is a quintile of order $(1-\alpha)$ for a random variable with the F-Snedecor distribution, and $(N-n-1)$ and $(n+1)$ are orders of freedom, $\hat{\sigma}_y^2$ is the estimate of variance described by the formulae (3.17). The confidence interval determined by the statistical approach is calculated with the level of confidence $(1-\alpha)=0.99$. The results of the identification procedure are shown in Fig. 3.3. The LS method gave the following values of parameters: $\hat{\boldsymbol{p}}_{LS}=[1.76\ 0.72]$, and the BEA method $\hat{\boldsymbol{p}}_{BEA}=[1.71\ 0.6]$. The results confirm the theoretical assumptions stating that the BEA method is able to determine parameters more precisely than the LS method if the expectation of disturbances is other than 0. Moreover, if disturbances are not normally distributed, the confidence area determined by the BEA is more adequate than the confidence set computed using the statistical approach. The example 3.2 shows that the statistical approach can determine the confidence set that does not include real parameters if the assumptions about disturbances are not satisfied. The BEA method guarantees that the admissible set of parameters contains real parameters in opposition to statistical approach, which guarantees that the admissible set of parameters contains real parameters but with the chosen confidence level. It is possible to define the high confidence level for the statistical method, but it results in a huge admissible set of parameters. The example confirms the effectiveness of the BEA method in determining the admissible set of parameters under non-strict assumptions concerning disturbances.

It seems that the BEA method can be successfully adapted for tuning the parameters of the Takagi-Sugeno N-F network and, moreover, the results of this process can be applied to determine the confidence interval for the network output. This fact is important in the context of fault detection applications using Takagi-Sugeno N-F models because the admissible set of parameters can be used to calculate the adaptive threshold of the residual signal. Such an approach makes robust fault detection under model uncertainty possible and lets the fault be detected at an early stage.

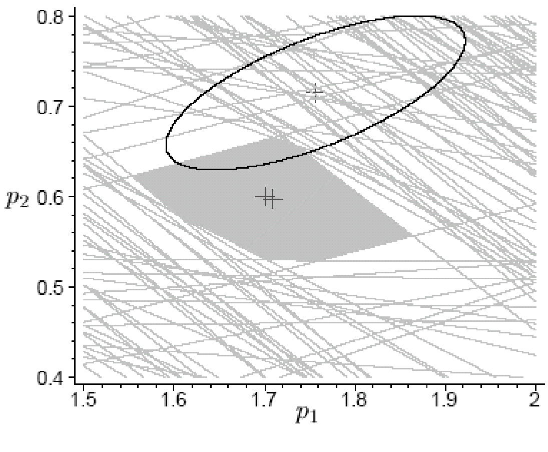

Fig. 3.3. Parameters and confidence regions determined by the LS method and the BEA method

3.3.1. Confidence interval for the model output

The admissible set of parameters $\mathbb{P}$ can be used to determine the confidence interval of the output signal of the model. Let $\mathbb{W}$ be the set of all vertices $\boldsymbol{p}_w^i$, $i = 1, \ldots, n_w$ for a polytope that bounds the set $\mathbb{P}$. The confidence interval for the model (3.1) is given by the following inequalities:

$$\boldsymbol{r}^T(k)\boldsymbol{p}^{min}(k) \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant \boldsymbol{r}^T(k)\boldsymbol{p}^{max}(k), \tag{3.53}$$

where

$$\boldsymbol{p}^{min}(k) = \arg\min_{\boldsymbol{p}\in\mathbb{W}} \boldsymbol{r}^T(k)\boldsymbol{p}, \tag{3.54}$$

$$\boldsymbol{p}^{max}(k) = \arg\max_{\boldsymbol{p}\in\mathbb{W}} \boldsymbol{r}^T(k)\boldsymbol{p}. \tag{3.55}$$

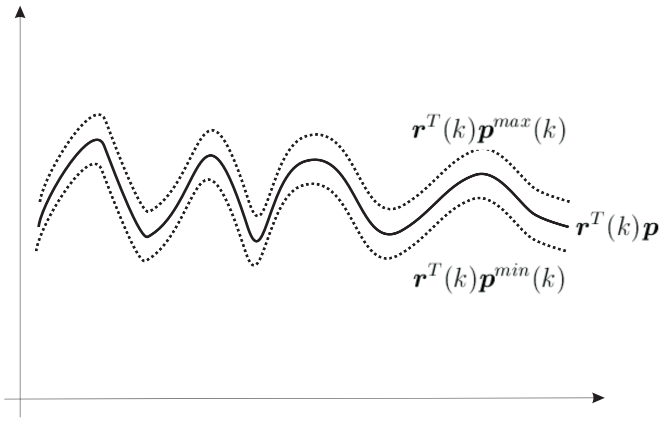
$\boldsymbol{r}^T(k)\boldsymbol{p}^{max}(k)$
$\boldsymbol{r}^T(k)\boldsymbol{p}$
$\boldsymbol{r}^T(k)\boldsymbol{p}^{min}(k)$

Fig. 3.4 . Confidence interval for the output of the model

If the output signal of the model is bounded by (3.54), then the output of the corresponding system is bounded by the confidence interval described by means of the following expression:

$$\boldsymbol{r}^T(k)\boldsymbol{p}^{min}(k) + \varepsilon^{min}(k) \leqslant y'(k) \leqslant \boldsymbol{r}^T(k)\boldsymbol{p}^{max}(k) + \varepsilon^{max}(k). \tag{3.56}$$

The fact that the output values of the system are bounded by the confidence interval is useful for fault detection. The procedure of fault detection using the adaptive threshold is shown in detail in the chapter concerning fault detection using the N-F model.

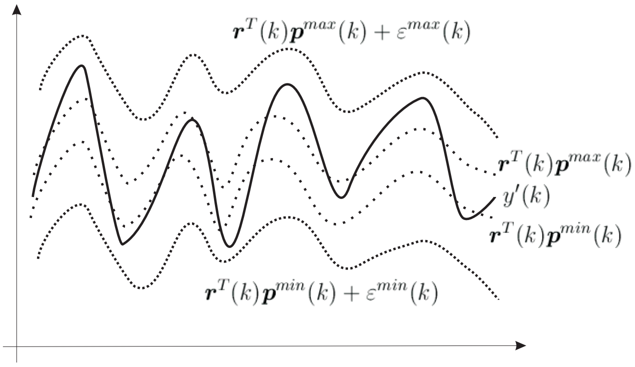

Fig. 3.5 . Confidence interval for the output of the system

3.3.2. Models with uncertain input variables

The application of diagnostic systems to real systems, i.e. industrial plants, must take into account the fact that practically all real objects must be treated as dynamic systems. For effective fault detection, dynamics must be simulated in models used to generate residuals. In the case of Takagi-Sugeno N-F networks, dynamics can be simulated by including dynamics for partial linear models that represent the consequents of fuzzy rules. Thus input vectors for partial models contain delayed values of input variables and delayed values of the output variable. A single partial model is described by Eqn. (3.32). If input vectors for partial models do not include delayed outputs, the previous deliberations concerning BEA estimation are correct for such a situation. However, if this assumption is not fulfilled, the BEA method must be modified due to the fact that the output variable is uncertain and its value is used as the input, which increases global uncertainty of the model. This additional uncertainty must be considered during computing the confidence interval for the output signal of the model to avoid the model-reality mismatch.

The real unknown input vector can be seen as the difference between the known values of inputs and their errors:

$$\boldsymbol{r}'(k) = \boldsymbol{r}(k) - \boldsymbol{e}(k). \tag{3.57}$$

Let us assume additionally that the error $\boldsymbol{e}(k)$ is bounded:

$$e_i^{min}(k) \leqslant e_i(k) \leqslant e_i^{max}(k), \quad i = 1, \ldots, N, \tag{3.58}$$

therefore the admissible set of parameters $\mathbb{P}$ is given by the following inequalities:

$$\begin{aligned}\mathbb{P} =& \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \varepsilon^{max}(k) + \boldsymbol{e}^T(k)\boldsymbol{p} \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant \\ & \leqslant y'(k) - \varepsilon^{min}(k) + \boldsymbol{e}^T(k)\boldsymbol{p},\, k = 1, \ldots, N\}.\end{aligned} \tag{3.59}$$

The constraints that determine the admissible set of parameters depend upon the unknown vector of parameters $\mathbb{P}$, which makes it difficult to determine the estimates of these parameters. Nevertheless, from the practical point of view, the procedure for calculating the estimates requires only information about the sign of the expression $\boldsymbol{e}^T(k)\boldsymbol{p}$. For this purpose each parameter is viewed in the form of the difference of two positive parameters:

$$p_i = p_i' - p_i'', \quad p_i', p_i'' \geqslant 0. \tag{3.60}$$

Such a modification of the task lets us replace the expression $\boldsymbol{e}^T(k)\boldsymbol{p}$ with an expression that satisfies the following constraints:

$$\boldsymbol{e}^T(k)\boldsymbol{p} \leqslant (\boldsymbol{e}^{max}(k))^T\boldsymbol{p}' - (\boldsymbol{e}^{min}(k))^T\boldsymbol{p}''. \tag{3.61}$$

From this modification there arises a new admissible set of parameters $\mathbb{P}$:

$$\begin{aligned}\mathbb{P} =\{&\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \varepsilon^{max}(k) - (\boldsymbol{e}^{max}(k))^T\boldsymbol{p}' + (\boldsymbol{e}^{min}(k))^T\boldsymbol{p}'' \leqslant \\ &\leqslant \boldsymbol{r}^T(k)(\boldsymbol{p}' - \boldsymbol{p}'') \leqslant \\ &\leqslant y'(k) - \varepsilon^{min}(k) - (\boldsymbol{e}^{max}(k))^T\boldsymbol{p}' + (\boldsymbol{e}^{min}(k))^T\boldsymbol{p}'',\, k = 1, \ldots, N\}.\end{aligned} \tag{3.62}$$

For such an admissible set of parameters the linear programming technique can be employed analogously to the approach shown in Section 3.3. The difference is revealed only in constraints defined by measurements, which are not paralel hyperplanes now and each hyperplane must be considered separately.

The admissible set of parameters $\mathbb{P}$ expressed by Eqn. (3.62) allows determining the confidence interval for the output signal of the model in the form of the following inequalities:

$$\begin{aligned}&[\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]^T\boldsymbol{p}'^{\,min}(k) - [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]\boldsymbol{p}''^{\,min}(k) \leqslant (\boldsymbol{r}'(k))^T\boldsymbol{p} \leqslant \\ &\leqslant [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]^T\boldsymbol{p}'^{\,max}(k) - [\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]\boldsymbol{p}''^{\,max}(k),\end{aligned} \tag{3.63}$$

where

$$\begin{aligned}(\boldsymbol{p}'^{\,min}(k), \boldsymbol{p}''^{\,min}(k)) = \arg \min_{(\boldsymbol{p}', \boldsymbol{p}'') \in \mathbb{W}} \big([\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]^T\boldsymbol{p}' + \\ - [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]\boldsymbol{p}''\big),\end{aligned} \tag{3.64}$$

$$\begin{aligned}(\boldsymbol{p}'^{\,max}(k), \boldsymbol{p}''^{\,max}(k)) = \arg \max_{(\boldsymbol{p}', \boldsymbol{p}'') \in \mathbb{W}} \big([\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]^T\boldsymbol{p}' + \\ - [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]\boldsymbol{p}''\big).\end{aligned} \tag{3.65}$$

The confidence interval for the output of the simulated system can be computed easily if the confidence interval for the output of the model is known:

$$\begin{aligned}&[\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]^T\boldsymbol{p}'^{\,min}(k) - [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]\boldsymbol{p}''^{\,min}(k) + \varepsilon^{min}(k) \leqslant y'(k) \leqslant \\ &\leqslant [\boldsymbol{r}(k) - \boldsymbol{e}^{min}(k)]^T\boldsymbol{p}'^{\,max}(k) - [\boldsymbol{r}(k) - \boldsymbol{e}^{max}(k)]\boldsymbol{p}''^{\,max}(k) + \varepsilon^{max}(k).\end{aligned} \tag{3.66}$$

3.3.3. Bounded-error approach for dynamic systems

A simple example is shown to explain the approach used to compute the estimates of the parameters and the uncertainty of the dynamic model. We are given the dynamic model

$$\begin{aligned} y(k) = & - a_1 y(k-1) - a_2 y(k-2) - \ldots - a_{n_a} y(k-n_a) + \\ & + b_0 u(k) + b_1 u(k-1) + \ldots + + b_{n_b} u(k-n_b), \end{aligned} \tag{3.67}$$

where the input values $u(k)$ are known roughly and are bounded by the following inequalities:

$$\varepsilon_u^{min}(k) \leqslant u_\varepsilon(k) - u(k) \leqslant \varepsilon_u^{max}(k), \tag{3.68}$$

where $u_\varepsilon(k)$ is the known input value corrupted by a disturbance. The output error is also bounded and the bounds are given in the form of the following inequalities:

$$\varepsilon_y^{min}(k) \leqslant y'(k) - y(k) \leqslant \varepsilon_y^{max}(k), \tag{3.69}$$

where $y'(k)$ represents the measured value of the output signal. The analyzed model is described by the expression (3.70) in order to determine the admissible set of parameters $\mathbb{P}$ for N given measurements in the form of the input and output values:

$$y(k) = \boldsymbol{r}^T(k)\boldsymbol{p}, \tag{3.70}$$

where parameters can be viewed as the vector

$$\boldsymbol{p} = [a_1, \ldots, a_{n_a}, b_0, b_1, \ldots, b_{n_b}]. \tag{3.71}$$

The unknown vector of the inputs,

$$\boldsymbol{r}(k) = [-y(k-1), \ldots, -y(k-n_a), u(k), u(k-1), \ldots, u(k-n_b)], \tag{3.72}$$

is approximated by the vector of measured input values,

$$\boldsymbol{r}'(k) = [-y'(k-1), \ldots, -y'(k-n_a), u_\varepsilon(k), u_\varepsilon(k-1), \ldots, u_\varepsilon(k-n_b)]. \tag{3.73}$$

For the sake of simplicity the constraints put on the input and output variables are simplified to the following form:

$$|u_\varepsilon(k) - u(k)| \leqslant \varepsilon_u, \tag{3.74}$$

$$|y'(k) - y(k)| \leqslant \varepsilon_y. \tag{3.75}$$

For such modified bounds the admissible set of parameters $\boldsymbol{p}$ has the following form:

$$\begin{aligned} \mathbb{P} = & \{\boldsymbol{p'}, \boldsymbol{p''} \in \mathbb{R}^n \mid -\varepsilon_y - \sum_{i=1}^{n_a} \varepsilon_y (p_i' + p_i'') - \sum_{i=n_a+1}^{n_a+n_b} \varepsilon_u (p_i' + p_i'') \leqslant \\ & \leqslant y'(k) - \boldsymbol{r}'(k)^T (\boldsymbol{p}' - \boldsymbol{p}'') \leqslant \\ & \leqslant \varepsilon_y + \sum_{i=1}^{n_a} \varepsilon_y (p_i' + p_i'') + \sum_{i=n_a+1}^{n_a+n_b} \varepsilon_u (p_i' + p_i''),\ k = 1, \ldots, N\}, \end{aligned} \tag{3.76}$$

where particular parameters are converted according to Eqn. (3.60).

3.3.4. Bounded-error approach for neuro-fuzzy networks

The application of the BEA algorithm for estimating N-F network parameters and computing the confidence interval for the output requires establishing some assumptions concerning the parameters and input variables of the N-F network. The assumptions mentioned are identical with these established for the LS method presented in Section 3.2.2. The idea behind this approach is the conversion of the Takagi-Sugeno N-F network to the LP system; therefore all assumptions concerning the BEA method discussed in previous sections are suitable also for Takagi-Sugeno systems. The main assumption required to view the Takagi-System as an LP system is based on the idea that the parameters of membership functions of fuzzy sets represented by the elements of the 1st layer are known. An appropriate selection of the values of these parameters has an essential influence on the uncertainty of the whole N-F model. Wrong values of these parameters can significantly increase model uncertainty, thus the model can be unsuitable for diagnostic tasks. The problem of tuning the discussed parameters is the main subject of the next chapter, where details of the proposed methods are presented. The BEA method considered in this part of the work lets us determine the parameters of linear consequents of fuzzy rules on the basis of measurements. The parameters of linear consequents are included in the structure of the Takagi-Sugeno N-F network as the elements of the 5th layer. The BEA method allows determining these parameters for both static and dynamic models, which are employed to represent linear consequents of fuzzy rules. Moreover, it is possible to determine the confidence interval for the output of the N-F model using the admissible set of parameters $\mathbb{P}$ determined earlier. This is very important in the context of fault detection because knowledge about the level of the confidence interval of the output of the model can be directly used to calculate the adaptive threshold of the residual signal. The BEA method in opposition to the LS method and the statistical approach does not require detailed assumptions concerning disturbances for computing the admissible set of parameters. Therefore, the adaptation of the BEA method for tuning parameters and determining the uncertainty of N-F models makes it possible to use it for fault detection in real systems, which in the case of LS methods and statistical approaches is practically impossible due to strong assumptions concerning disturbances, which in reality are rarely satisfied. In the case of using the BEA method for tuning the parameters of N-F networks, problems with computational complexity of BEA algorithms can appear due to high computational costs of determining the polytope that bounds the admissible set of parameters. It is especially important when the model consists of many partial models with numerous input variables, because the time and memory space required for such computations can exceed the acceptable bounds. Unfortunately, practical tests have shown that the usage of the BEA method for N-F models with more than 5 parameters is practically impossible. The problem can be overcome by using a simplified BEA method presented in the next section. This method is much less complex but, unfortunately, also less precise in determining the admissible set of parameters and the confidence interval. (Bai and Huang, 1999; Walter and Pronzato, 1997).

3.4. Outer bounding ellipsoid method

Some simplified methods of computing the admissible set of parameters have been proposed in order to overcome the problem concerning the classical BEA algorithm mentioned in the previous section. The idea of the new methods is to approximate the actual set $\mathbb{P}$ by an area which has a simplified shape (Broman and Shensa, 1990; Dabbenea *et al.*, 2003; Walter and Pronzato, 1997). The main assumption concerning the proposed methods is related to enclosing the actual set $\mathbb{P}$ in the new bounded area. The most popular method approximates the actual admissible set of parameters $\mathbb{P}$ by the outer bounded ellipsoid (OBE)(Fig. 3.6):

$$\mathbb{E} = \{\boldsymbol{p} \in \mathbb{R}^n |\, (\boldsymbol{p} - \hat{\boldsymbol{p}})^T \boldsymbol{M}^{-1} (\boldsymbol{p} - \hat{\boldsymbol{p}}) \leqslant \sigma^2\}, \tag{3.77}$$

where $\hat{\boldsymbol{p}}$ is the vector that defines the center of the ellipsoid, $\boldsymbol{M}$ is a positively defined matrix which defines the orientation of the ellipsoid in the space, and the coefficient σ has an impact on the size of the ellipsoid. The BEA algorithm initially requires a sufficiently big ellipsoid in order to enclose the real set $\mathbb{P}$:

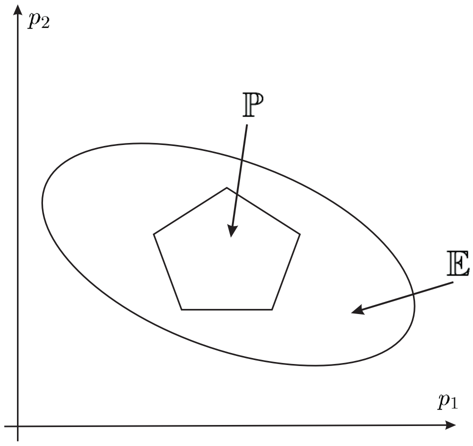

Fig. 3.6 . Ellipsoid approximates the real set $\mathbb{P}$

$$\mathbb{E}(0) = \{\boldsymbol{p} \in \mathbb{R}^n |\, (\boldsymbol{p} - \hat{\boldsymbol{p}}(0))^T \boldsymbol{M}^{-1}(0) (\boldsymbol{p} - \hat{\boldsymbol{p}}(0)) \leqslant \sigma^2(0)\}, \tag{3.78}$$

where the following assumptions are made: $\boldsymbol{M}^{-1}(0)\sigma^2(0) = c\boldsymbol{I}_N$, with the value of the coefficient c big enough, and the center of the ellipsoid defined in the following form: $\hat{\boldsymbol{p}}(0) = \boldsymbol{0}$. For the sake of simplicity, let us assume that the error $\varepsilon(k)$ is bounded in the following form:

$$|\,\varepsilon(k)| \leqslant \varepsilon_g(k). \tag{3.79}$$

Therefore, the admissible set of parameters $\mathbb{P}$ is described by the following inequalities:

$$\mathbb{P} = \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(k) - \varepsilon_g(k) \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant y'(k) + \varepsilon_g(k), k = 1, \ldots, N\}. \tag{3.80}$$

The main idea of the OBE method is to determine iteratively in every step a smaller and smaller ellipsoid $\mathbb{E}(k)$, which is a result of the intersection of the area $\mathbb{U}(k)$ (Fig. 3.7) bounded by the two parallel hyperplanes $\mathbb{H}^{+}(k)$ and $\mathbb{H}^{-}(k)$ with the ellipsoid $\mathbb{E}(k-1)$ calculated in the previous step of the algorithm:

$$\mathbb{E}(k-1) \cap \mathbb{U}(k) \subset \mathbb{E}(k). \tag{3.81}$$

The result of the described intersection is an area that encloses the real set $\mathbb{P}$, and this area can be approximated by the set of the following ellipsoids:

$$\begin{aligned}\mathbb{E}(k) =&\{\boldsymbol{p} \in \mathbb{R}^n |\, \eta_1(k)(\boldsymbol{p} - \hat{\boldsymbol{p}}(k-1))^T \boldsymbol{M}^{-1}(k-1)(\boldsymbol{p} - \hat{\boldsymbol{p}}(k-1))+\\ &+ \eta_2(k)(y'(k) - \boldsymbol{r}^T(k)\boldsymbol{p})^2 \leqslant \eta_1(k)\sigma^2(k-1) + \eta_2(k)\varepsilon_g^2(k)\},\end{aligned} \tag{3.82}$$

where $\eta_1(k)$, $\eta_2(k) \in [0,1]$. Each ellipsoid that fulfills the presented constraints encloses the area $\mathbb{E}(k-1) \cap \mathbb{U}(k)$; however, the algorithm needs to determine only one ellipsoid that minimizes the size of the admissible set of parameters. Three criterions are commonly used in order to find the optimal ellipsoid (in a geometrical sense):

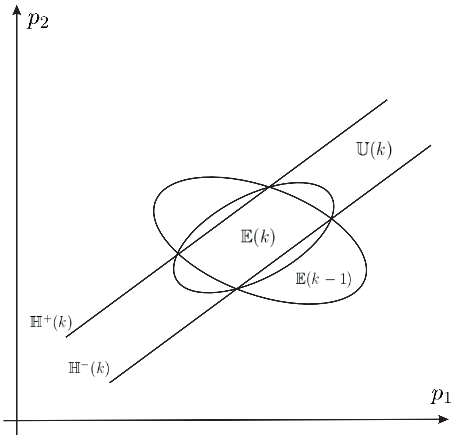

Fig. 3.7. Outer Bounding Ellipsoid algorithm

- the volume of the ellipsoid,
- the sum of the squares of the semi-axes of the ellipsoid,
- the value of the coefficient $\sigma(k)$.

The following ellipsoid is obtained using one of the criteria:

$$\mathbb{E}(k) = \{\boldsymbol{p} \in \mathbb{R}^n |\, (\boldsymbol{p} - \hat{\boldsymbol{p}}(k))^T \boldsymbol{M}^{-1}(k)(\boldsymbol{p} - \hat{\boldsymbol{p}}(k)) \leqslant \sigma^2(k)\}, \tag{3.83}$$

where $\boldsymbol{M}(k), \hat{\boldsymbol{p}}(k), \sigma(k)$ are calculated recursively by means of the following formulae:

$$\boldsymbol{M}(k) = \frac{1}{\eta_1(k)}\left[\boldsymbol{M}(k-1) - \frac{\eta_2(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)}{\eta_1(k)+\eta_2(k)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)}\right], \tag{3.84}$$

$$\hat{\boldsymbol{p}}(k) = \hat{\boldsymbol{p}}(k-1) + \eta_2(k)\boldsymbol{M}(k)\boldsymbol{r}(k)\big(y'(k) - \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}(k-1)\big), \tag{3.85}$$

$$\sigma^2(k) = \eta_1(k)\sigma^2(k-1) + \eta_2(k)\varepsilon_g^2(k) - \frac{\eta_1(k)\eta_2(k)[y'(k) - \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}(k-1)]^2}{\eta_1(k)+\eta_2(k)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)}, \tag{3.86}$$

where for the first and second criteria

$$\eta_1(k) = \frac{1}{\sigma^2(k-1)}, \tag{3.87}$$

$$\eta_2(k) = \frac{\lambda(k)}{\varepsilon_g^2(k)}, \tag{3.88}$$

and for the third criteria

$$\eta_1(k) = 1 - \lambda(k), \tag{3.89}$$

$$\eta_2(k) = \lambda(k). \tag{3.90}$$

The coefficient $\lambda(k)$ is calculated according to the chosen criteria. If the volume of the ellipsoid decides about the chosen ellipsoid,

$$vol\,\mathbb{E}(k) = \Omega_n\sqrt{det\big(\sigma^2(k)\boldsymbol{M}(k)\big)}, \tag{3.91}$$

where Ω_n is the volume of the unit sphere, then the following cost function is minimized:

$$J_{det}(k) = det(\sigma^2(k)\boldsymbol{M}(k)). \tag{3.92}$$

For such a procedure, the coefficient $\lambda(k)$ is obtained through solving the following quadratic equation:

$$a_1\lambda^2(k) + a_2\lambda(k) + a_3 = 0, \tag{3.93}$$

where

$$\begin{aligned}
a_1 =&(n-1)\sigma^4(k-1)[\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)]^2\\
a_2 =&\Big((2n-1)\varepsilon_g^2(k) - \sigma^2(k-1)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)+\\
&+\big(y'(k) - \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}(k-1)\big)^2\Big)\sigma^2(k-1)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)\\
a_3 =&\Big[n\big[\varepsilon_g^2(k) - \big(y'(k) - \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}(k-1)\big)^2\big]-\\
&-\sigma^2(k-1)\boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k)\Big]\varepsilon_g^2(k).
\end{aligned} \tag{3.94}$$

Finally, the optimal value of $\lambda(k)$ is given by the following equation:

$$\lambda(k) = \begin{cases} 0 & \text{if } a_3 \geqslant 0, \\ \frac{-a_2+\sqrt{a_2^2-4a_1a_3}}{2a_1} & \text{if } a_3 < 0. \end{cases} \tag{3.95}$$

The disadvantage of the presented criteria is the fact that the volume of the final ellipsoid may be small enough, but very wide along one of the axes. It means that the confidence interval for some parameters can be very big although the admissible set of parameters is generally small. The second criterion in the form of the sum of the squares of the semi-axes is free of the above-mentioned problem. The task of minimizing the width of semi-axes is equivalent to minimizing the trace of the matrix $\sigma^2(k)\boldsymbol{M}(k)$. In this case the coefficient $\lambda(k)$ is calculated by solving the following equation:

$$\lambda^3(k) + b_1\lambda^2(k) + b_2\lambda(k) + b_3 = 0, \tag{3.96}$$

where

$$\begin{aligned} b_1 =& \frac{3\varepsilon_g^2(k)}{\sigma^2(k-1)h(k)}, \\ b_2 =& \Big[\varepsilon_g^2(k)h(k)[J_{tr}(k-1)(\varepsilon_g^2(k) - g^2(k)) - \sigma^4(k-1)\gamma(k)] + \\ & + 2\varepsilon_g^2(k)[\varepsilon_g^2(k)h(k)J_{tr}(k-1) - \sigma^2(k-1)\gamma(k)(\varepsilon_g^2(k) - g(k))]\Big]/\psi(k), \\ b_3 =& \varepsilon_g^4(k)[(\varepsilon_g^2(k) - g^2(k))J_{tr}(k-1) - \sigma^4(k-1)\gamma(k)]/(\sigma^2(k-1)\psi(k)), \end{aligned} \tag{3.97}$$

and

$$\begin{aligned} h(k) &= \boldsymbol{r}^T(k)\boldsymbol{M}(k-1)\boldsymbol{r}(k), \\ g(k) &= y'(k) - \boldsymbol{r}^T(k)\hat{\boldsymbol{p}}(k-1), \\ \gamma(k) &= \boldsymbol{r}^T(k)\boldsymbol{M}^2(k-1)\boldsymbol{r}(k), \\ \psi(k) &= \sigma^4(k-1)h^2(k)[h(k)J_{tr}(k-1) - \sigma^2(k-1)\gamma(k)], \\ J_{tr}(k) &= trace(\sigma^2(k)\boldsymbol{M}(k)). \end{aligned} \tag{3.98}$$

The value of $\lambda(k)$ is determined using the following procedure:

$$\lambda(k) = \begin{cases} 0 & \text{jeżeli } b_3 \geqslant 0, \\ \lambda^*(k) & \text{jeżeli } b_3 < 0, \end{cases} \tag{3.99}$$

where $\lambda^*(k)$ is the positive real root of Eqn. (3.96).

The third approach for minimizing the size of the ellipsoid concerns minimizing the value of the coefficient $\sigma^2(k)$ (Dasgupta and Huang, 1987). The following constraints are introduced in order to calculate the parameter $\lambda(k)$:

$$0 \leqslant \lambda(k) \leqslant \vartheta < 1, \tag{3.100}$$

where the variable ϑ is introduced to ensure the boundedness of the matrix $\boldsymbol{M}(k)$. For such conditions, $\lambda(k)$ is described by the expression

$$\lambda(k) = \begin{cases} 0 & \text{if } \gamma(k) \geqslant 0, \\ \lambda^*(k) & \text{if } \gamma(k) < 0, \end{cases} \tag{3.101}$$

where

$$\lambda^*(k) = \min(\vartheta, \epsilon(k)). \tag{3.102}$$

In the previous formulae the following variables were introduced:

$$\gamma(k) = \frac{\varepsilon_g^2(k) - \sigma^2(k-1)}{g^2(k)}, \tag{3.103}$$

$$\epsilon(k) = \begin{cases} \vartheta & \text{if } g(k) = 0, \\ \frac{1-\gamma(k)}{2} & \text{if } h(k) = 1, \\ \vartheta & \text{if } \gamma(k)(h(k)-1)+1 \leqslant 0, \\ \frac{1}{1-h(k)}\left(1 - \sqrt{\frac{h(k)}{\gamma(k)(h(k)-1)+1}}\right) & \text{if } \gamma(k)(h(k)-1)+1 > 0, \end{cases} \tag{3.104}$$

and the variables $g(k)$ and $h(k)$ were defined by Eqn. (3.98).

The shown variant of the BEA algorithm is also called the *degenerate minimal volume algorithm* because the procedure is not able to find the minimal ellipsoid but successively finds ellipsoids with a smaller volume.

3.4.1. Calculating model uncertainty using the Outer Bounding Ellipsoid algorithm

In the previous section the approach for calculating the parameters of the model using the OBE algorithm was shown; however, in the context of fault detection a very important task is to determine the confidence interval for the output of the model based on known admissible set of parameters defined for that model. In the previous sections it was shown that the Takagi-Sugeno N-F network can be viewed in the form of the LP system, thus all deliberations conducted here for such systems are also true for Takagi-Sugeno models. The confidence interval for the output of the model with parameters determined by the OBE method is calculated similarly to the approach used in the case of the BEA algorithm. Two worst cases are chosen in the form of parameter vectors belonging to the ellipsoid which generate correspondingly the smallest possible and the biggest possible value of the model output:

$$\boldsymbol{r}^T(k)\hat{\boldsymbol{p}} - \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)} \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant \boldsymbol{r}^T(k)\hat{\boldsymbol{p}} + \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)}, \tag{3.105}$$

where $\boldsymbol{M}$ is the matrix determining the orientation of the ellipsoid, $\hat{\boldsymbol{p}}$ is the vector defining the center of the ellipsoid. The above-mentioned formulae result from the analysis of the size of the ellipsoid, as is shown in Fig. 3.8.

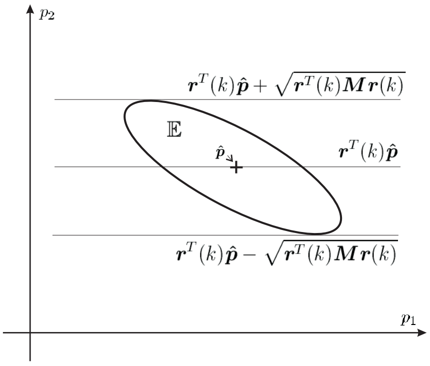
p_2
$\boldsymbol{r}^T(k)\hat{\boldsymbol{p}} + \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)}$
$\mathbb{E}$
$\hat{\boldsymbol{p}}$
$\boldsymbol{r}^T(k)\hat{\boldsymbol{p}}$
$\boldsymbol{r}^T(k)\hat{\boldsymbol{p}} - \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)}$
p_1

Fig. 3.8 . Determining the confidence interval by analyzing the size of the ellipsoid

For such a confidence interval defined for the model, the corresponding confidence interval for the system is defined by means of the following inequalities:

$$\begin{aligned}\boldsymbol{r}^T(k)\hat{\boldsymbol{p}} - \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)} - \varepsilon_g(k) \leqslant y'(k) \leqslant \\ \boldsymbol{r}^T(k)\hat{\boldsymbol{p}} + \sqrt{\boldsymbol{r}^T(k)\boldsymbol{M}\boldsymbol{r}(k)} + \varepsilon_g(k),\end{aligned} \tag{3.106}$$

where $\varepsilon_g(k)$ is a constraint introduced in the formulae (3.79)

This work deals with the problem of fault detection in real industrial systems using N-F models, so it must be assumed that input signals are uncertain, like output signals of the model. Such an assumption requires some modifications in the above procedure for calculating the confidence interval. Similarly like for the BEA algorithm, let us assume that the errors of particular inputs are bounded:

$$-e_i^g(k) \leqslant e_i(k) \leqslant e_i^g(k). \tag{3.107}$$

For such constraints, the confidence interval for the output is given by means of the following inequalities:

$$\begin{aligned}(\boldsymbol{r}'(k))^T\hat{\boldsymbol{p}} + \sum_{i=1}^{n} sgn(\hat{p}_i)\hat{p}_i e_i^g(k) - \sqrt{(\hat{\boldsymbol{r}}'(k))^T\boldsymbol{M}\hat{\boldsymbol{r}}'(k))} \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} \leqslant \\ (\boldsymbol{r}'(k))^T\hat{\boldsymbol{p}} + \sum_{i=1}^{n} sgn(\hat{p}_i)\hat{p}_i e_i^g(k) + \sqrt{(\hat{\boldsymbol{r}}'(k))^T\boldsymbol{M}\hat{\boldsymbol{r}}'(k))},\end{aligned} \tag{3.108}$$

where

$$\hat{r}_i'(k) = r_i'(k) + sgn(r_i'(k))e_i^g(k). \tag{3.109}$$

The presented constraints depend on the sign of the parameters and the sign of the parameters is unknown, thus it is required to replace each parameter with the difference of two positive parameters, as was done for the BEA algorithm (3.60). This approach allows using the discussed method to determine the confidence interval for the output of the Takagi-Sugeno N-F network. The only required modification regards the replacing of the original input variables with modified input variables, as was described in Section 3.2.2. In order to use the method for calculating the confidence interval for dynamic systems, the approach presented in Section 3.3.4 must be considered.

3.5. Summary

A lot of different algorithms for estimating the parameters of models using the available measurements are widely presented in scientific literature. However, these methods rarely ensure the high level of reliability required in the case of diagnostic problems. Reliability may be defined in the form of robustness against disturbances that corrupt measurements. Therefore, a robust fault detection system should be able to tolerate disturbances in order to avoid the generation of false alarms. The well-known approach that ensures robustness against disturbances is based on the estimation of the confidence interval for the output of the model, which should include the correct value of the model output. Obviously, the confidence interval is calculated using the available knowledge about the range of disturbances which are the reasons for model uncertainty. From the diagnostic point of view the confidence interval should be calculated in such a way that it could represent the worst scenario corresponding to the highest level of disturbances. This approach makes the avoidance of false alarms possible but at the same time it reduces the sensitivity of the system to real faults due to the wide confidence interval. In order to overcome this problem, all available knowledge about disturbances must be used to minimize the confidence interval to an acceptable size. The problem of determining the confidence interval is strongly related to the problem of parameter estimation due to the fact that parameters are estimated using corrupted measurements.

This chapter presents two approaches to tackle the problem of N-F model uncertainty and the estimation of its parameters. The first approach is based on the statistical method, which can be used to estimate the uncertainty of the model obtained using the LS method, and the second approach is based on the BEA method, which can be used to estimate model parameters and to calculate model uncertainty as well. Both methods are applicable only to LP systems, and in the case of non-linear systems linearization around the working point is required. Unfortunately, this approach can introduce some unprecision into the confidence interval, thus the required reliability of fault detection may be deteriorated. Therefore in this work an alternative approach is proposed to overcome the problem. A non-linear system in the form of the Takagi-Sugeno N-F network is converted to an LP system assuming that the consequents of fuzzy rules have the form of linear models and the parameters of fuzzy sets are known because they are determined earlier with another estimation procedure.

The practical usefulness of the methods of parameter estimation or model uncertainty determination is very important in the context of fault detection for real systems. Unfortunately, the LS method and the statistical approach do not satisfy these criteria due to the fact that severe assumptions concerning disturbances have to be fulfilled in order to make the use of those methods possible. These assumptions restrict the use of the methods to systems that are corrupted by disturbances, which are very rarely met in reality (normal distribution and expectation equal to 0). The application of these methods without fulfilled assumptions usually leads to strongly inaccurate models, which in the case of diagnostics is unacceptable. The solution to these problems is the BEA method, which requires only a range of disturbances to work properly. The simplest approach states that the output of the model is the only uncertain variable, but from the practical point of view the inputs of the model must be usually treated like uncertain variables as well. This case is taken into account in the modified BEA method also presented in this chapter. Dynamic autoregressive systems are special cases of systems with uncertain input and output variables. A special algorithm that calculates on-line the range of errors in input variables must be employed for such systems. Unfortunately, the computational cost of BEA methods is very high, especially the requirement concerning the memory space is very hard to satisfy, thus the method is applicable only to simple models. The problem can be overcome in the case of N-F models by replacing the BEA method with the much less complicated OBE algorithm. The computational cost of the OBE method is smaller because it only approximates the exact admissible set of parameters (polytope) with the ellipsoid. However, the confidence interval calculated using the admissible set of parameters and bounded by the ellipsoid is usually more pessimistic than the real confidence interval due to the fact that the ellipsoid encloses the real admissible set of parameters. The important advantage of the BEA method is the fact that data required for calculating the uncertainty admissible set of parameters can be determined during the estimation of model parameters.

The main problem concerning the presented methods is the fact that the parameters of fuzzy sets have to be known in order to recive the Takagi-Sugeno N-F model as the LP system and to make use of the BEA or the OBE algorithm possible. The next chapter deals with this problem by proposing methods of estimating the parameters of fuzzy sets included in the N-F model.

Chapter 4

NEURO-FUZZY NETWORK DESIGN

4.1. Introduction

The procedure of N-F network design, similarly to the other techniques, consists of the structure identification stage and the parameter estimation stage (Rutkowska, 1997; Rutkowska, 2002). The pessimistic scenario assumes the construction of the N-F network only on the basis of the available measurements. The main problem is to obtain the required accuracy and transparency of the rule base in such a situation. A lot of different methods have already been developed both for structure selection and parameter estimation of the N-F network, but there is a demand for better, more effective algorithms and active research is still conducted in this area (Diez *et al.*, 2002; Rutkowski, 2004a; Rutkowska *et al.*, 2000; Guven and Passino, 2001; Hadjili and Wertz, 2002; Hong and Harris, 2001; Juang and Lin, 1999; Korbicz and Kowal, 2001).

The application of N-F networks in diagnostic areas creates a demand for suitable design procedures which would take into account the specificity of the fault diagnosis task. An important problem from the diagnostic point of view is residual confidence interval minimization because it makes it possible to detect a fault appropriately early. It has to be stressed that the value of the confidence interval for residuals depends directly on the uncertainty of the model which is used to generate the residuals. If the confidence interval is not consistent with model uncertainty, the fault detection system can trigger off a lot of false alarms. It is obvious in such a situation that model uncertainty has to be considered in fault detection threshold calculations (Patton and Chen, 1999; Witczak, 2003; Mrugalski, 2003). It is also important to minimize model uncertainty in order to obtain a reliable fault detection system that would be able to detect a fault fast and at an early stage, so special procedures for N-F model design must be developed.

The present chapter deals with the problem of N-F model design for fault detection applications. A general overview of the known algorithms for N-F network structure design and parameter estimation is given. Some problems connected with the presented algorithms in the context of their construction for fault detection applications are indicated. A new effective algorithm for N-F network design is developed.

4.2. Neuro-fuzzy network design and knowledge transparency

The N-F network design procedure, similarly to neural network design algorithms, relies mainly on quantitative knowledge in the form of the available measurements. It is obvious in such a situation that algorithms known for neural networks can be adapted for N-F networks due to a similar topology (Duch *et al.*, 2000; Rutkowska, 1997). However, it should be mentioned that neural networks are treated like *black boxes*, thus knowledge included in their parameters is not transparent for the user and cannot be easily understood or analyzed. In contrast to that, N-F networks represent knowledge in the form of fuzzy rules and so it can be understood by the user. Systems that represent knowledge in the form of fuzzy rules are usually called *grey boxes* due to the properties mentioned above. Algorithms for neural network design do not take into consideration the criteria of knowledge transparency, so their direct applications to N-F networks usually lead to the construction of obscure structures (Piegat, 2003). Nevertheless, such an approach is commonly used in practical applications because it ensures the required accuracy. Unfortunately, it also reduces the N-F network to a *black box* system, so the main advantage of the approach is lost and probably similar accuracy can be obtained using neural networks. Algorithms for neural network design must be adjusted to the specificity of N-F techniques in order to take advantage of the important property, i.e. the transparency of knowledge (Lin and Cunningham, 1991; Kowal *et al.*, 2002; Kowal and Korbicz, 2002a; Kowal and Korbicz, 2002b; Kowal and Korbicz, 2001).

N-F networks, similarly to fuzzy systems, store knowledge in the form of fuzzy rules. A single fuzzy rule is responsible for a certain sector of the analysed space, where the consequent of the rule is active. The bounds of activity areas are realized by antecedents, which are described by fuzzy sets. The degree of the activity of the rule can be different for different points in the area which it represents, and depend on the shape of the membership function of the fuzzy set. Taking into account the above facts, it can be claimed that the transparency of knowledge depends on the way fuzzy rules partition the input-output space. The following conditions should be satisfied to ensure the transparency of knowledge coded using fuzzy rules:

- the way the space is partitioned should be clear, the number of rules should come from a natural division of the space considered,

- a single rule should group points from the input-output space that are in the neighborhood and are similar taking into consideration significant properties,

- the granulity of the space should not be too high because it leads to small significance of a single rule,

- the rule base should be complete,

- the rule base should be consistent.

The specified conditions can be satisfied by introducing appropriate constraints during the structure selection and parameter estimation stage. Nevertheless, the design procedure without constraints makes it possible to build accurate N-F systems, but the obtained knowledge is abstract and difficult to interpret (Piegat, 2003; Babuška, 1998). Such an effect is manifested in the form of areas in the space considered where the sum of rule fulfilments is higher than 1 due to strong activities of several rules in the same region of the space. An opposite situation occurs if fuzzy rules are located far away from each other and the sum of the fulfilments of the rules is lower than 1 or even equal to 0. The situation when the sum of fulfilments is higher than 1 can be interpreted by the inconsistency of the rules. Few rules for the same point in the input-output space have a high degree of fulfillment and the consequents of these rules do not describe properly the local properties of the analysed area of the space, so the transparency of these rules is lost. The situation when the sum of fulfillments is lower than 1 means that the obtained conclusion is justified insufficiently, so the rule base is not complete.

The following advantages arise from transparent knowledge:

- the analysis of knowledge is simplified,
- the process of N-F network design can be automated,
- the structure of the N-F network is clear,
- the parameter tuning process is simplified and shortened,
- the system is more robust against disturbances.

It is important for fuzzy rules to catch natural local dependencies in the input-output space in order to increase the probability that the parameters of fuzzy rules will have physical interpretation. This fact can be very useful in practical applications, where systems built using automatic algorithms must be verified before their implementation in order to ensure maximal reliability of the system. Transparent knowledge makes this task easier and faster. An important condition required for knowledge transparency states that fuzzy rules partition the input-output space in such a way that similar regions of the space are grouped and described by a single rule. Taking into account this fact, the design process can be automated using clustering algorithms and adjusting the granulity of information for the expected accuracy of the system (Abonyi *et al.*, 2002; Babuška, 1998; Chen *et al.*, 1998). If the designed system does not satisfy the accuracy considered, the system can be easily analyzed in order to find the reasons for the error, and the accuracy can be improved by increasing the granulity of information. It is also possible to use data-mining techniques to extract the local behavior of the process in order to initialize the parameters of the N-F network (Geva, 1999; Jin, 2000; Setnes, 2000). Such an initialization usually significantly shortens the time required for parameter tuning and ensures the transparency of knowledge.

4.3. Methods of neuro-fuzzy network design

Takagi-Sugeno N-F networks can be viewed as multi-model systems which consist of some rules, and each rule defines a single model as the consequent of the rule (Babuška, 1998; Kowal and Korbicz, 2003; Kowal and Korbicz, 2002a; Kowal and Korbicz, 2002b). The global N-F system is a set of N_r partial models, where N_r determines the number of fuzzy rules. The output of the global system is calculated as a mixture of partial model outputs. The rule fulfillment is determined by fuzzy sets. In order to ensure the desired accuracy of the N-F system, the membership functions of fuzzy sets must be placed properly in the input space, the number of rules must be appropriate and the parameters of partial models must be chosen to minimize the defined error. Two main strategies for placing fuzzy sets in the input space can be distinguished: the first one proposes to minimize the output error of the global model (Leith and Leithead, 1999), and the other one is based on partial models that model the local behavior of the system (Rice and Xu, 1996; Abonyi *et al.*, 2002). A typical property of the first approach is to arrange fuzzy sets in the input space in such a way that all partial models are active in the whole domain of input variables. In this case, the accuracy of the global model is guaranteed by the proper mixture of partial model outputs. The alternative approach does not examine the global accuracy of the model but concentrates on partial models, which should tune in to the local behavior of the system. In the case considered, where the consequents of fuzzy rules are defined in the form of linear partial models and their parameters are determined using the BEA algorithm, only the second approach may be effectively applied to tune fuzzy sets because computation costs of the first approach are very high due to the fact that the parameters of all consequents must be determined in each iteration of the hypothetical algorithm.

4.3.1. Rule base declaration

The problem of rule base declaration reduces to the determination of the number of rules required for precise description of the problem to be solved. It can be viewed as the process of choosing the right granulity of information which is processed by the N-F network. The analysed step of N-F model design is fundamental regarding the accuracy of the model, its generalization properties and robustness against disturbances. Unfortunately, there does not exist an analytical criterion which can help to determine the optimal number of rules if only measurements are available to build the N-F network. This is so because it is hard to estimate the influence of rule base selection on the accuracy of the whole N-F model at this stage of model design. Only general directions concerning structure selection are known, which recommend to find a certain compromise between structure complexity and the accuracy of operation (Osowski, 1996; Duch *et al.*, 2000). It is important not to pursuit the accuracy of the model at all costs because the generalization properties of the model are lost, the structure of the model is very complex and the model is not robust against disturbances.

4.3.1.1. Grid methods

The simplest method used to determine the number of rules is based on generating a uniformly distributed grid of rules in the input space (Fig. 4.1). The usage of such an approach is limited only to simple systems with a small number of inputs. The approach does not work well for more complicated systems because it generates a combinatorial explosion of rules, which make this method useless. Practically, the described method does not help the designer to choose the right number of rules but only limits the number of all possible configurations of rules by using the uniformly distributed grid, so the indicated problem is still open. A certain rationalization of the method introduces the approach which proposes to partition the input space into a great number of small areas with a single rule generated for each area. Next, adjacent rules are tested to determine the similarity between them using the chosen criterion (Yen and Wang, 1999; Pomares *et al.*, 2000). A sample similarity criterion is expressed by the following formula:

$$S(A,B) = \frac{M(A \cap B)}{M(A) + M(B) - M(A \cap B)}, \tag{4.1}$$

where A and B are adjacent multidimensional fuzzy sets and the expression $M(A)$ determines the size of the fuzzy set A:

$$M(A) = \int \cdots \int_D A(x_1, \ldots, x_n)\, \mathrm{d}x_1 \, \ldots \, \mathrm{d}x_n. \tag{4.2}$$

Unfortunately, this method requires to check the accuracy of the N-F network after each step of the algorithm, so all stages of N-F network design must be completed and the procedure is time consuming. The algorithm can be simplified by introducing the value γ, and if all values S_i are smaller than γ, the algorithm can be stopped. However, the procedure is simplified, but there is no certainty that the assumed accuracy of the N-F network will be obtained, and it is hard to determine the right value of γ.

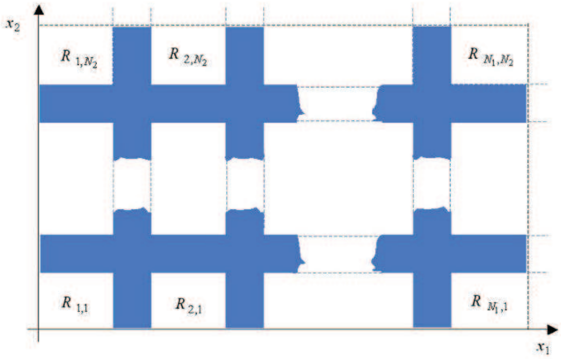

Fig. 4.1 . Grid partition method

4.3.1.2. Clustering algorithms

Fuzzy clustering algorithms are another technique which is often used for fuzzy rule generation (Bezdek, 1981; Kaymak and Setnes, 2002; Kirshnapuram and Kim, 1999; Chen *et al.*, 1998; Chiu, 1994; Babuška, 1998). The idea of this approach is to find natural groups of data in order to apply to each group one fuzzy rule (Babuška, 1998). Generally, clustering algorithms can be divided into two main classes, i.e. hard clustering and fuzzy clustering. It seems to be natural to use fuzzy clustering algorithms in the case of N-F networks. The task of fuzzy clustering is usually reduced to finding the local minimum of the nonlinear cost function, defined by the following expression:

$$J(\boldsymbol{X};\boldsymbol{U},\boldsymbol{V}) = \sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}D_{ik}^{2}, \tag{4.3}$$

where the matrix $\boldsymbol{U}$ contains the membership degrees of data points from the matrix to the defined clusters $\boldsymbol{X}$, $\boldsymbol{V} = [\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_c}], \boldsymbol{v_i} \in R_n$ is a matrix which defines the centers of the clusters, D_{ik} is a metric used to determine the distance between data points and cluster centers:

$$D_{ik}^{2} = \|\boldsymbol{x_k} - \boldsymbol{v_i}\|^2 = (\boldsymbol{x_k} - \boldsymbol{v_i})^T\boldsymbol{A}(\boldsymbol{x_k} - \boldsymbol{v_i}), \tag{4.4}$$

and the parameter m takes values from 1 to ∞ and determines the degree of fuzziness of clusters. The cost function (4.3) can be viewed as a total variance of the data $\boldsymbol{x_k}$ with respect to the cluster centers $\boldsymbol{v_i}$. The matrix A which occurs in the expression (4.4) is used to tune the shape and orientation of clusters in the space. In the simplest approach, the matrix A is unitary, thus the distance measure D_{ik} is an Euclidean norm. The fuzzy clustering algorithm which uses such a norm to calculate the distance between data points and cluster centers is called the *Fuzzy C-Mean* (FCM) (Bezdek, 1981). The procedure used to find the minimum of the cost function (4.3) usually employs the Picard iteration through first-order conditions for stationary points of the cost function. A family of clustering algorithms can be derived from the basic FCM by developing new approaches for calculating the matrix $\boldsymbol{A}$. The family of such fuzzy clustering algorithms can be divided into two classes:

- adaptive distance measure D_{ik} algorithms (i.e. the Gustafson-Kessel algorithm, the Gath and Geva algorithm) (Gath and Geva, 1989; Gustaffson and Kessel, 1975),
- algorithms which use cluster prototypes in the form of linear or nonlinear subspaces of data space (i.e. Fuzzy C-Varieties (FCV), Fuzzy C-Elliptotypes (FCE), Fuzzy C-Regression Models (FCRM))(Babuška, 1998).

The main disadvantage of the FCM algorithm is the fact that the algorithm strongly prefers spherical clusters due to a fixed distance norm. The algorithm detects spherical clusters even if such clusters do not exist. A remedy for such a drawback is the adaptive distance measure. Similarly to the FCM, the Gustafson-Kessel algorithm uses the matrices U and V as optimization variables in the cost functional (4.3), but in opposition to the standard FCM scheme also the matrices $\boldsymbol{A_i}$ are treated as optimization variables. The presented modifications make it possible to detect clusters of different geometrical shapes because each cluster has its own norm inducing the matrix $\boldsymbol{A_i}$. Unfortunately, the Gustafson-Kessel algorithm is able to detect clusters of similar sizes only. In order to overcome this problem, the Gath and Geva algorithm generates the distance measure D_{ik} using the fuzzy maximum likelihood estimates proposed in (Bezdek and Dunn, 1975), and is able to detect clusters of varying shapes, volumes and densities.

Another approach which makes it possible to detect clusters different than spherical ones states that a cluster can be described by its center but also by the function that determines its shape. The simplest method from the family of such clustering methods is the FCV algorithm. The main idea behind this algorithm is to measure the distance of data from r-dimensional linear varieties. The distance measure is defined in the form of a squared orthogonal distance from the data point to linear variety:

$$D_{ik} = \|\boldsymbol{x_k} - \boldsymbol{v_i}\|^{\boldsymbol{2}} - \sum_{\boldsymbol{j=1}}^{\boldsymbol{r}} (\boldsymbol{x_k} - (\boldsymbol{v_i}, \boldsymbol{s_{ij}}))^{\boldsymbol{2}}, \tag{4.5}$$

where $\boldsymbol{v_i}$ is a point which the variety passes, $\boldsymbol{s_{i1}}, \boldsymbol{s_{i2}}, \ldots, \boldsymbol{s_{ir}}$ is an r-tuple of linearly independent vectors which spans the hyperplane, and $\boldsymbol{(,)}$ denotes the scalar product. Thanks to such an approach the algorithm is able to detect clusters lying in r-dimensional linear subspaces of R^n. Nevertheless, the algorithm has a drawback, i.e. linear varieties are not limited in size so the algorithm tends to connect similar linear clusters which are physically well separated.

The FCE algorithm, which is a combination of the FCM and FCV, has been developed in order to tackle the above-mentioned problem. A new coefficient in the form of the center of gravity of the cluster $\boldsymbol{v_i}$ is used to describe the cluster. The distance measure employed in the FCE algorithm is a mixture of distance measures known for FCM and FCV:

$$D_{ik} = \alpha D_{ik}^{FCM} + (1-\alpha) D_{ik}^{FCV}, \tag{4.6}$$

where the coefficient α decides about the shape of clusters.

The main disadvantage of the presented methods in the context of N-F network design is the fact that these algorithms do not take into account the accuracy of the N-F network in the cost function. In some sense this problem is solved by the FCRM clustering algorithm. The algorithm partitions the data space and simultaneously estimates the parameters of regression models associated with clusters. The prediction error defined for regression models has the following form:

$$E_{ik}(\theta_i) = [y_k - f_i(\boldsymbol{x_k}; \theta_i)]^2, \tag{4.7}$$

where θ_i is a vector of model parameters. The introduced error is employed to define the distance measure used to calculate the distance between data and clusters, thus the cost functional is given by

$$J(\boldsymbol{X};\boldsymbol{U},\{\theta_i\}) = \sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}E_{ik}(\theta_i). \tag{4.8}$$

The algorithm inherits from the FCV algorithm a specific inconvenience, i.e. the clusters are not limited in size. Sometimes this fact can cause wrong partitioning of the data space by the algorithm. The main advantage of the algorithm in the context of N-F network design is the possibility to derive the parameters of linear consequents from regression models used to describe clusters. However, such models ensure precise behavior of individual rules only and do not take into account the global behavior of the N-F model (but other techniques do not take into account this fact at all).

The presented algorithms have one main disadvantage if they are used for rule base declaration because they are not able to determine the proper number of clusters which should be used to partition the data space. Unappropriate selection of clusters is manifested in weak cluster separation and weak cluster concentration, therefore the rule base extracted from clusters is not correct. Two main strategies for estimating the optimal number of clusters can be distinguished. The first one is based on repeating the clustering procedure for different numbers of clusters, and then the best configuration in some sense is chosen (Babuška, 1998; Kothari and Pitts, 1999). The following criteria employed to evaluate the quality of clustering are commonly used:

- fuzzy hypervolume:

  $$V_h = \sum_{i=1}^{c}[det(\boldsymbol{F}_i)]^{\frac{1}{2}}, \tag{4.9}$$

  where $\boldsymbol{F}_i$ is a cluster covariance matrix; a properly partitioned data space is characterized by small values of V_h;

- average cluster density:

  $$D_A = \frac{1}{c}\sum_{i=1}^{c}\frac{S_i}{[det(\boldsymbol{F}_i)]^{\frac{1}{2}}}, \tag{4.10}$$

  where $S_i = \sum_k \mu_{ik}, \quad \forall k$ such that $(\boldsymbol{x_k}-\boldsymbol{v_i})^T\boldsymbol{F}_i^{-1}(\boldsymbol{x_k}-\boldsymbol{v_i})$; good clustering is indicated by large values of D_A;

- cluster density:

  $$D_P = \frac{\sum_{i=1}^{c} S_i}{V_h}, \tag{4.11}$$

  where, similarly to the previous criterion, a larger value of D_P indicates better clustering results;

- within-cluster distance (Kirshnapuram and Freg, 1992):

$$W_T = \sum_{i=1}^{c} \sum_{k=1}^{N} \mu_{ik} D_{ik}^2, \tag{4.12}$$

- average within-cluster distance (Kirshnapuram and Freg, 1992):

$$W_A = \frac{1}{c} \sum_{i=1}^{c} \frac{\sum_{k=1}^{N} \mu_{ik} D_{ik}^2}{\sum_{k=1}^{N} \mu_{ik}}, \tag{4.13}$$

- average cluster flatness (Babuška and Verbruggen, 1995):

$$F_A = \frac{1}{c} \sum_{i=1}^{c} \frac{\lambda_{min}^i}{\lambda_{max}^i}, \tag{4.14}$$

  where λ_{min}^i and λ_{max}^i are the minimum and the maximum eigenvalue of the fuzzy covariance matrix $\boldsymbol{F}_i$, respectively. The criterion considered is used together with the sum of square errors:

$$E = \frac{1}{N} \sum_{k=1}^{N} (y_k - \hat{y}_k)^2, \tag{4.15}$$

  where y_k and $\hat{y}_k$ are true data and the predicted output, respectively, and N is the number of data items. The performance criterion is obtained by the combination of (4.14) with (4.15):

$$E_F = F_A E. \tag{4.16}$$

  The criterion given prefers solutions with a small number of flat clusters over partitioning using a large number of clusters if both solutions lead to a similar error E.

The presented techniques, which help to determine the number of clusters, state that the whole procedure is repeated for different cluster configurations c. Sometimes the procedure must be repeated for the same number of clusters due to a problem with the optimization algorithm, which can get stuck in local minima. For each iteration of the algorithm the value of the chosen criterion is calculated and the best configuration of clusters is determined. The problem of the presented method is a high computational cost of the algorithm and the necessity to define empirically thresholds for measures in order to stop the searching procedure. It should be noticed that the proposed measures do not minimize the modelling error or the uncertainty of the model, which in the case of diagnostic applications is a very important issue.

The second strategy for detecting the optimal number of clusters proposes to generate a large number of clusters in order to minimize the number of clusters by merging compatible ones (Babuška, 1998). The algorithm requires the criteria to measure the similarity of clusters (Kaymak and Babuška, 1995). The following similarity measures are commonly used:

$$s_{ij}^1 = |\phi_i^T \phi_j| \geqslant k_1, \tag{4.17}$$

$$s_{ij}^2 = \|\boldsymbol{v}_i^{'} - \boldsymbol{v}_j^{'}\| \leqslant k_2, \tag{4.18}$$

where ϕ_i i ϕ_j are unit eigenvectors for the fuzzy covariance matrices $\boldsymbol{F}_i$ and $\boldsymbol{F}_j$, $\boldsymbol{v}_i^{'}$ and $\boldsymbol{v}_j^{'}$ are normalized cluster centers:

$$\boldsymbol{v}_k^{'} = \frac{\boldsymbol{v}_k - \bar{\boldsymbol{v}}}{\sigma_k}. \tag{4.19}$$

The presented approach is computationally less complex than the previous one, which tested different cluster configurations, but again the problem with defining suitable thresholds controlling the merging procedure arises. It is hard to determine if the clusters are compatible enough to be joined. The problem is crucial due to the fact that the value of the merging threshold decides indirectly about the number of clusters.

Among the different clustering algorithms there exist a group of grid-type clustering algorithms, which can be used to declare the rule base. The mountain method is a sample algorithm from this class (Yager and Filev, 1994; Yager and Filev, 1994). It does not require knowledge about the number of clusters and this value is determined automatically by the clustering procedure. However, the number of the found clusters strongly depends on the values of coefficients, which must be defined by the designer at the beginning of the procedure, so the application of the algorithm is difficult.

The discussed clustering methods despite their numerous disadvantages are used in practical applications to build, e.g. N-F networks. An example is the toolbox ANFIS (*Adaptive Network Based Fuzzy Inference System*) from MATLAB (Jang, 1993), which uses subtractive clustering to determine the number of rules and to initialize the parameters of the rules.

4.3.1.3. Maximal error method

The idea of the maximal error method is to estimate roughly the system considered with a single fuzzy rule (Higgins and Goodman, 1994). Next, the N-F network is built and the output error for all samples is calculated:

$$e_i = (y_i - \hat{y}_i)^2. \tag{4.20}$$

The data point for which N-F networks make the maximal error is determined and it becomes the center of a new fuzzy rule. The procedure is repeated until the expected level of the error is reached (4.15). The method is simple but has a lot of disadvantages: it is not robust against disturbances, in some cases it generates a lot of rules, the parameters of fuzzy rules do not have physical interpretation. A modification of the method was proposed in (Piegat, 2003), where rules modelling the error are added in each step of the algorithm. The output of the rules which models the error is subtracted from the output of the main N-F model. Unfortunately, this method can lead to complicated structures of N-F networks and is rarely used in practice.

4.3.1.4. Search algorithms

The idea of the method is to build a tree which contains nodes. Each node stores information about the structure of the N-F network. Next, a search algorithm is used to find the optimal structure of the N-F network (Doering *et al.*, 1997). There exist two main strategies for searching for the solution: an uninformed search and an informed search. The uninformed search is very easy to implement but the computational costs are too high, thus only informed strategies can be effectively applied to build the N-F network. A special algorithm LOLIMOT (*Local Linear Model Tree*) for this purpose was proposed in (Nelles *et al.*, 2000). The procedure begins by defining the hypercube in the input space. A simple N-F model with one rule is built to cover the area bounded by the hypercube. Next, the hypercube is partitioned into smaller hypercubes and for each hypercube one fuzzy rule is generated. From the set of possible hypercubes, the hypercube that is responsible for the highest error is always chosen for the partition procedure. A sample tree obtained during the search for the structure of the N-F network is shown in Fig. 4.2. The procedure is repeated until the desired accuracy of the model is obtained.

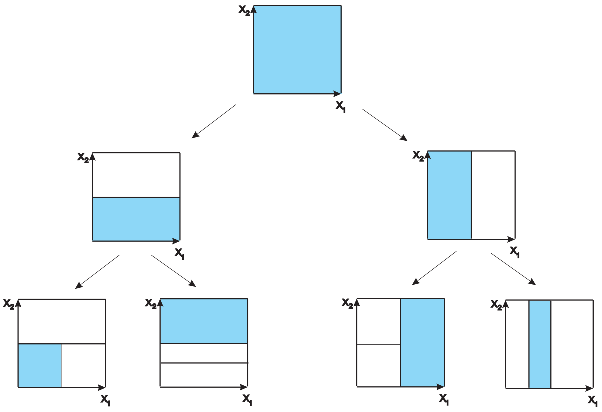

Fig. 4.2. Sample partition procedure

4.3.2. Parameter tuning

Parameter tuning is the last stage of the N-F network design procedure. The main aim of this stage is to achieve the desired precision of the N-F network by choosing suitable values of parameters (Rutkowska, 1997; Rutkowska *et al.*, 1997; Shi and Mizumoto, 2000; Wang and Mendel, 1992). In the case of the Takagi-Sugeno N-F network two types of parameters can be distinguished: parameters of fuzzy sets and parameters of consequents. The former are stored in the nodes from the 1st layer of the N-F structure, which is shown in Fig. 2.9. Each node stores the same set of parameters which describe a single fuzzy set. If Gaussian membership functions are used to describe the fuzzy set,

$$\mu_A(x) = e^{-(\frac{x-c}{d})^2}, \tag{4.21}$$

each node has two parameters c and d, where c is the center of the Gaussian function and d is the width. Of course, other membership functions can have different parameters. In the case of parameters stored in consequents, they can be treated like weights of artificial neurons due to the fact that each element of the 4th layer has on the input a single weight, which is used to multiply an adequate input signal. If the consequents have a linear form,

$$y = \sum_{i=0}^{n} w_i x_i, \tag{4.22}$$

then the nodes from the 4th layer can be treated like linear neurons with a set of weights $\boldsymbol{w} = [w_0, w_1, ..., w_n]$, where w_0 is a *bias*. The parameters from the 1st layer and from the 4th layer have different physical interpretation thus different algorithms can be employed for their tuning. The following approaches are used:

- parameters from the 1st layer are tuned using gradient descent algorithms and parameters of linear neurons from the 4th layer are determined using the LS algorithm

- parameters from the 1st layer are tuned using evolutionary algorithms and parameters of linear neurons are tuned using the LS algorithm,

- parameters from the 1st layer are determined using the clustering algorithm and parameters of linear neurons are tuned using the LS algorithm.

It is also possible to apply the same optimization algorithms for both types of parameters, i.e. the gradient descent approach or the evolutionary algorithm. Sometimes clustering algorithms are used to initialize the parameters and then the gradient descent approach or the evolutionary algorithm are used to tune them precisely. The main problem of the presented strategies is a lack of analysis concerning their influence on model uncertainty. Such an analysis is indispensable if the N-F network is applied to build a diagnostic system.

4.4. Bounded-error approach for rule base generation

The N-F network has a structure which may be easily converted to a set of fuzzy rules. This fact is often used to determine the structure of the N-F network because it is enough to known fuzzy rules in order to define the number of nodes in the 1st, the 2nd and the 5th layer. Moreover, information about membership functions and fuzzy operators can be also extracted from fuzzy rules. The identification of the N-F network can be viewed as a two-stage procedure, which first determines fuzzy rules and next the parameters of the fuzzy rules are tuned to minimize the criterion

$$J(\boldsymbol{y}', \boldsymbol{y}(s, \hat{\boldsymbol{p}})), \tag{4.23}$$

where $\boldsymbol{y}'$ represents the output of the system, $\boldsymbol{y}(s, \hat{\boldsymbol{p}})$ is the output of the N-F network, s is an index that describes the chosen structure, and $\hat{\boldsymbol{p}}$ is the vector of parameter estimates. Let us assume for the sake of simplicity that the N-F network has only one output. Then the criterion (4.23) takes the following form:

$$J(\boldsymbol{y}', \boldsymbol{y}(s, \hat{\boldsymbol{p}})) = \sum_{k=1}^{N} \big(y'(k) - y(s, \hat{\boldsymbol{p}})(k)\big)^2. \tag{4.24}$$

The identification procedure is to find such a structure of the N-F network and such parameters that the following condition is satisfied:

$$J(\boldsymbol{y}', \boldsymbol{y}(s, \hat{\boldsymbol{p}})) \leqslant e, \tag{4.25}$$

where e is a bound put on the modelling error.

In order to test generalization properties of the structure, the available data are divided into a learning and a testing set. The structure and parameters are determined using the learning set and the quality of the model is tested using the rest of the data.

The criterion (4.24) is most commonly used for the evaluation of N-F network accuracy in the case of modelling applications. Nevertheless, specific requirements of diagnostic applications force a modification of the criterion. The minimization of model uncertainty in the form of the confidence interval is more important than the minimization of sum square errors from the diagnostic point of view. Such a fact is considered in the following criterion:

$$J(\boldsymbol{y}^{max}(s, \hat{\boldsymbol{p}}), \boldsymbol{y}^{min}(s, \hat{\boldsymbol{p}})) = \sum_{k=1}^{N} \big(y^{max}(s, \hat{\boldsymbol{p}})(k) - y^{min}(s, \hat{\boldsymbol{p}})(k)\big), \tag{4.26}$$

where $y^{max}(s, \hat{\boldsymbol{p}})(k)$ and $y^{min}(s, \hat{\boldsymbol{p}})(k)$ are bounds that define the confidence interval for the output of the model.

Although the criterion is known, the problem of structure selection of the N-F model is not trivial. There are no algorithms which would be able, based on the available measurements, to determine in one step the optimal structure s_{min} of the N-F network. The proposed methods can be divided into two groups: discrete optimization methods and data exploration methods. The first approach proposes to search for the solution in the space of possible structures $\mathbb{S}$. The procedure contains also the parameter estimation stage in order to test in each iteration of the algorithm the quality of the model. The task of searching for the optimal structure is usually performed by informed search strategies or evolutionary algorithms. Unfortunately, the complexity of the procedure is huge due to a large number of possible structures and the requirement of parameter estimation in each iteration of the algorithm. Moreover, methods of parameter estimation can get stuck in local minima so they must be run many times in one iteration of the main algorithm.

Taking into account the problems mentioned above, data exploration algorithms seem to be less problematic than the previous approach. They usually separate the procedure of structure identification from the parameter estimation task. In the approach considered, data exploration algorithms should be understood as a procedure which searches for natural groups of data in the data space in order to generate for each group one rule that is able to describe important properties of the found group. It is often advisable to use clustering algorithms in order to detect the groups of data. In the case of Takagi-Sugeno N-F networks with linear consequences, the task is changed into a search for approximately linear dependencies in the data space. The number of rules with linear consequences can be estimated based on information about the number of the found approximately linear dependencies. Additionally, it is possible to estimate the parameters of fuzzy sets and linear consequences based on information about the size and location of the found approximately linear dependencies. However, the calculated parameters usually cannot ensure the desired accuracy but they can be used as a starting point for other methods of parameter estimation. A lot of research works in this area confirm the effectiveness of such an approach for N-F networks structure identification (Babuška, 1998). The main advantage of the approach is the fact that the method is not computationally complex and generates transparent knowledge. The disadvantage of the approach is that the method is not able to guarantee the accuracy of the generated model and the designer must have experience and a deep knowledge about clustering algorithms in order to define the parameters of the clustering to partition the data space properly.

4.4.1. Detection of linear dependencies

The procedure of rule base declaration using the BEA method, similarly to the data exploration algorithm, is based on the detection of approximately linear dependencies in the data space in order to generate a single fuzzy rule for each found linear dependency. The developed approach assumes that a non-linear system can be described by a mixture of linear partial models:

$$y_i(k) = \boldsymbol{r}^T(k)\boldsymbol{p}_i. \tag{4.27}$$

It is assumed that the number of partial models and their parameters $\boldsymbol{p}_i$ are unknown. Practically, it is impossible to ideally model the behavior of the nonlinear system by a finite number of linear partial models, thus it is required to define for each partial model an acceptable level of its uncertainty. Uncertainty can be described by the maximal error which is generated by such a model. The exact value of the error must be defined by the designer in order to meet the demanded accuracy of the model. The value of the mentioned coefficient has a direct impact on the number of partial models which will be generated. It should be supposed that a very small value of the error may cause the generation of a large number of partial models, while in the opposite situation the suitable accuracy of the model can be threatened. The designer must find a certain compromise between the accuracy of the model and its complexity, choosing an adequate value of the coefficient ε.

The analytical form of system characteristics is usually unknown in practical applications so it must be represented by the available measurements. In order to represent the characteristics of the system properly, a suitable experiment must be performed and the system must be actuated to make it possible to catch its behavior at many working points. The result of the experiment has the form of a set of N data points:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}(1) \\ \vdots \\ \boldsymbol{r}(N) \end{bmatrix}, \boldsymbol{y}' = \begin{bmatrix} y'(1) \\ \vdots \\ y'(N) \end{bmatrix}.$$

The obtained measurements represent the characteristics of the system and are used to detect approximately linear dependencies within system characteristics. Let us simplify for the moment the problem and assume that a single approximately linear dependency must be detected in a given data set. The main idea of the developed algorithm is to bound the error made by the single partial model:

$$\varepsilon \leqslant \boldsymbol{r}^T(k)\boldsymbol{p} - y'(k) \leqslant \varepsilon. \tag{4.28}$$

The output of the system in the approximately linear part of its characteristics can be given by the following equation:

$$y_i'(k) = \boldsymbol{r}^T(k)\boldsymbol{p}_i + \varepsilon, \tag{4.29}$$

where ε describes the permitted deviation for linear behavior of the system in the defined interval of the characteristics. In such a case the task can be formulated as the search for the set of data points which compose the approximately linear dependency in the data space. Additionally, it must be assumed that points that form linearity have to lay in the neighborhood. The developed procedure consists of the following steps:

- from the set of N measurements, a single data point is selected (in the first step of the algorithm the point which represents the beginning of system characteristics is selected),
- for the selected data point and defined ε, a feasible set of parameters is determined using the BEA algorithm:

$$\mathbb{S}_1 = \{\boldsymbol{p} \in \mathbb{R}^n \,|\, y'(1) - \varepsilon \leqslant \boldsymbol{r}^T(1)\boldsymbol{p} \leqslant y'(1) + \varepsilon\}, \tag{4.30}$$

- the next data point that lays nearby the previous point is selected,
- the feasible set of parameters $\mathbb{S}_2$ is determined using the BEA algorithm,

- if the product of the set $\mathbb{S}_1$ and $\mathbb{S}_2$ is non-empty, a new set is generated: $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$. The new set $\mathbb{S}$ is used to check the consistency of the remaining data points with already tested ones. If the product of the discussed sets is empty, the algorithm is stopped because the current data point is not consistent with the previous data points, thus it cannot compose an approximately linear dependency.
- the procedure is repeated for the remaining data points until the conditions presented above are satisfied.

The sample four steps of the procedure are presented in Fig. 4.3.

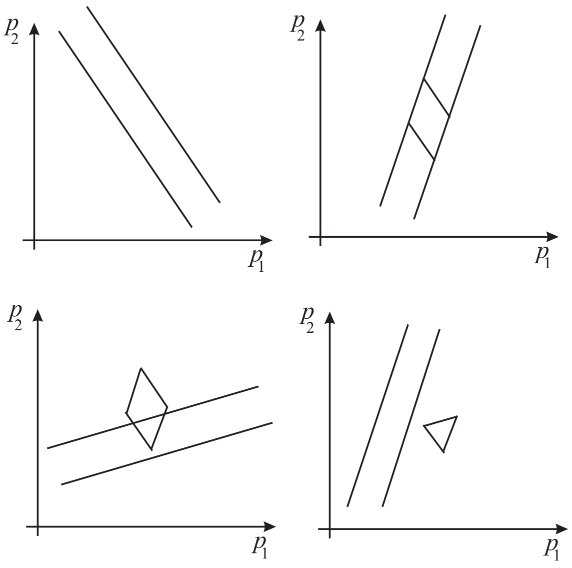

Fig. 4.3. Set $\mathbb{S}$ in the successive iterations of the algorithm

The presented algorithm can be used to detect a single linear dependency starting from the beginning of system characteristics. In order to determine the N-F model structure it is required to detect all approximately linear dependencies within system characteristics. For this purpose a modified algorithm, which is based on the discussed approach, has been developed (Fig. 4.4). The following notation is introduced in the presented scheme: $\boldsymbol{X}$ – sorted set of data points, $\boldsymbol{L}$ – set of data points which form an approximately linear dependency, s – single data point tested for consistency with data points from the set $\boldsymbol{L}$, $\boldsymbol{S}$ – feasible set of parameters determined for the data point s, $\boldsymbol{P}$ – polytope which determines the feasible set of parameters for data points from the set $\boldsymbol{L}$, e – acceptable error of the partial model.

In the first step of the algorithm all data points are sorted with respect to the defined starting point. Next, a hypercube large enough is defined in the parameter space, which is used to test the consistency of data points. The error e is chosen

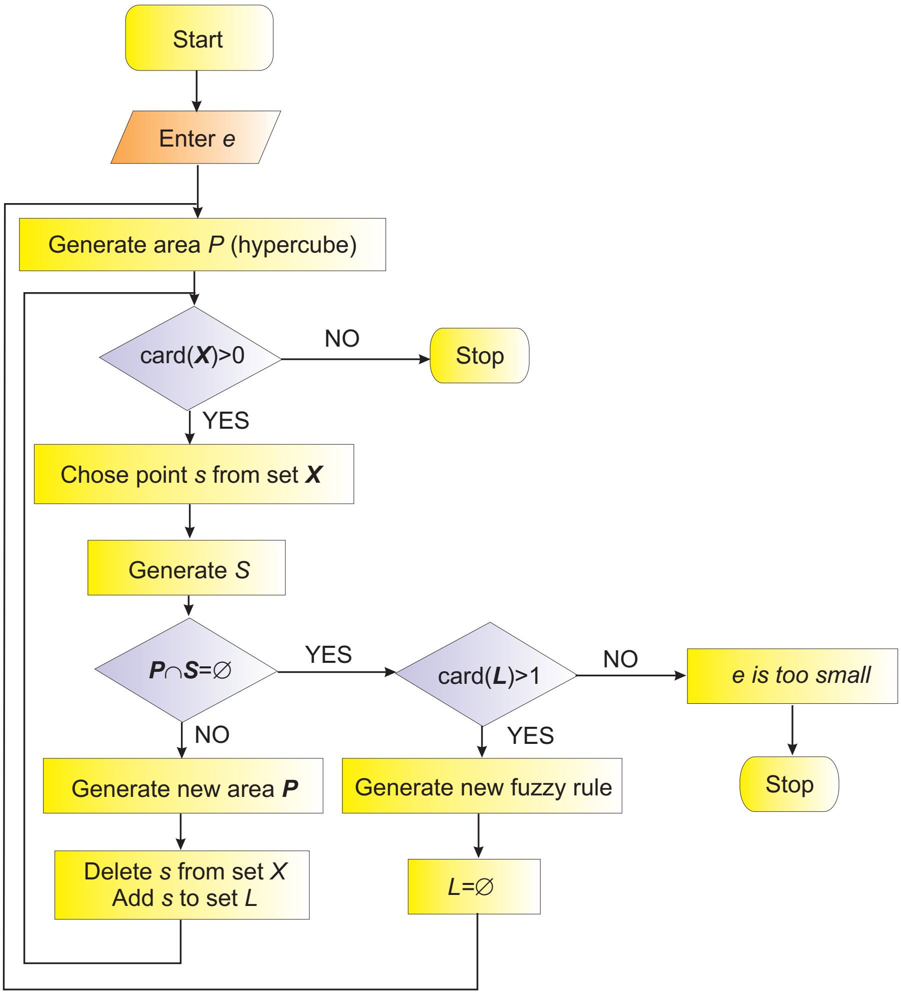

Fig. 4.4. Algorithm for rule base declaration

by the designer to ensure the accuracy of all partial models. The successive data points are drawn from the sorted table, which stores all data points. The main idea of the approach is to compute for each data point two hyperplanes which determine the feasible set of parameters. Such information is next used to decide if the chosen data points form with the previously tested points an approximately linear dependency. If the data point is not consistent with the previously tested points with the assumed error e, the search procedure is stopped and the fuzzy rule is generated using data points which are consistent with each other. The point which interrupts the procedure becomes the starting point and the whole procedure is repeated in order to find the next approximately linear dependency. The algorithm is repeated until the table with the available measurements is empty.

It is also possible that the algorithm stops but the table with measurements is not empty. In such a situation the whole procedure must be repeated for a bigger value of e because at least one data point is not consistent with the other data points. The algorithm is able to detect such problems and inform about them the designer, who must change the initial configuration of the algorithm. Finally, the results generated by the algorithm are used to declare the rule base, which next is used to build the N-F model. In order to illustrate the functioning of the algorithm the example 4.1 is presented.

Example 4.1

The following function is given:

$$y = \begin{cases} x+1 & \text{for} \quad x \in [0;5], \\ -x+11 & \text{for} \quad x \in (5;10], \\ 0.5x-4 & \text{for} \quad x \in (10;15]. \end{cases}$$

Measurements are disturbed by the uniformly distributed $U(0,1)$ random variable. The acceptable level of error defined for partial models has the value $\varepsilon = 0.4$. The algorithm detects three approximately linear dependencies for such an initial configuration. Based on this, three fuzzy rules are generated and then used to build the N-F model. The results obtained for the designed N-F model are presented in Fig. 4.5.

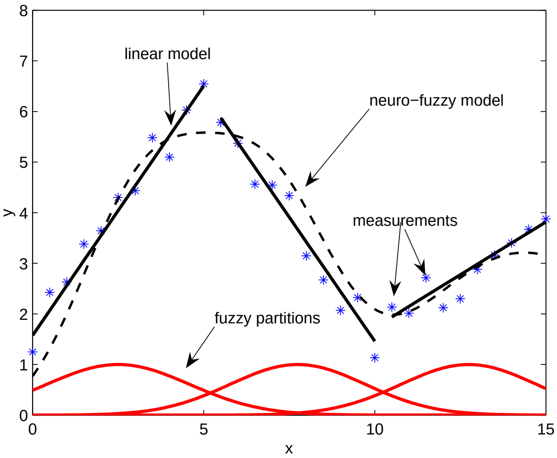

Fig. 4.5 . Detection of approximately linear dependencies

4.5. Summary

The problem of structure selection is a very important step in the context of the N-F model design procedure. Unfortunately, there do not exist algorithms which would be able to choose an optimal structure of the N-F model. The work concentrates on the problem of rule selection, which should ensure the desired accuracy of the N-F model. Taking into account diagnostic tasks, an algorithm which helps the designer to choose a suitable structure of the N-F model was formulated. It is based on the BEA algorithm and its main task is to determine the number of approximately linear dependencies present in the characteristics of the system. Next, for each found local linear dependency a single fuzzy rule is generated. Moreover, the procedure is able to determine the spacing of fuzzy sets which represent antecedents, and the BEA algorithm is used to estimate the parameters of partial models which represent linear consequents of rules. The main problem of the algorithm was to define the concept of the approximately linear dependency. This is done by introducing the maximal error for the linear partial model, which is used to evaluate the quality of linear approximation. Next, data points are tested in order to check if they form an approximately linear dependency using the defined error. The procedure is repeated for all available measurements. However, it must be pointed out that the procedure can end unsuccessfully if at least one data point is not associated with an approximately linear dependency. It is an effect of the situation when the designer defines too small a value of the error ε. In this case the whole procedure should be repeated for a larger value of the error ε. The main task of the designer is to find a trade-off between model accuracy and complexity by choosing a proper value of the error ε. The developed algorithm has one main advantage in relation to alternative approaches, i.e. clustering algorithms, because only one parameter ε must be defined to start the algorithm. Moreover, the value of this parameter has a direct impact on the accuracy of the obtained model. In the case of clustering algorithms usually few initial parameters must be defined, thus the designer should have a deep knowledge about and experience in using such algorithms. An extra advantage of the proposed algorithm is the fact that the parameters of partial models can be directly obtained form the results generated by the method. If such parameters do not guarantee the required accuracy of the global model, then simple tuning of these parameters usually makes it possible to meet the required global accuracy.

Chapter 5

APPLICATION OF NEURO-FUZZY NETWORKS TO DIAGNOSTICS

5.1. Introduction

The reliability, safety and availability of industrial plants play an important role in operational use. It is important especially nowadays when industrial installations and control algorithms are becoming more and more sophisticated, and economy presses us to reduce the downtime of plants and to shorten the time necessary to create a product. Simple technical systems can be inspected by the human expert but complex industrial systems require automated diagnostics in order to determine the location of and reason for the fault fast and precisely. (Calado *et al.*, 2003; Kowal and Korbicz, 2000b; Kowal and Korbicz, 2000c; Kowal, 2000; Kościelny, 2001). The model-based fault detection strategy is the subject of intensive research in the area of diagnostics due to many important properties:

- the ability to detect small-scale faults,
- the solution is relatively cheap because sophisticated equipment is not required; suitable software and computer is usually enough,
- the installation of the fault diagnosis system usually does not require intervention in the existed system; the installed sensors can usually be used for data acquisition.

The method is based on residual generation by comparing the estimates of the measured signals with their originals. Then the residuals are used to detect and isolate faults. The most common approach to form residuals is based on the difference between the estimate and the original signal, thus prompt fault detection requires accurate models of processes and leads directly to the problem of system identification. Real processes are usually dynamic, non-linear and stochastic, and analytical approaches to identification are not suitable for them. An alternative approach proposes to use artificial intelligence methods like neural networks, fuzzy systems, neuro-fuzzy systems and expert systems (Korbicz *et al.*, 2004; Obuchowicz, 2003; Chan *et al.*, 1999; Chen and Linkens, 2001). The present work focuses on N-F networks. The attractiveness of N-F methods arises from the fact that they can be employed when there are no phenomenological models available. In such a case, N-F models can be identified using simultaneously quantitative and qualitative knowledge, i.e. a human expert may code his/her knowledge in the form of fuzzy rules, which are then introduced into the N-F system (Rutkowska, 2003; Rutkowska, 1997). The N-F model can be also identified using the available data and methods of learning known for neural networks, e.g. gradient descent methods. Two types of N-F networks are commonly used for the modelling purpose: the Mamdani N-F network and the Takagi-Sugeno N-F network. Generally, Takagi-Sugeno structures have better performance qualities in modelling than other structures due to their possibility to decompose non-linear systems into a collection of local linear models.

In real situations, regardless of the kind of identification method used, there is always a model-reality mismatch, which arises usually from wrong assumptions about the structure of the model or the type of disturbances which corrupt measurements. The uncertainty of the model can dramatically decrease the reliability of fault detection, thus robust fault detection systems under model uncertainty are required (Dinca *et al.*, 1999). Two main approaches can be used to overcome the problem: the active approach, which is based on robust observers, and the passive approach, which is based on the adaptive threshold technique (Patton and Chen, 1999). In the present work the adaptive threshold technique is employed to implement a robust model-based fault detection system. This technique is based on knowledge about the uncertainty of the model. Unfortunately, there do not exist effective methods that would allow determining the uncertainty of non-linear systems. The existing methods like the statistical approach or the BEA method can be effectively applied only to linear systems. Several different approaches for neural models were proposed in order to overcome the problem. Linearization around the working point and applying the BEA algorithm to calculate uncertainty were proposed in (Witczak and Mrugalski, 2003). An interesting approach, which does not require linearization, was proposed in (Mrugalski, 2003), where GMDH networks are used to build robust model-based fault detection systems.

The Takagi-Sugeno N-F network is a suitable method for building models for fault detection systems due to many positive properties mentioned above. Unfortunately, there do not exist methods that would allow estimating the uncertainty of such a model in terms of quantity. Therefore, the application of N-F models was limited because it was hard to assure robustness against model uncertainty, which in the case of fault detection is very important. The present work proposes a solution to this problem in the form of an algorithm that uses the BEA or the OBE algorithm to calculate the uncertainty of the Takagi-Sugeno N-F model. A robust fault detection system is obtained using knowledge about N-F model uncertainty to calculate the adaptive threshold.

The present chapter presents fundamental information about model-based fault detection, and then the adaptive threshold technique is described. In order to prove the effectiveness of the developed algorithms, they are applied to fault detection of a valve that is part of a technical installation in the Lublin sugar factory, and to fault detection of a laboratory electrical engine. The experimental results are presented and described in the final part of the chapter.

5.2. Diagnostics of technical processes

The main aim of diagnostic systems is to evaluate the state of the technical system or its process of exploitation in order to investigate the model-reality mismatch. The first approach is based on the inspection of the entire process and the second one assumes separate inspection of individual components of the process. Four main groups of elements can be distinguished in industrial processes: controllers, actuators, sensors and other plants of the installation. Specific sorts of faults are associated with every group of elements. Besides faults, the fault detection system must deal with the problem of disturbances that corrupt measurements, and model uncertainty. However, these problems are not a direct danger for the process, although they can lead to incorrect detections and false alarms. The main property of a correct fault detection system is being sensitive to faults and robust against disturbances as much as possible.

The fault diagnosis task can be divided into three steps (Fig. 5.1): detection, localization and identification (Isermann and Ball, 1997; Korbicz *et al.*, 2004). Fault detection is to signal the fact of fault appearance in the process. The main task of localization is to determine the place and time of fault appearance, whereas the identification procedure is used to evaluate the size, character of and reason for the fault. The fault detection step is responsible for generating symptoms which describe the fault. This procedure is very important because only correct symptoms permit appropriate localization and identification. Model-based fault detection is used to generate the residual signal, which indicates the state of the process (Fig. 5.2). The residual evaluation step covers the conversion of quantitative residual signals into qualitative diagnostic signals, while simultaneously the decision about fault appearance must be made. Residuals are usually defined in the form of the

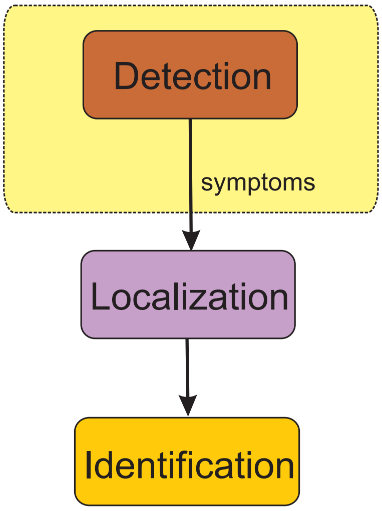

Fig. 5.1. Three steps of the fault diagnosis procedure

difference between the output of the process $y'(k)$ and the output of the model $y(k)$:

$$e_r(k) = y'(k) - y(k). \tag{5.1}$$

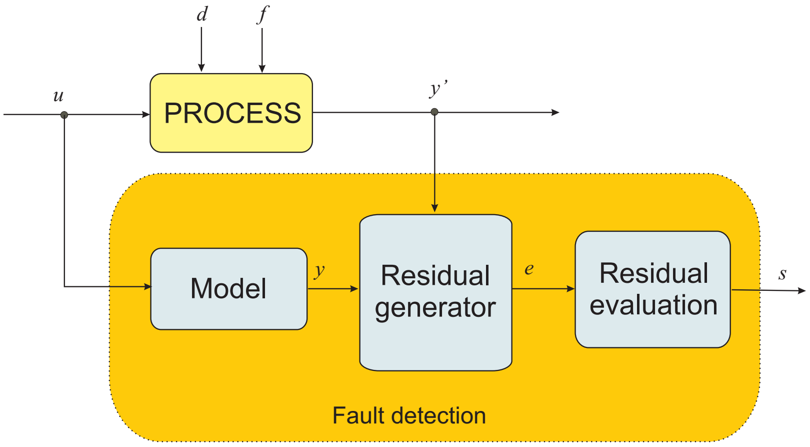

Fig. 5.2. Model-based fault detection

The residual signal should be close to 0 for the fault-free mode and considerably different from 0 in the case of the faulty mode. Therefore, it is very important to build an accurate model of the process in order to avoid false alarms generated by residuals different from 0 although the process works in nominal conditions. Unfortunately, in reality, models are uncertain, so model-based fault detection systems can work incorrectly. For this reason robustness against model uncertainty is a very advisable property for the fault detection system. Such robustness can be obtained by using the adaptive threshold technique for the residual signal.

5.3. Fault detection with the adaptive threshold method

The main task of the fault detection system is to signal the fault and generate the symptoms of the fault. The fault detection procedure using the model-based scheme reduces to residual calculation and evaluation. The residual evaluation procedure is critical for correct fault detection because it is responsible for alarm activation. In order to avoid false alarms, thresholds are used to define the interval which determines the values of residuals that correspond to the fault-free mode. The evaluation algorithm is based on the fact that residual signals exceed the defined interval in the case of a fault and should be inside the interval in the fault-free mode.

The residual signal is affected by model uncertainty and disturbances. The uncertainty of the residual signal can change with time but the constant values of thresholds must take into account the worst scenario with the highest possible uncertainty. This usually produces unacceptably big values of thresholds, thus the sensitivity of detection is too low. Of course, thresholds can be attenuated in order to increase sensitivity but such a solution leads to false alarms, and the reliability of the fault detection system is lost. The adaptive threshold technique can be used to overcome this problem (Frank and Ding, 1997; Seliger and Frank, 2000). The adaptive approach proposes to calculate thresholds online for each moment in time and the values of the thresholds depend on model uncertainty and disturbances at the moment. The technique of the constant threshold is presented schematically in Fig. 5.3. The fault is detected at the moment 295 for the constant threshold. The speed of detection can be improved by using lower thresholds, but in this case false alarms appear in the interval $0-200$. The adaptive threshold allows detecting the fault at the moment 200 and false alarms are not generated, so the method is robust against disturbances. Correct fault detection is also very important in the context of fault localization and identification. In the diagnostic system considered, the model is implemented using the Takagi Sugeno N-F network. The uncertainty of such a model must be calculated in order to determine the value of the adaptive threshold. Uncertainty is represented by the interval defined for the output of the N-F model and the interval is calculated using the approach developed in Section 3. The Takagi-Sugeno N-F network must be viewed in the form of the LP system according to the assumptions established for the developed algorithm:

$$y(k) = \boldsymbol{x}^T(k)\boldsymbol{p}, \tag{5.2}$$

where the vector $\boldsymbol{x}(k)$ includes all input variables defined for all linear consequences and multiplied by firing levels of fuzzy rules.

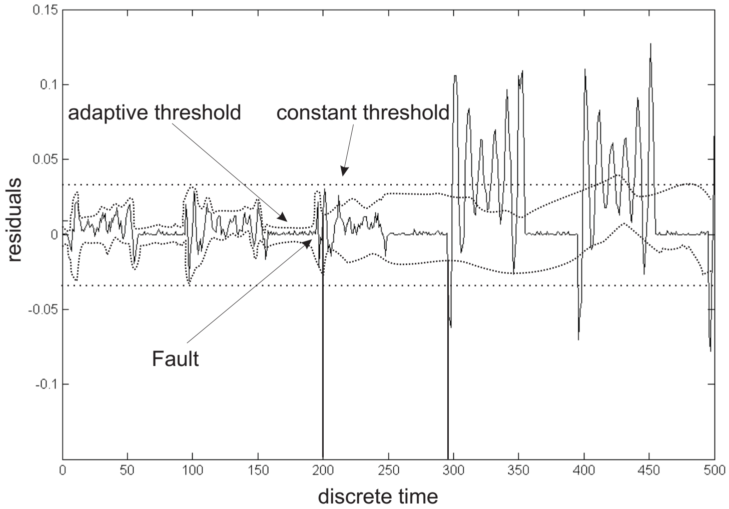

Fig. 5.3. Constant and adaptive threshold

Let us assume that $\mu_i(\boldsymbol{r}^{fs}(k), \boldsymbol{p}_i^{fs})$ define the firing level of the ith rule, and ϕ_i is equal to the value $\mu_i(\boldsymbol{r}^{fs}(k), \boldsymbol{p}_i^{fs})$ divided by the sum of firing levels for all rules:

$$\phi_i = \frac{\mu_i(\boldsymbol{r}^{fs}(k), \boldsymbol{p}_i^{fs})}{\sum_{j=1}^{N} \mu_j(\boldsymbol{r}^{fs}(k), \boldsymbol{p}_j^{fs})}, \tag{5.3}$$

where $\boldsymbol{r}^{fs}(k)$ is a vector of inputs for the first layer of the N-F network , and $\boldsymbol{p}_i^{fs}$ is a set of parameters that describe the antecedent of the ith fuzzy rule. In this case the vector $\boldsymbol{x}(k)$ has the following form:

$$\boldsymbol{x}(k) = \begin{bmatrix} \phi_1 r_1(k) \\ \vdots \\ \phi_1 r_n(k) \\ \phi_2 r_1(k) \\ \vdots \\ \phi_2 r_n(k) \\ \vdots \\ \phi_N r_1(k) \\ \vdots \\ \phi_N r_n(k) \end{bmatrix},$$

where $\boldsymbol{r}(k) = [r_1(k), r_2(k), \ldots, r_n(k)]^T$ is a vector of input variables defined for linear models, which represent the consequences of fuzzy rules. If the residual signal is defined by means of the expression

$$e_r(k) = y'(k) - y(k), \tag{5.4}$$

and the confidence interval determined for $y'(k)$ using the OBE algorithm is given by the following inequalities:

$$\boldsymbol{x}^T(k)\boldsymbol{p}^{min}(k) + \varepsilon^{min}(k) \leqslant y'(k) \leqslant \boldsymbol{x}^T(k)\boldsymbol{p}^{max}(k) + \varepsilon^{max}(k), \tag{5.5}$$

then the residual $e_r(k)$ must satisfy the following inequalities:

$$\boldsymbol{x}^T(k)\boldsymbol{p}^{min}(k) + \varepsilon^{min}(k) - y(k) \leqslant e_r(k) \leqslant \boldsymbol{x}^T(k)\boldsymbol{p}^{max}(k) + \varepsilon^{max}(k) - y(k). \tag{5.6}$$

A sample fault detection procedure using the approach with the adaptive threshold is shown in Fig. 5.4. The situation when the residual signal exceeds the range defined by thresholds is interpreted as a fault and the alarm is activated.

In the case of the confidence interval calculated for the N-F model using the OBE algorithm, the inequalities 5.6 are converted to the following form:

$$\boldsymbol{x}^T(k)\hat{\boldsymbol{p}} - \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)} \leqslant \boldsymbol{x}^T(k)\boldsymbol{p} \leqslant \boldsymbol{x}^T(k)\hat{\boldsymbol{p}} + \sqrt{\boldsymbol{x}^T(k)\boldsymbol{M}\boldsymbol{x}(k)}, \tag{5.7}$$

where $\boldsymbol{M}$ is a matrix that describes the orientation of the ellipsoid, and $\hat{\boldsymbol{p}}$ defines the center of the ellipsoid.

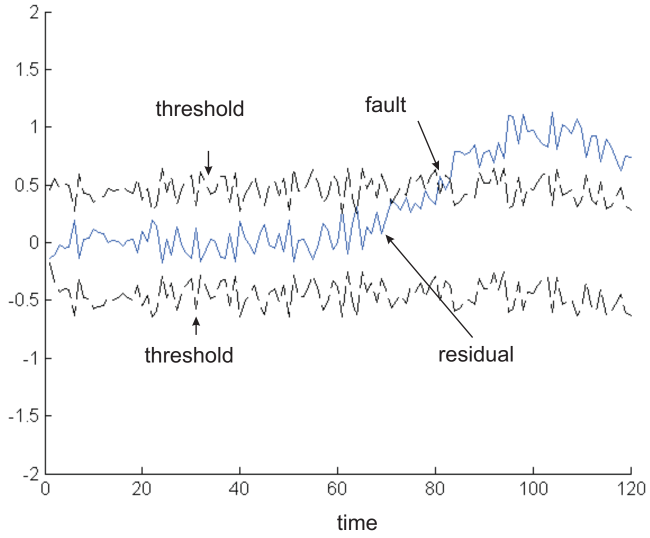

Fig. 5.4. Adaptive threshold for the residual signal

The adaptive threshold for models with uncertain input variables is calculated analogously to the method shown above. The only difference appears in the point concerning the calculation of the confidence interval. For the OBE method, Eqn. (3.108) is used for this purpose, and for the BEA method, Eqn. (3.66) is used.

5.4. Fault detection system for an electrical engine

Electrical DC (Direct Current) and AC (Alternating Current) engines are very often used in many industrial applications (Leonhard, 1996). The changing conditions of operation and intensive exploitation result in systematic wearing off of individual parts of engines. This phenomenon can be interpreted as an incipient fault, which in the final phase changes to an abrupt fault and causes big damages in the engine. It is very important in this case to detect the fault at an early stage and apply a special procedure to avoid the fault so that the worn off elements can be replaced. The faults considered manifest themselves at an early stage by a decreased efficiency, but finally, if the fault is not detected some parts of the engine can be damaged. Thus it is important to develop a reliable fault detection algorithm which should detect even small changes in system behavior. The traditional methods of monitoring engines require a direct inspection, so the whole process must be stopped for the time of the inspection. Such an approach is usually time consuming and causes financial loses for the company. Another approach uses online identification of engine parameters. Fault detection in this case is done by monitoring the values of the parameters. Unfortunately, this method requires a detailed mathematical model of the engine and the expert that interprets the results must have a deep knowledge about the engine and experience in evaluating the behavior of the engine.

An interesting alternative for the above approaches is to use soft computing methods to build a model-based fault detection system (Liang *et al.*, 2002; Gao and Ovaska, 2001). N-F models seem to be a good choice to build the model of the engine due to the possibility of using heuristic knowledge gathered from the operator and measurements, which can be obtained during the normal operation of the engine. Such an approach allows building accurate models, which can be used to detect incipient faults. However, it is still possible to improve the effectiveness and reliability of fault detection systems, i.e. the methods developed in this work concerning model uncertainty and adaptive thresholds can help to diagnose incipient faults more effectively.

The effectiveness of the robust fault detection method using the Takagi-Sugeno N-F network and adaptive thresholds has been examined using a laboratory stand of the Institute of Control and Computation Engineering of the University of Zielona Góra – *AMIRA*. The laboratory stand can be used to control the rotational speed of a DC engine with a changing load (Fig. 5.5). The main part of the stand is the DC engine, and during the experiments the engine was used as the diagnosed object.

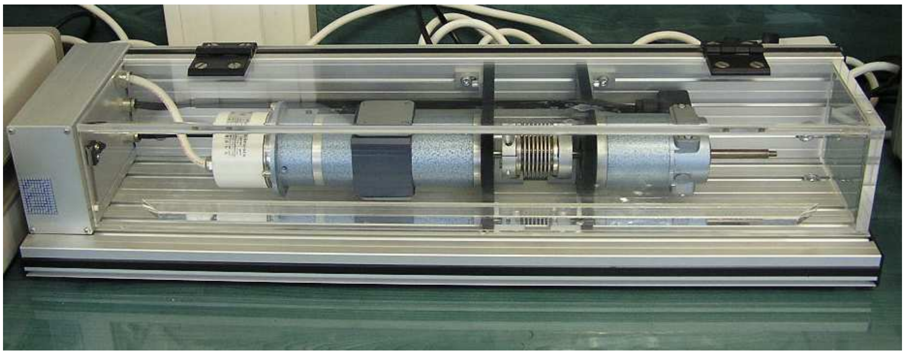

Fig. 5.5. Laboratory stand with a DC engine

5.4.1. System specification

The laboratory object considered consists of five main elements:

- DC engine M_1,
- DC engine M_2,
- two engine-speed indicators,
- clutch K.

The input signal for the engine M_1 is an armature current. Its value is determined by a cascade control loop. The servo-amplifier of the current controller works in four modes, thus the current direction can be chosen appropriately according to the demanded direction of rotation. The output of the object is the rotational speed of the engine. The rotational speed can be measured using two sensors: a tachometer or an optical sensor, which generates impulses that correspond to the rotations of the engine. The shaft of the engine M_1 is connected with the identical engine M_2 by the clutch K. The engine M_2 works in the generator mode and the generated current is controlled by another controller. The basic technical data concerning the laboratory system are shown in Table 5.1.

Table 5.1. Laboratory system technical data

variable	value
Engine	
rated voltage	24 V
rated current	2 A
rated power	30 W
rated speed	3000 ob/min
rated moment	0.096 Nm
moment of inertia	$17.7 * 10^{-6}$ Kgm^2
resistance	3.13 Ω
Tachometer	
output voltage	$5 \frac{\mathrm{mV}}{\mathrm{ob/min}}$
moment of inertia	$10.6 * 10^{-6}$ Kgm^2
Clutch	
moment of inertia	$33.0 * 10^{-6}$ Kgm^2

The engine M_1 is controlled using the servo-amplifier, where the control signal has the form of the voltage from the range -10V–+10V with the amplification 0.4 A/V. The engine M_2 is also controlled using the the servo-amplifier, where the control signal has the form of the voltage from the range -10V–+10V with the amplification 0.237 A/V. The tachometer serves to measure indirectly the rotational speed of the engine M_1. The output range of the tachometer is -10V–+10V.

5.4.2. Fault detection using the Takagi-Sugeno N-F model

The model of the engine M_1 must be designed in order to build a fault detection system for the engine. The engine considered is a dynamic object, thus dynamics have to be included in the N-F network in the form of dynamic consequences. Experiments with different structures of dynamic consequences show that the best results can be obtained using the following linear consequences for fuzzy rules:

$$y_i(k) = a_1 y_i(k-1) + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3) + b_4 u(k-4) + b_0, \quad (5.8)$$

where $y_i(k)$ is the output of the ith rule, which should be interpreted as the rotational speed. The input variable $u(k)$ is a voltage responsible for controlling the rotational speed. The N-F model built has only one global input variable $u(k)$.

In order to identify the structure of the N-F model and its parameters, the input-output data were generated using the prepared input signal. The experiment was done using an open-loop control scheme. The obtained measurements are presented in Fig. 5.6. All variables were normalized to the range -1–+1.

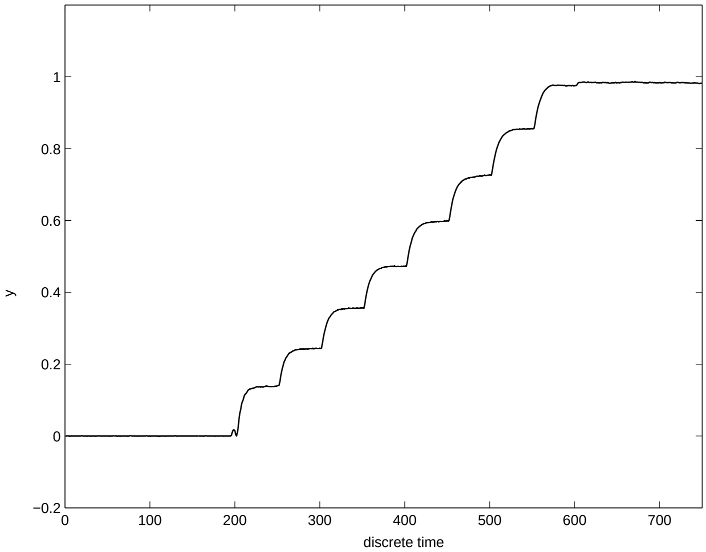

Fig. 5.6 . Measurements: open loop control

The generated data were used to detect the structure of the Takagi-Sugeno N-F model using the algorithm developed in Section 4. It was assumed that a single linear model that describes the consequence of the fuzzy rule can produce the maximum error on the level 0.04. For such a value of the error the algorithm generated nine fuzzy rules, which were then included in the N-F model. The behavior of the model was tested using the data that was not used during the design procedure. The model was tested in the open-loop control environment (Fig. 5.7) and in the closed-loop control environment (Fig. 5.8). A set of potential faults is defined for the engine considered. The set includes seven faults, and information about faults is summarized in Table 5.2. The faults were simulated artificially us-

Table 5.2. Types of faults

No	Description	S	M	B	I
f_1	Tachometer fault	●	●	●	●
f_2	Mechanical fault of the engine	●	●		●

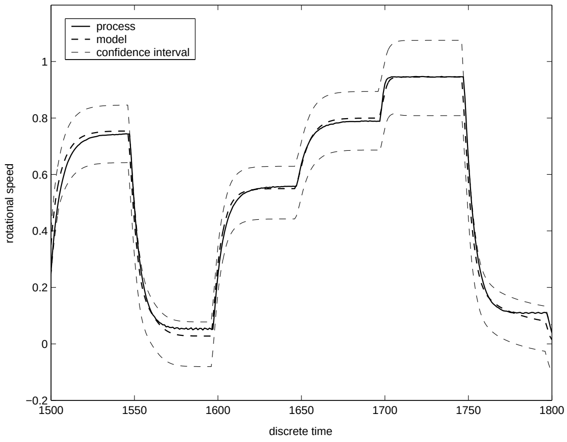

Fig. 5.7. Test: open loop control

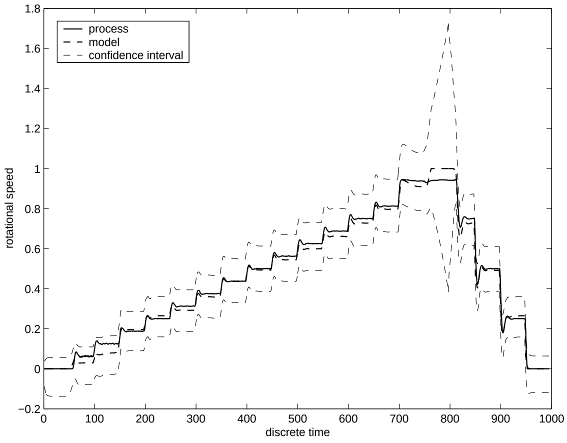

Fig. 5.8. Test: closed loop control

ing the elements of the laboratory system. It was impossible to generate real faults in the laboratory environment. The faults are divided into two groups: tachometer faults and mechanical faults of the engine M_1, which manifest themselves as a decreasing efficiency of the engine. Tachometer faults were simulated by disturbing its output signal using different types of noise. Such disturbed samples given by the tachometer were used to calculate the control signal in the closed-loop control scheme. In order to generate the second fault, the engine M_2 connected with the engine M_1 via the clutch K was used to simulate an additional faulty load. The aim of such an approach was to simulate the incipient mechanical fault in the engine M_1, i.e. a worn-off bearing. It was assumed that faults can be incipient (I) or abrupt, and abrupt faults are divided into small (S), medium (M) and big (B)

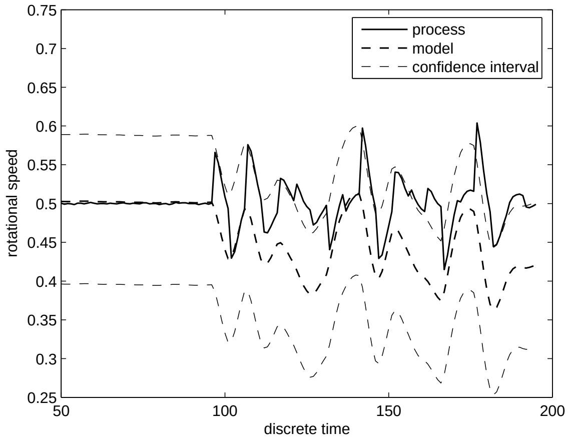

Fig. 5.9. Process and model output: small fault f_1

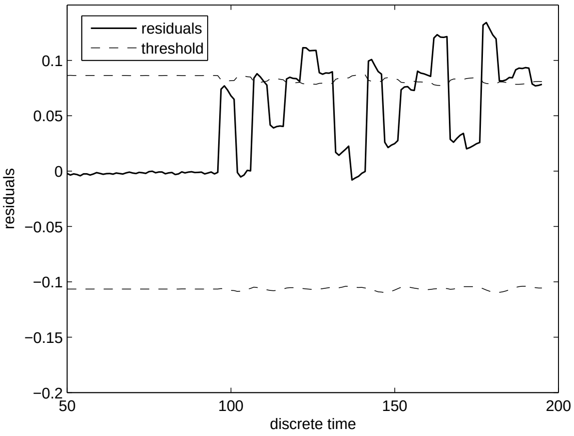

Fig. 5.10. Residuals: small fault f_1

faults. The effectiveness of the designed fault detection system was tested using data generated during fault simulations. Faulty data were prepared for all designed scenarios. The sample results, which present the output of the real object, model output and residuals, are presented in the figures. Sampling time equal to 0.1 s is used for all figures. The data for a small fault of the tachometer is presented in Fig. 5.9 and Fig. 5.10. The fault is simulated at the moment 90, and is manifested as small random noise that corrupts the tachometer output. The fault is detected at the moment 115 because the residual signal exceeds the interval determined by adaptive thresholds. It is important to detect not only abrupt, big faults, but

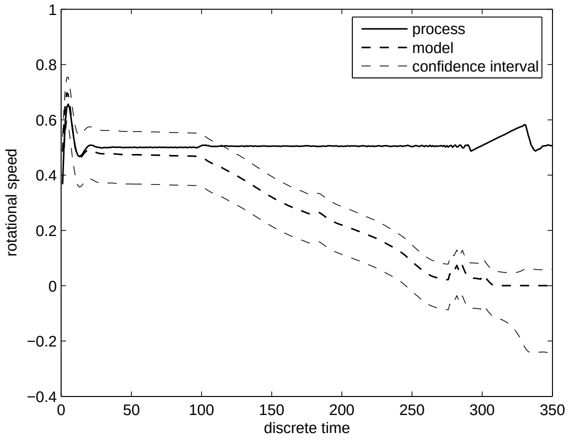

Fig. 5.11 . Process and model output: incipient fault f_1

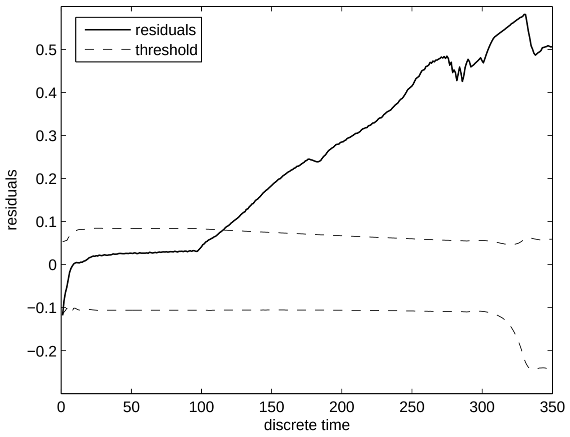

Fig. 5.12 . Residuals: incipient fault f_1

also incipient fault from the point of view of effective diagnostics. Such faults are caused by the slow and progressive process of the wearing off of the parts of the engine so that faults are incipient and their scale is increasing with time, thus it is hard to detect them at an early stage. In order to illustrate the effectiveness of the developed methods for fault detection of incipient faults, experimental results obtained for the incipient fault f_1 are presented in Fig. 5.11 and Fig. 5.12. Fault detection of the incipient fault f_2 is illustrated by the data presented in Fig. 5.13 and Fig. 5.14. Both simulated faults were detected after 2 s and 2.5 s, respectively. The fault detection system built is able to detect all simulated faults. The speed of fault detection is different for different faults only. Fault detection of incipient faults is a fundamental ability of the designed system because it keeps to minimum losses caused by damages. The essential speed-up of fault detection was possible due to the use of the adaptive threshold, which is based on knowledge about model uncertainty. The conducted experiments allow claiming that the developed method can be applied to other industrial plants and electric drives.

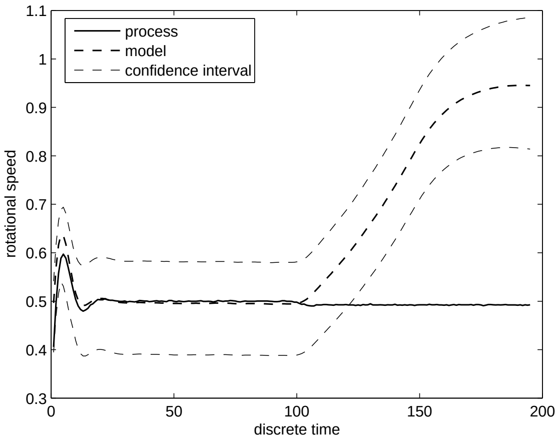

Fig. 5.13 . Process and model output: incipient fault f_2

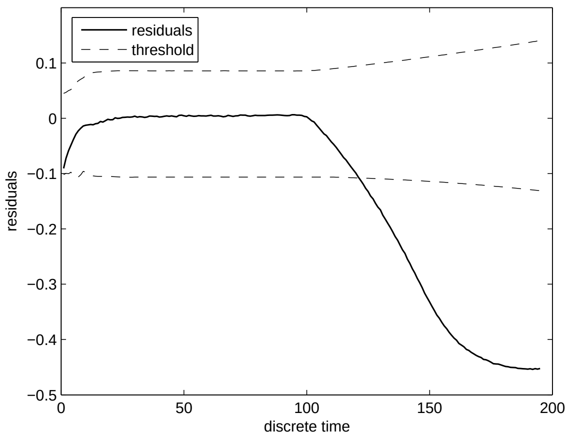

Fig. 5.14 . Residuals: incipient fault f_2

5.5. Fault detection system for the valve

An element of the evaporation station of the Lublin sugar factory is the second object to which the proposed methods of fault detection were applied (Kościelny *et al.*, 2004). The analyzed benchmark for testing different diagnostic algorithms was created through the cooperation of the Lublin sugar factory and the international training network *Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS)*. The main aim of the DAMADICS (2002-2004) project was to develop a diagnostic system for modern actuators. A special simulator for testing the effectiveness and robustness of different diagnostic procedures was developed in the framework of the project. Moreover, real data from the sugar factory were available. The detailed information about the object, simulator and measurements is available on the website of the project (DAMADICS, 2002).

5.5.1. System specification

The evaporation station consists of seven inter-connected evaporators. The main technological task of the evaporation station is to thicken sugar-beet juice. Evaporation stations are grouped into five sections (Sections I,VI,V consist of one evaporator each and Sections II and III consist of two evaporators each). The first five evaporators work with natural juice circulation and the last two have a different construction and work with juice circulation forced by pumps. The juice condensation process is performed using steam and vapour, which are the same quantities from the physical point of view but have different sources. Steam is produced by the water-steam-boiler and is mainly delivered to Section I of the evaporation station, whereas vapour, as a recyclable medium, is produced in each evaporator and the heat accumulator. Vapour is used as a heating medium in other technological stations. An additional task of the evaporation station is to produce condensate, which is then delivered to the steam boiler. The process of evaporation is controlled by many different control loops, i.e. temperature control, pressure control, juice level control. The juice level control is implemented for each evaporation station and the level is actuated using the valve, which is placed at the front of each evaporation station.

The main subsystems of the evaporation process are evaporation stations. A scheme of the 7th evaporation station is presented in Fig. 5.15. The essential influence on the operation of the station comes from the valve FC57_03, presented in Fig. 5.16. The valve works as an actuator and controls the juice outflow from the evaporation station.

Actuators widely widespread in industry are installed usually in a harsh environment, i.e. they are affected by high temperature, pressure, humidity, pollution, chemical solvents, aggressive media, vibrations etc., so their liftime can be significantly shorter. The malfunction or failure of actuators cause long-term process disturbances or sometimes even force the installation shutdown. Moreover, actuators are usually affected by incipient faults, which change process behavior slowly and may stay undetected for a long time. Therefore, sensitive fault detection sys-

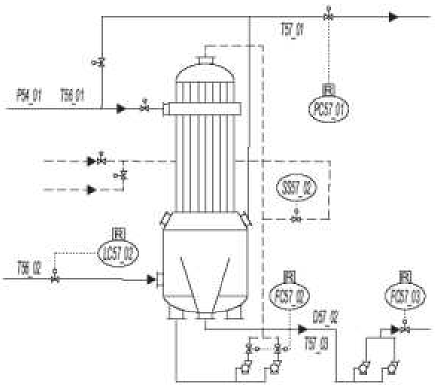

Fig. 5.15. Evaporation station

Fig. 5.16. Valve actuator

tems are demanded in such cases in order to detect faults early so that actuators can be repaired and damages of other components can be avoided.

A detailed scheme of the chosen actuator is shown in Fig. 5.17. The plant consists of a control valve, a spring-and-diaphragm pneumatic servomotor and a positioner. The following notations are introduced for the scheme: Z_1, Z_2, Z_3 – bypass valves, ACQ – data acquisition unit, CPU – positioner central processing unit, F/E – electro-pneumatic transducer, PC – pressure transducer, PP – displacement transducer, PS – volume flow rate transducer. A list of variables measured for the process (Fig. 5.17) is presented in Table 5.3. The set of potential faults was defined for the actuator considered. The set consists of 19 different faults, and information about them is presented in Table 5.4.

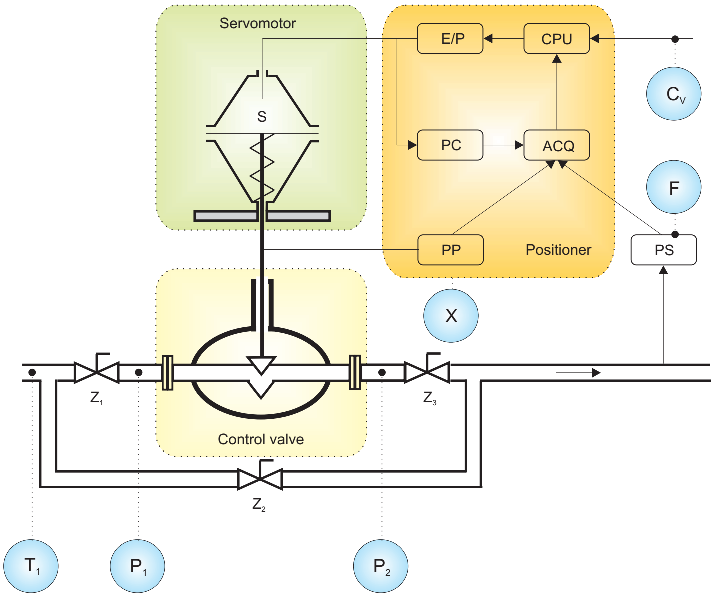

Fig. 5.17. Scheme of the valve actuator

Table 5.3. Variables measured for the actuator

Name	Symbol	Description	Range	Unit
F	FC57_03	Flow	0-100	m^3/h
X	FC57_03X	Servomotor rod displacement	0-100	%
C_V	57_03X	Control value	0-100	%
T_1	57_03X	Juice temperature	0-150	C
P_1	P57_03	Juice pressure (inlet)	0-1000	kPa
P_2	P57_04	Juice pressure (outlet)	0-1000	kPa

In the framework of the DAMADICS project some experiments were conducted using a real valve from the technical installation of the Lublin sugar factory. The measurements were collected during 19 days in Novemeber 2002. The faults were artificially simulated using the available variables in order to generate data for faulty scenarios. Data for four faulty scenarios, f_{16}, f_{17}, f_{18} and f_{19}, were obtained.

Data for only few faults were generated during the above-mentioned experiments. In order to test the fault detection system, it is necessary to obtain data for all faults indicated in Table 5.4. Nevertheless, it is impossible to simulate all possible faults in a real environment for safety reasons. In order to overcome this problem, a simulator of the actuator (Fig. 5.17) was created in the framework of the DAMADICS project using the *MATLAB Simulink* program. The designed simulator allows simulating all processes going on in the actuator and makes it possible

Table 5.4. Types of faults

No.	Description	S	M	B	I
	Control valve faults				
f_1	Valve clogging	●	●	●	
f_2	Valve plug or valve seat sedimentation			●	●
f_3	Valve plug or valve seat erosion				●
f_4	Increased of valve or bushing friction				●
f_5	External leakage				●
f_6	Internal leakage				●
f_7	Medium evaporation or critical flow	●	●	●	
	Pneumatic servo-motor faults				
f_8	Twisted servo-motor's piston rod	●	●	●	
f_9	Servo-motor's housing or terminals tightness				●
f_{10}	Servo-motor's diaphragm perforation	●	●	●	
f_{11}	Servo-motor's spring fault			●	●
	Positioner faults				
f_{12}	Electro-pneumatic transducer fault	●	●	●	
f_{13}	Rod displacement sensor fault	●	●	●	●
f_{14}	Pressure sensor fault	●	●	●	
f_{15}	Positioner feedback fault			●	
	General faults				
f_{16}	Positioner supply presser drop	●	●	●	
f_{17}	Unexpected pressure change across the valve			●	●
f_{18}	Fully or partly opened bypass valves	●	●	●	●
f_{19}	Flow rate sensor fault	●	●	●	

to simulate all the defined faulty scenarios. Additionally, it is possible to define the scale of the fault and its speed. It is assumed that faults can be abrupt (A) and incipient (I). Abrupt faults are divided into three groups: small (S), medium (M), and big (B) faults. Taking into account those facts, it is possible to generate 44 different faulty scenarios using the *MATLAB Simulink* environment.

5.5.2. Fault detection using the N-F model

The fault detection system was prepared for the described actuator in order to verify the effectiveness of the developed methods for designing N-F models for fault detection. The proposed algorithm detects faults by residual signal evaluation, and the residuals are generated as a difference between the object and the Takagi-Sugeno N-F model. The analysis of the evaporation process and expert knowledge allows formulating two relationships between variables in the actuator considered (Edelmayer, 2000; Kościelny *et al.*, 2000):

$$F = f_F(X, P_1, P_2, T_1), \tag{5.9}$$

$$X = f_X(C_V, P_1, P_2, T_1). \tag{5.10}$$

Such knowledge is used to build two Takagi-Sugeno N-F models. The results obtained during the identification procedure indicate that the variable X is the most significant variable for the model f_F. This fact is used to simplify the structure of this model by reducing the number of its global inputs to one variable X. The full set of input variables is used only by linear models, which represent the consequents of fuzzy sets. A similar situation occurs for the second model, where the most significant input variable C_V is used to describe the global input of the N-F model. The following structures of the Takagi-Sugeno N-F model were obtained during the identification procedure using real measurements and data generated by the simulator:

Table 5.5. Structures of the Takagi-Sugeno N-F models

	f_F^s	f_X^s	f_F^r	f_X^r
global inputs	X	C_V	X	C_V
local inputs	X, P_1, P_2, T_1	C_V, P_1, P_2, T_1	X, P_1, P_2, T_1	$C_V, P_1, P_2, T_1 t$
number of rules	7	3	7	2

The following notation is introduced in Table 5.5: f_F^s – model (5.9) built using data generated by the simulator, f_X^s – model (5.10) built using data generated by the simulator, f_F^r – model (5.9) built using real data, and f_X^r – model (5.10) built using real data. The performance of the designed fault detection system is verified for the nominal work of the actuator and for all defined faults. Sample results obtained for the model f_F^r during the nominal work of the actuator are shown in Fig. 5.18, where the output of the process and the model as well as the confidence interval for the output of the process are presented. Figure 5.19 presents the residual signal and adaptive thresholds calculated for the data from Fig. 5.18. The results presented on these Figs. were obtained for real data. However, due to the fact that only few faults can be simulated using real data, the remaining experiments were conducted using data generated by the simulator in *MATLAB Simulink*.

Data for all possible faulty scenarios were prepared. The results obtained for the prepared tests are presented in Table 5.6, where the symbol Y means that a fault can be detected by the designed system and the symbol N means that a fault is not detected by the system. Sample results, which contain the output of the process, model and residuals, are presented for some chosen faults. The sampling rate for all presented results is the same and equals 1 s The data presented in Fig. 5.20 and 5.21 describe the behavior of the process, model output and residuals for a big valve fault, i.e. valve clogging. The fault was simulated at the moment 250 and the fault detection system signaled the fault with the delay of 5 s The fault detection system does not have any trouble detecting the fault due to the fact that the residual signal crucially exceeds the interval determined by adaptive thresholds. The results obtained for the next simulated fault are illustrated by Figs. 5.22 and 5.23. Again, valve clogging is generated but now the size of the fault is small. The fault appears at the moment 350 and now the fault detection system needs 23 s to activate the alarm.

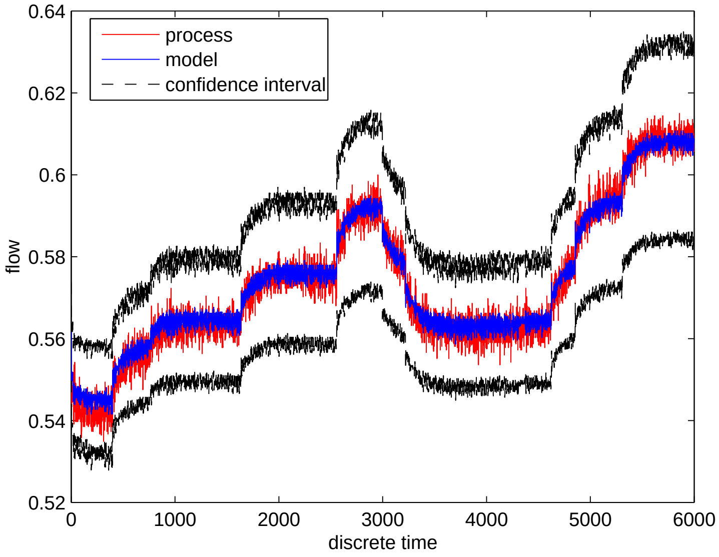

Fig. 5.18. Behavior of the model f_F^r and the actuator for real data

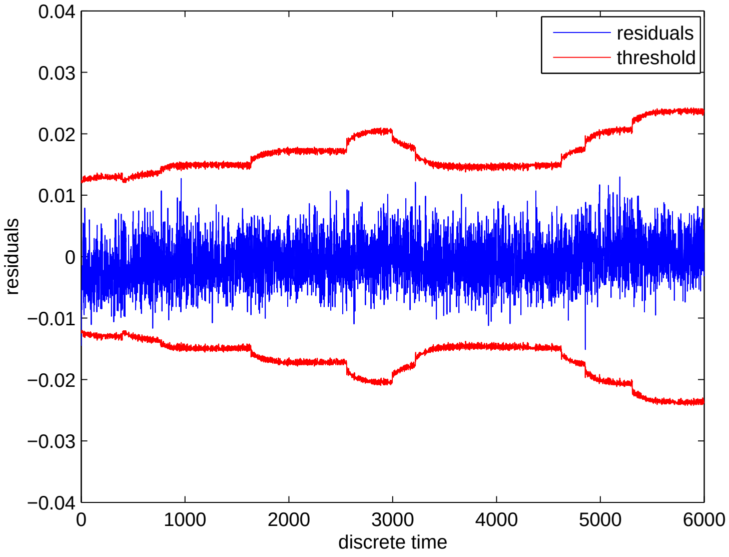

Fig. 5.19. Residuals and adaptive thresholds

Table 5.6. Results

No	Description	S	M	B	I
	Control valve faults				
f_1	Valve clogging	Y	Y	Y	
f_2	Valve plug or valve seat sedimentation			Y	Y
f_3	Valve plug or valve seat erosion				Y
f_4	Increased of valve or bushing friction				Y
f_5	External leakage				N
f_6	Internal leakage				Y
f_7	Medium evaporation or critical flow	Y	Y	Y	
	Pneumatic servo-motor faults				
f_8	Twisted servo-motor's piston rod	N	N	Y	
f_9	Servo-motor's housing or terminals tightness				N
f_{10}	Servo-motor's diaphragm perforation	Y	Y	Y	
f_{11}	Servo-motor's spring fault			Y	Y
	Positioner faults				
f_{12}	Electro-pneumatic transducer fault	N	N	N	
f_{13}	Rod displacement sensor fault	Y	Y	Y	Y
f_{14}	Pressure sensor fault	N	N	N	
f_{15}	Positioner feedback fault			Y	
	General faults				
f_{16}	Positioner supply presser drop	Y	Y	Y	
f_{17}	Unexpected pressure change across the valve			Y	Y
f_{18}	Fully or partly opened bypass valves	Y	Y	Y	Y
f_{19}	Flow rate sensor fault F	Y	Y	Y	

The presented results prove that the developed fault detection system is able to detect abrupt faults, which give rise to jumps in process parameters, resulting in a significant deviation from the normal system behavior. However, it is fundamental for effective fault detection to detect faults which affect process behavior slowly without significant deviations. Such faults are caused by worn-off elements, which may function properly but their parameters are changing in time and finally the element can break down. It is hard to detect such faults at an early stage because symptoms that characterize incipient faults are very similar to symptoms that describe the nominal work of the element. Thus only very sensitive fault detection systems can detect such faults. However, the performance of the sensitive fault detection system can be disturbed by disturbances that corrupt measurements so an ideal fault detection system must be also robust against such undesirable disturbances. In order to illustrate the effectiveness of the developed methods for fault detection of incipient faults, some results obtained for the incipient fault f_4 (bushing frictions)(Fig. 5.24 and 5.25) and the fault (f_{11}) (servo-motor's spring fault) are shown (Figs. 5.26 and 5.27).

The results of the conducted experiments, which are summarized in Table 5.6, show that the developed fault detection system is not able to detect the faults

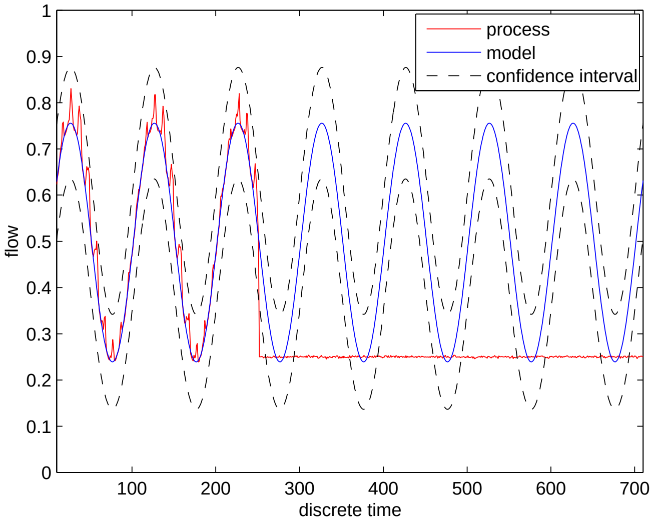

Fig. 5.20 . Big fault f_1: process and model output

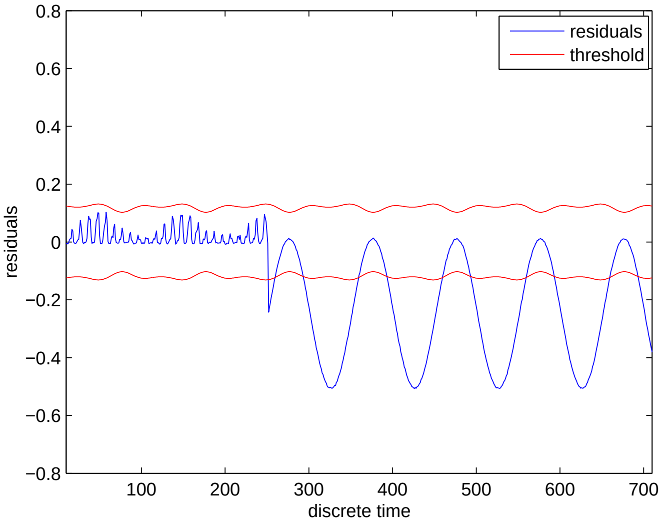

Fig. 5.21 . Big fault f_1: residuals and adaptive thresholds

f_5, f_9, f_{12}, and f_{14}. The fault f_8 (twisted servo-motor's piston rod) is detectable only by the model f_X^s and only in the case of a big fault. The specified faults are not detectable due to the fact that they affect the behavior of the residual

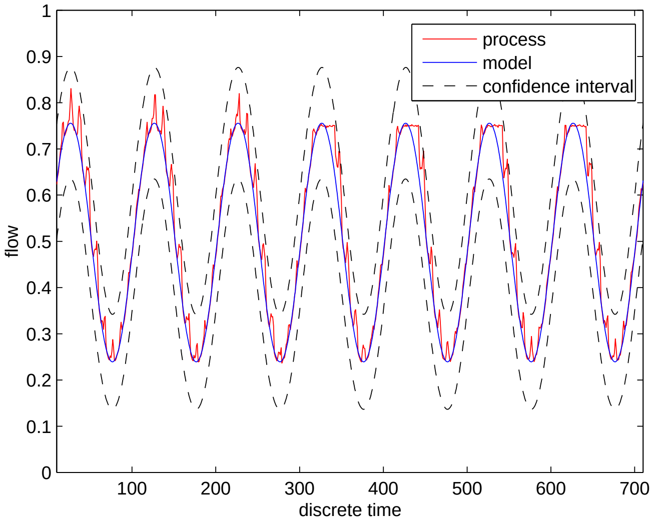

Fig. 5.22. Small fault f_1: process and model output

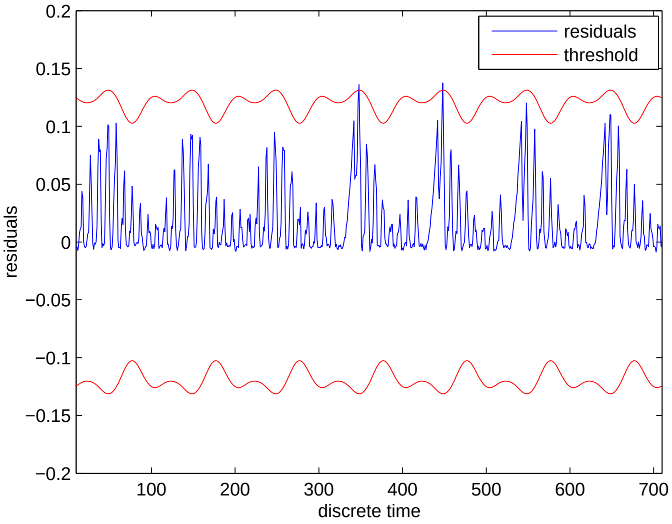

Fig. 5.23. Small fault f_1: residuals and adaptive thresholds

signal to a lower degree than disturbances. As was expected, big faults are usually detected earlier than medium and small faults. An example of such a situation is shown for the experiment conducted for the fault f_1. The residual signal exceeds

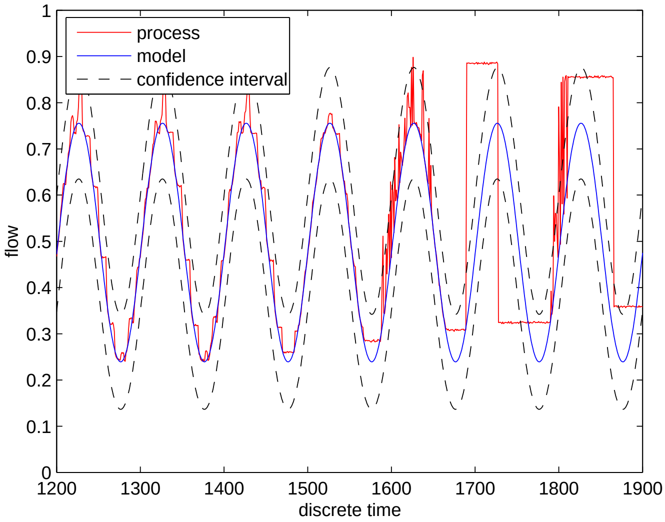

Fig. 5.24. Incipient fault f_4: process and model output

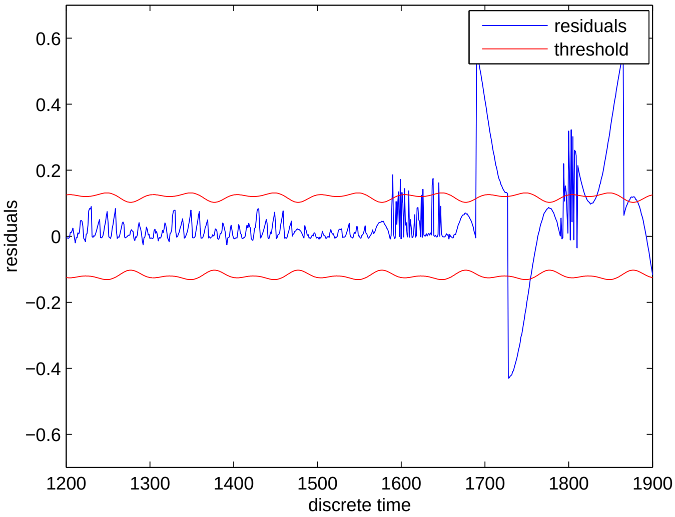

Fig. 5.25. Incipient fault f_4: residuals and adaptive thresholds

considerably the intervals defined by thresholds in the case of a big fault (Fig. 5.23), but for a small fault residuals exceed the calculated interval only slightly (Fig. 5.21). An interesting problem is connected with fault detection of incipient

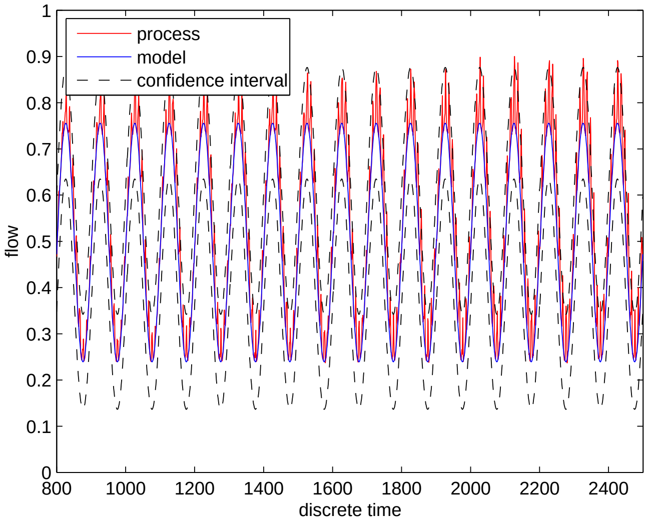

Fig. 5.26. Incipient fault f_{11}: process and model output

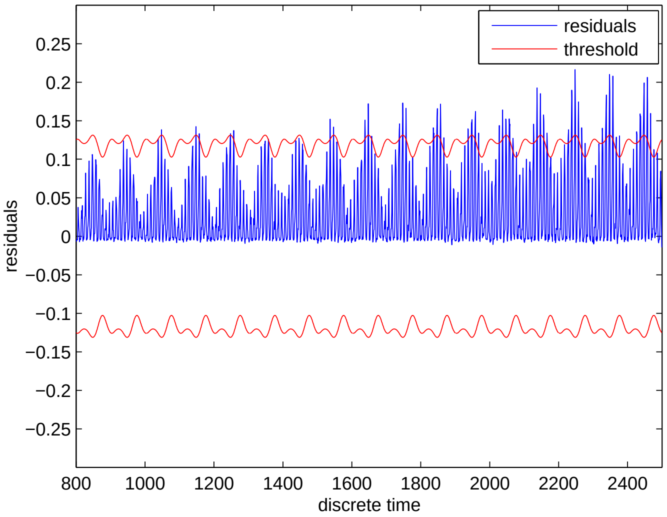

Fig. 5.27. Incipient fault f_{11}: residuals and adaptive thresholds

faults in the control valve, such as f_2 – valve plug or valve seat sedimentation, f_3 – valve plug or valve seat erosion or f_4 – increased valve or bushing friction. Such types of faults affect the process slowly during their exploatation. So it is important to detect such faults before the incipient fault changes into an abrupt one and damages other elements of the process, i.e. the incipient fault valve seat erosion can turn into the twisted servo-motor's piston rod. The developed fault detection system has the ability to detect the incipient fault f_2 at an early stage, whereas the incipient fault f_4 is detected only when the fault is big. The incipient fault valve seat erosion f_3 is detectable by the fault detection system based on the model f_F^s and is not detectable by the system with the model f_X^s. Therefore, it is worth designing fault detection systems that consist of many models in order to ensure a detailed knowledge about the state of the process.

5.6. Summary

The present chapter deals with the problem of fault detection using Takagi-Sugeno N-F networks. The main problems of fault detection are considered in the introduction. The problem of robust fault detection is considered in particular and studied thoroughly. Robustness is understood here as the unsensitivity of the fault detection system to model uncertainty and other disturbances. A method of robust fault detection under model uncertainty is developed using the adaptive threshold approach. The effectiveness of the proposed approach is tested using an actuator, an element of the evaporation station, and a DC engine. The methods developed in Section 4 to design the Takagi-Sugeno N-F model were employed to build models required for the fault detection system. In order to calculate the adaptive threshold, the algorithms proposed in Section 3 were used to determine the uncertainty of N-F models. Experimental results confirm the effectiveness of the proposed solutions. The ability to promptly detect incipient faults is particularly important because such faults affect process behavior slowly and it may take a long time before they are detected by a simple method based on level or trend monitoring. The adaptive threshold technique makes it possible to detect such faults early in the incipient phase and to avoid huge damages and losses.

Chapter 6

CONCLUSIONS

The growing complexity of modern technological processes force the development of sophisticated diagnostics techniques. The model-based approach is one of the most important and well-developed areas of diagnostic. The large grade of the complexity of real processes and the resulting problem with the design of accurate analytical models focus deliberations of scientists on models which can be built using artificial intelligence techniques. Recently, special attention has been paid to hybrid techniques, which connect the features of fuzzy systems and artificial neural networks. Such hybrid models obtained using special identification procedures are commonly employed to fault detection tasks, and they are an interesting alternative when other techniques fail to reach the desired accuracy. Although hybrid techniques possess many advantages, which are very useful in fault detection applications, there are still many unresolved problems connected with them. Structure selection of the N-F model, which ensures the selected accuracy, is a sample fundamental problem. In order to overcome it, a procedure which helps the designer to select the proper structure is developed in this work. The proposed method uses the OBE algorithm in order to estimate the number of approximately linear parts of process input-output characteristics. Next, such knowledge is used to determine the number of fuzzy rules which should be included in the N-F model, and it is possible to estimate the parameters of fuzzy antecedents and linear consequents as well. The developed method of N-F model design usually allows obtaining good accuracy, but the final model is never perfect and some kind of model uncertainty always exists. The main property of a good diagnostic system is robustness against model uncertainty or other disturbances and, simultaneously, the preservation of strong sensitivity of the system to faults. In order to satisfy these requirements, it was necessary to develop a method which would allow estimating the uncertainty of the N-F model. The method is based on the fact that the N-F model can be viewed in the form of an LP system if some assumptions concerning antecedents are established. The uncertainty of the N-F model is represented in the form of the confidence interval for the output signal. The confidence interval is determined using a feasible set of consequent parameters, which is computed by the BEA algorithm. Knowledge about the confidence interval of the model output allows developing a robust fault detection system which uses the adaptive threshold technique to detect faults.

The original results presented in this work can be summarized as follows:

- a method developed in order to determine the uncertainty of the N-F model using the BEA or the OBE algorithm,

- robust fault detection under N-F model uncertainty using the adaptive threshold method, which is based on knowledge about the confidence interval determined for the model output,

- an algorithm for structure selection and parameter estimation of the N-F model, which is based on the detection of approximately linear dependencies using the BEA algorithm,

- the application of the developed methods to the identification task and robust fault detection for the electrical engine and the actuator which are elements of the Lublin sugar factory technological installation.

Streszczenie

Rosnąca złożoność współczesnych procesów technicznych wymusza konieczność opracowywania nowoczesnych technik diagnostycznych. Jednym z dynamicznie rozwijających się obszarów diagnostyki są metody wykorzystujące modele diagnozowanego obiektu. Duży stopień skomplikowania diagnozowanych procesów i wiążąca się z nią trudność pozyskiwania odpowiednich modeli analitycznych, kieruje uwagę projektantów w kierunku modeli opartych o techniki sztucznej inteligencji. W ostatnich latach szczególne zainteresowanie uzyskały techniki hybrydowe łączące cechy systemów rozmytych oraz sieci neuronowych a nazywane rozmytymi sieciami neuronowymi (RSN). Modele te otrzymywane w procesie identyfikacji stanowią interesującą alternatywę i chętnie są wykorzystywane podczas projektowania układów detekcji uszkodzeń. Mimo wielu zalet takiego rozwiązania, istnieje również szereg ciągle nierozwiązanych problemów. Jednym z nich jest problem doboru odpowiedniej struktury sieci umożliwiającej uzyskanie modelu o zadanej dokładności. W tym celu opracowano metodę wspierającą projektanta w procedurze określania struktury RSN. Metoda ta bazuje na algorytmie estymacji przy ograniczonych wartościach błędów EOWB i pozwala oszacować liczbę w przybliżeniu liniowych części charakterystyki modelowanego obiektu. Na podstawie tej wiedzy możliwe okazało się określenie liczby reguł jakie należy zakodować w strukturze RSN oraz uzyskano możliwość oszacowania parametrów zbiorów rozmytych odpowiedzialnych za przełączanie liniowych modeli cząstkowych. Mimo, że zaproponowane metody projektowania RSN uzyskują zadawalające efekty w procedurze identyfikacji modelu, to otrzymany model jest zawsze obarczony niepewnością. Ponieważ jedną z istotniejszych cech niezawodnego systemu diagnostycznego jest odporność na niepewność modelu oraz inne czynniki mogące prowadzić do błędnej diagnozy, przy jednoczesnym zachowaniu wrażliwości na uszkodzenia, bardzo ważnym zadaniem stało się opracowanie metody pozwalającej budować odporne układy detekcji uszkodzeń z wykorzystaniem RSN.

Idea zaproponowanego podejścia zakłada wykorzystanie techniki adaptacyjnych progów decyzyjnych dla sygnału residuum wyznaczonego jako różnica odpowiedzi modelu i diagnozowanego obiektu. Zadaniem progów decyzyjnych jest określenie przedziału wartości sygnału residuum, który odpowiada poprawnej pracy obiektu. Detekcja uszkodzeń w takim przypadku odbywa się poprzez wykrywanie zdarzeń polegających na wykraczaniu sygnału residuum poza wyznaczony przedział ufności. Adaptacyjne progi decyzyjne wyznacza się na podstawie wiedzy o niepewności modelu wykorzystywanego do generacji residuów. Niepewność modelu reprezentowana jest poprzez przedział ufności określany dla sygnały wyjściowego modelu a przedział ten wyznacza się bazując na wiedzy o zbiorze parametrów dopuszczalnych dla danego modelu. W pracy przedstawiono szczegółowo dwa podejścia do problemu wyznaczania niepewności RSN i jednocześnie estymacji jej

parametrów: metodę statystyczną analizy niepewności estymat parametrów uzyskanych metodą najmniejszych kwadratów MNK oraz metodę EOWB. Obie metody pozwalają efektywnie wyznaczyć parametry modelu oraz określić przedział ufności sygnału wyjściowego modelu gdy model ma strukturę systemu liniowego względem parametrów. W przypadku systemów nieliniowych np. RSN konieczna staje się linearyzacja nieliniowej charakterystyki wokół punktu pracy. Niestety takie rozwiązanie może wprowadzać pewną niedokładność odnośnie wyznaczanego przedziału ufności a co za tym idzie wiarygodność detekcji projektowanego układu może być zagrożona. Aby przezwyciężyć tą trudność zastosowano dla RSN rozwiązanie alternatywne, a mianowicie korzystając z faktu, że następniki reguł RSN typu Takagi-Sugeno mają postać liniową, przedstawiono RSN jako system liniowy względem parametrów co wymagało przyjęcia założenia, że parametry poprzedników reguł są znane i wyznaczone zostały wcześniej inną metodą estymacji parametrów.

Ważnym zagadnieniem w kontekście zastosowania RSN do diagnostyki obiektów rzeczywistych jest użyteczność do tego celu zastosowanych metod estymacji parametrów i procedur określania niepewności modelu. Niestety metoda estymacji MNK oraz podejście statystyczne wyznaczania przedziału ufności dla sygnałów wyjściowych nie spełnia tego kryterium. Wiąże się to z koniecznością przyjęcia bardzo rygorystycznych założeń dotyczących zakłóceń obecnych w danych pomiarowych, które w przypadku tych metod muszą charakteryzować się wartością oczekiwaną równą 0 oraz rozkładem normalnym. Założenie to ogranicza w dużym stopniu stosowalność tej metody ponieważ w praktyce rzadko spotyka się obiekty z takim rodzajem zakłóceń. Alternatywą dla opisywanego podejścia jest metoda EOWB, która wymaga jedynie definicji maksymalnych wartości dla zakłóceń danych pomiarowych używanych w procedurze estymacji parametrów modelu. Najprostsze podejście zakłada, że sygnałem niepewnym jest jedynie sygnał wyjściowy obiektu. Jednak w praktyce często okazuje się, że sygnały wejściowe należy traktować również jako niepewne co zostało uwzględnione w odpowiednio zmodyfikowanej metodzie EOWB. Niestety zalety metody EOWB są okupione dużą złożonością obliczeniową, co w przypadku modeli ze znaczną liczbą parametrów może istotnie wydłużyć czas potrzebnych obliczeń. Aby uniknąć takich sytuacji zaadaptowano dla RSN pewną odmianę metody EOWB, algorytm zewnętrznych elipsoid ograniczających ZEO. Algorytm ten jest mniej wymagający obliczeniowo ponieważ jedynie aproksymuje obszar ufności parametrów, który można wyznaczyć dokładnie metodą EOWB.

W końcowej części pracy przedstawiono wyniki badań eksperymentalnych, których celem była praktyczna weryfikacja zaproponowanych wcześniej rozwiązań do odpornej detekcji uszkodzeń. Przedstawione metody zastosowano do detekcji uszkodzeń silnika elektrycznego oraz urządzenia wykonawczego będącego elementem instalacji technologicznej w cukrowni "Lublin". Wyniki badań ukazują dużą skuteczność i efektywność zaproponowanych metod.

BIBLIOGRAPHY

Abonyi J., Babuška R. and Szeifert F. (2002): *Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models.* – IEEE Trans. on Systems, Man, and Cybernetics, Part B, Vol. 32, No. 5, pp. 612-621.

Arabas J. (2001): *Lectures of Evolutionary Algorithms* – Warsaw, Poland: Wydawnictwa Naukowo-Techniczne. (in Polish)

Babuška R. (1998): *Fuzzy Modeling for Control.* – London, UK: Kluwer Academic Publisher.

Babuška R. and Verbruggen H. B. (1995): *A new identification method for linguistic fuzzy models.* – Proc. 4th IEEE Conf. on Fuzzy Systems and 2nd Int. Fuzzy Engineering Symp., Yokohama, Japan, pp. 905-912.

Bai E-W. and Huang Y-F. (1999): *Convergence of optimal sequential outer bounding sets in bounded error parameter estimation.* – Mathematics and Computers in Simulation, Vol. 49, pp 307-317.

Bezdek D. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms.* – New York, USA: Plenum Press.

Bezdek J. and Dunn J. (1975): *Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions.* – IEEE Trans. on Computers, Vol. 24, pp. 835-838.

Broman V and Shensa M. J. (1990): *A compact algorithm for the intersection and approximation of N-dimensional polytopes.* – Math. and Comp. in Simulation, Vol. 32, pp. 469-480.

Calado J. M. F., Louro R., Mendes M. J. G. C., Sa da Costa J. M. G. and Kowal M. (2003): *Fault isolation based on HSFNN applied to DAMADICS benchmark problem.* – Proc. 5th IFAC Symp. on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2003, Washington, DC, USA, pp. 1053-1058.

Calado J. M. F., Kowal M., Mendes M. J. G. C., Korbicz J. and Sa da Costa J. M. G. (2002): *Neuro-fuzzy fault detection approach using a profibus network.* – Proc. 10th Mediterranean Conf. on Control and Automation, MED 2002, Lisbon, Portugal, CD-ROM.

Chan W. C., Chan C. W., Cheung K. C. and Wang Y. (1999): *Modelling of nonlinear stochastic dynamical systems using neuro-fuzzy networks.* – Proc. 38th Conf. on Decision and Control, Pheonix, AZ, USA.

Chen S., Billings S. A. and Luo W. (1999): *Orthogonal least squares methods and their application to nonlinear system identification.* – Int. Journal of Control, Vol. 50, No. 12, pp. 1873-1896.

Chen J-Q., Xi Y-G. and Zhang Z-J. (1998): *A clustering algorithm for fuzzy model identification.* – Fuzzy Sets and Systems, Vol. 98, pp. 319-329.

Chen M-Y. and Linkens D. A. (2001): *A systematic neuro-fuzzy modeling framework with application to material property prediction.* – IEEE Trans. on Systems, Man and Cybernetics – Part B, Vol. 31, No. 5, pp. 781-790.

Chiu S. (1994): *Fuzzy model identification based on cluster estimation.* – Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, pp. 665-685.

Chryssolouris G., Lee M. and Ramsey A. (1996): *Confidence interval prediction for neural network models.* – IEEE Trans. on Neural Networks, Vol. 7, No. 1, pp. 229-232.

Cordon O. and Herrera F. (1999): *A two-stage evolutionary process for designing TSK fuzzy rule based systems.* – IEEE Trans. on Systems, Man and Cybernetics – Part B, Vol. 29, No. 6, pp. 703-715.

Czogała E. and Łęski J. (2000): *Fuzzy and Neuro-Fuzzy Intelligent Systems.* – Heidelberg, Germany: Physica-Verlag.

Dabbenea F., Gayb P. and Polyak B. T. (2003): *Recursive algorithms for inner ellipsoidal approximation of convex polytopes.* – Automatica, Vol. 39, No. 6, pp. 1773-1781.

DAMADICS (2002): *Website of the RTN DAMADICS: Development and Application of Methods for Actutator Diagnosis in Industrial Control Systems.* http://diag.mchtr.pw.edu.pl/damadics.

Dasgupta S. and Huang Y. F. (1987): *Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise.* – IEEE Trans. on Inf. Theory, Vol. 33, No. 3, pp. 383-392.

Diez J. L., Sala A. and Navarro J. L. (2002): *Fuzzy clustering algorithm for local model control.* – Proc. 15th Triennial World Congress of the IFAC, Barcelona, Spain, CD-ROM.

Dinca L., Aldemir T. and Rizzoni G. (1999): *Fault detection and identification in dynamic systems with noisy data and parameter/modeling uncertainties.* – Reliability Engineering and System Safety, Vol. 65, pp. 17-28.

Doering A., Galicki M. and Witte H. (1997): *Structure optimization of neural networks with the A*–algorithm.* – IEEE Trans. on Neural Networks, Vol. 8, No. 6, pp. 1434-1445.

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

Duch W., Korbicz J., Rutkowski L. and Tadeusiewicz R. (Eds.) (2000): *Biocybernetics and Biomedical Engineering 2000. Neural Networks.* – Warsaw, Poland: Akademicka Oficyna Wydawnicza EXIT, Vol. 6. (in Polish)

Edelmayer A. (2000): *Research of quantitative and qualitative FDI methods based on data from Lublin Sugar Factory.* – Proc. 4th IFAC Symp. Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2000, Budapest, Hungary, pp. 351-358.

Eykhoff P. (1980): *Identification in Dynamic Systems* – Warsaw, Poland: Wydawnictwo Naukowe PWN. (in Polish)

Frank P.M. and Ding X. (1997): *Survey of robust residual generation and evaluation methods.* – Jour. of Process Control, Vol. 7, No. 6, pp. 403-424.

Gao X. Z. and Ovaska S. J. (2001): *Soft computing methods in motor fault diagnosis.* – Applied Soft Computing, Vol. 1, pp. 73-81.

Gath I. and Geva A. B. (1989): *Unsupervised optimal fuzzy clustering.* – IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 7, pp. 773-781.

Gertler J. (1998): *Fault Detection and Diagnosis in Engineering Systems.* – New York, USA: Marcel Dekker.

Gertler J. and Kowalczuk Z. (1997): *Detection and distinction of errors using analitical models* – Proc. 2nd Nat. Conf. Diagnostics of Industrial Processes, DPP'97, Łagów Lubuski, Poland, pp. 169-174. (in Polish)

Geva A. B. (1999): *Hierarchical unsupervised fuzzy clustering.* – IEEE Trans. on Fuzzy Systems, Vol. 7 No. 6, pp. 723-733.

Goldberg D. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning* – Massachusetts, USA: Addison-Wesley Publishing Company.

Gustaffson D. E. and Kessel W. C. (1975): *Fuzzy clustering with a fuzzy covariance matrix.* – IEEE Trans. on Fuzzy Systems, Vol. 9, No. 1, pp. 194-199.

Guven M. K. and Passino K. M. (2001): *Avoiding exponential parameter growth in fuzzy systems.* – Proc. IEEE CDC, San Diego, CA, USA, pp. 761-766.

Hadjili M. L. and Wertz V. (2002): *Takagi-Sugeno fuzzy modeling incorporating input variables selection.* – IEEE Trans. on Fuzzy Systems, Vol. 10, No. 6, pp. 728-742.

Higgins C. M. and Goodman R. M. (1994): *Fuzzy rule-based networks for control.* – IEEE Trans. on Fuzzy Systems, Vol. 2, No. 2, pp. 82-88.

Homaifar A. and McCormick E. (1995): *Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms.* – IEEE Trans. on Fuzzy Systems, Vol. 3, No. 6, pp. 129-139.

Hong X. and Harris C. J. (2001): *Variable selection algorithm for construction of MIMO operating point dependent neurofuzzy networks.* – IEEE Trans. on Fuzzy Systems, Vol. 9, No. 1, pp. 88-101.

Isermann R. and Ball P. (1997): *Trends in the application of model-based fault detection and diagnosis of technical processes.* – Control Eng. Practice, Vol. 5, No. 5, pp. 709-719.

Isermann R. (1993): *Fault diagnosis of machines via parameter estimation and knowledge processing.* – Automatica, Vol. 29, No. 4, pp. 815-835.

Janczak A. (2004): *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models. A Block-Oriented Approach.* – Berlin, Heidelberg, Germany: Springer-Verlag.

Jang J. S. R. (1993): *ANFIS: Adaptive network based fuzzy inference system.* – IEEE Trans. on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665-685.

Jin Y. (2000): *Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement.* – IEEE Trans. on Fuzzy Systems, Vol. 8, No. 2, pp. 212-221.

Joh J., Chen Y-H. and Langari R. (1998): *On the stability issues of linear Takagi-Sugeno fuzzy models.* – IEEE Trans. on Fuzzy Systems, Vol. 6, No. 3, pp. 402-410.

Johansson P., Rantzer A. and Arzen K-E. (1999): *Piecewise quadratic stability of fuzzy systems.* – IEEE Trans. on Fuzzy Systems, Vol. 7, No. 6, pp. 713-722.

Juang C-F. and Lin C-T.(1999): *A recurrent self-organizing neural fuzzy inference network.* – IEEE Trans. on Neural Networks, Vol. 10, No. 4, pp. 828-845.

Kang S-J., Woo C-H., Hwang H-S. and Woo K. B. (2000): *Evolutionary design of fuzzy rule base for nonlinear system modeling and control.* – IEEE Trans. on Fuzzy Systems, Vol. 8, No. 1, pp. 37-45.

Kaymak U. and Setnes M. (2002): *Fuzzy clustering with volume prototypes and adaptive cluster merging.* – IEEE Trans. on Fuzzy Systems, Vol. 10, No. 6, pp. 705-712.

Kaymak U. and Babuška R. (1995): *Compatible cluster merging for fuzzy modeling.* – Proc. 4th IEEE Conf. on Fuzzy Systems and 2nd Int. Fuzzy Engineering Symp., Yokohama, Japan, pp. 897-904.

Kirshnapuram R. and Kim J. (1999): *A note on the Gustafson-Kessel and adaptive clustering algorithms.* – IEEE Trans. on Fuzzy Systems, Vol. 7, No. 4, pp. 453-461.

Kirshnapuram R. and Freg C. P. (1992): *Fitting an unknown number of lines and planesto image data through compatible cluster merging.* – Pattern Recognition, Vol. 25, No. 4, pp. 385-400.

Korbicz J., Kościelny J. M., Kowalczuk Z. and Cholewa W. (Eds.) (2004): *Fault Diagnosis. Models, Artificial Intelligence, Applications.* – Berlin, Heidelberg, Germany: Springer-Verlag.

Korbicz J. and Kowal M. (2001): *Neuro-Fuzzy Systems in Processes Diagnostics.* – In: Chojcan J. and Łęski J. (Eds.), Fuzzy Sets and their Applications: book dedicated to Professor Ernest Czogała – Gliwice, Poland: Wydawnictwo Politechniki Śląskiej, pp. 333-379. (in Polish)

Korbicz J., Obuchowicz A. and Uciński D. (1994): *Artificial Neural Networks. Fundations and Applications.* – Warsaw, Poland: Akademicka Oficyna Wydawnicza PLJ. (in Polish)

Korbicz J. and Bidyuk P. (1993): *State and Parameter Estimation. Digital and Optimal Filtering, Applications.* – Zielona Góra, Poland: Oficyna Wydawnicza Wyższej Szkoły Inżynierskiej.

Kościelny J.M. (2004): *Process diagnostics methodology* – In: Korbicz J., Kościelny J.M., Kowalczuk Z. and Cholewa W., (Eds.), Fault Diagnosis. Models, Artificial Intelligence, Applications. – Berlin, Heidelberg, Germany: Springer-Verlag, pp. 59-118.

Kościelny J.M., Bartyś M., Syfert M. and Pawlak M. (2004): *Models in the diagnostics of processes.* — In: Korbicz J., Kościelny J. M., Kowalczuk Z. and Cholewa W. (Eds.), Fault Diagnosis. Models, Artificial Intelligence, Applications. – Berlin, Heidelberg, Germany: Springer-Verlag, pp. 865-902.

Kościelny J.M. (2001): *Diagnostics of Automated Industrial Processes.* – Warsaw, Poland: Akademicka Oficyna Wydawnicza EXIT. (in Polish)

Kościelny J.M., Ostasz A. and Wasiewicz P. (2000): *Fault detection based fuzzy neural networks. Application to sugar factory evaporation.* – Proc. 4th IFAC Symp. Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2000, Budapest, Hungary, pp. 337-342.

Kothari R. and Pitts D. (1999): *On finding the number of clusters.* – Pattern Recognition Letters, Vol. 20, pp. 405-416.

Kowal M. and Korbicz J. (2003): *Self-organizing Takagi-Sugeno fuzzy model in a fault detection system.* – Proc. 6th Nat. Conf. Diagnostics of Industrial Processes, DPP'03, Władysławowo, Poland, pp. 253-258. (in Polish)

Kowal M., Korbicz J., Mendes M. J. G. C. and Calado J. M. F. (2002): *Fault detection using neuro-fuzzy networks.* – Systems Science, Vol. 28, No. 1, pp. 45-57.

Kowal M. and Korbicz J.(2002a): *Fault detection using neuro-fuzzy networks* – Proc. 14th Polish Control Conf., KKA'02, Zielona Góra, Poland, pp. 595-600. (in Polish)

Kowal M. and Korbicz J. (2002b): *Faul detection using Takagi-Sugeno neuro-fuzzy networks and ARX models.* – Nat. Conf. Fuzzy Systems, Cracow, Poland, pp. 198-205.

Kowal M. and Korbicz J. (2001): *Robust diagnosis of process faults using neuro-fuzzy networks.* – Proc. 14th Int. Conf. on Systems Science, Wrocław, Poland, pp. 306-313.

Kowal M. (2001): *Application of the time window method and neuro-fuzzy networks to fault diagnostic* – Proc. 5th Nat. Conf. Diagnostics of Industrial Processes, DPP'01, Zielona Góra, Poland, pp. 199-204. (in Polish)

Kowal M. and Korbicz J. (2000a): *Neuro-fuzzy detector for industrial process.* –In: Hampel R., Wagenknecht M. and Chaker N. (Eds.) Fuzzy Control. Theory and Practice, – Heidelberg, Germany: Physica-Verlag, pp. 294-301.

Kowal M., Korbicz J. (2000b): *Neuro-fuzzy approach for fault diagnosis of dynamic systems.* – Proc. 5th Conf. Neural Networks and Soft Computing, Zakopane, Poland, pp. 503-508.

Kowal M. and Korbicz J. (2000c): *Neuro-fuzzy approach to fault diagnosis in two-tank system.* – Proc. 3rd Conf. on Theory and Applications of Artificial Intelligence, Colloquia in Artificial Intelligence, CAI 2000, Łódź, Poland, pp. 107-120.

Kowal M. (2000): *Neuro-fuzzy networks in diagnostic of industrial processes.* – Proc. 2nd Polish Doctoral Workshop, Istebna, Poland, pp. 34-39. (in Polish)

Kowal M. and Korbicz J. (1999): *Neuro-fuzzy networks in fault detection system for dynamic objects.* – Proc. 2nd Nat. Conf. Methods and Computer Systems in Science Researches and Engineering Design, Cracow, Poland, pp. 613-618. (in Polish)

Kowal M. (1998): *Neuro-fuzzy networks in a fault detection system.* – Proc. 23rd Conf. of Electrical Engineers Scientific Society, Warsaw, Poland, pp. 229-236. (in Polish)

Landajo M., Rio M. J. and Perez R. (2001): *A note on smooth approximation capabilities of fuzzy systems.* – IEEE Trans. on Fuzzy Systems, Vol. 9, No. 2, pp. 229-237.

Leith D. J. and Leithead W. E. (1999): *Analytic framework for blended multiple model systems using local models.* – Int. Jour. of Control, Vol. 72, No. 7/8, pp. 605-619.

Leonhard W. (1996): *Control of Electrical Drives.* – Berlin, Germany: Springer.

Liang B., Payne B. S., Ball A. D. and Iwnicki S. D. (2002): *Simulation and fault detection of three-phase induction motor.* – Mathematics and Computers in Simulation, Vol. 61, pp. 1-15.

Lin Y. and Cunningham G. A. (1991): *A new approach to fuzzy-neural system modeling.* – IEEE Trans. on Computers, Vol. 40, No. 12, pp. 190-198.

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

Łęski J. M. (2003): *Neuro-fuzzy system with learning tolerant to imprecision.* – Fuzzy Sets and Systems, Vol. 138, pp. 427-439.

Maksarov D. and Norton J. P. (1996): *Tuning of noise bounds in state bounding.* – Proc. Symp. Modeling, Analysis and Simulation, CESA'96, IMACS Multiconference, Lille, France, pp. 837-842.

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

Mańczak K. (1976): *Experimental Design Technique.* – Warsaw, Poland: Wydawnictwo Naukowe PWN. (in Polish)

Matia F., Al-Hadithi B. M. and Jimenez A. (2002): *Generalization of stability criterion for Takagi-Sugeno continous fuzzy model.* – Fuzzy Sets and Systems, No. 129, pp. 295-309.

Mendes M. J. G. C., Kowal M., Korbicz J. and Sa da Costa J. M. G. (2002): *Neuro-fuzzy structures in FDI system.* –Proc. 15th Triennial World Congress of the IFAC, Barcelona, Spain, CD-ROM.

Michalewicz Z. (1996): *Genetic algorithms + Data Structures = Evolution Programs.* – Berlin, Germany: Springer.

Milanese M., Norton J. P., Piet-Lahanier H. and Walter E. (1996): *Bounding Approaches to Identification.* – New York, USA: Plenum Press.

Milanese M. and Novara C. (2004): *Set membership identification of nonlinear systems.* – Automatica, Vol. 40, pp. 957-975.

Mouzouris G. C. and Mendel J. M. (1997): *Nonsingleton fuzzy logic systems: Theory and application.* – IEEE Trans. on Fuzzy Systems, Vol. 5, No. 1, pp. 56-71.

Mrugalski M. (2003): *Neural network based modelling of nonlinear systems in fault detection schemes.* Ph.D. thesis, University of Zielona Góra, Faculty of Electrical Engineering and Telecommunications. (in Polish)

Nahorski Z. and Mańczak K. (1983): *Computer Identification of Dynamic Objects.* – Warsaw, Poland: Wydawnictwo Naukowe PWN. (in Polish)

Nelles O. (2001): *Nonlinear System Identification.* – Berlin, Germany: Springer-Verlag.

Nelles O., Fink A., Babuška R. and Setnes M. (2000): *Comparison of two construction algorithms for Takagi-Sugeno fuzzy models.* – International Journal of Applied Mathematic and Computer Science, Vol. 10, No. 4, pp. 835-855.

Obuchowicz A. (2003): *Evolutionary Algorithms for Global Optimization and Dynamic System Diagnosis.* – Zielona Góra, Poland: Lubuskie Towarzystwo Naukowe.

Osowski S. (1996): *Neural Networks in an Algorithmic View* – Warsaw, Poland: Wydawnictwa Naukowo-Techniczne. (In Polish)

Papadopoulos G., Edwards P. J. and Murray A. F. (2001): *Confidence estimation methods for neural networks: A practical comparison.* – IEEE Trans. on Neural Networks, Vol. 12, No. 6, pp. 1278-1287.

Patton R. and Korbicz J. (Eds.) (1999): *Advances in Computational Intelligence for Fault Diagnosis Systems.* – Special issue of Int. J. Appl. Math and Comp. Sci., Vol. 9, No. 3.

Patton R. J. and Chen J. (1999): *Robust Model-Based Fault Diagnosis for Dynamic Systems.* – London, UK: Kluwer Academic Publishers.

Patton R. J., Frank P. and Clark R. N. (Eds.) (2000): *Issues of Fault Diagnosis for Dynamic Systems.* – Berlin, Germany: Springer-Verlag.

Pau L.F. (1981): *Failure Diagnosis and Performance Monitoring.* – New York, USA: Marcel Dekker.

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

Piegat A. (2003): *Modelling and Fuzzy Systems.* – Warsaw, Poland: Akademicka Oficyna Wydawaniczna EXIT. (in Polish)

Piet-Lahanier H. and Walter E. (1994): *Exact description of feasible parameter sets and minimax estimation.* – Int. J. of Adaptive Control and Signal Processing, Vol. 8, pp. 5-14.

Pomares H., Rojas I., Ortega J., Gonzalez J. and Prieto A. (2000): *A systematic approach to a self-generating fuzzy rule-table for function approximation.* – IEEE Trans. on Systems, Man and Cybernetics – Part B, Vol. 30, No. 3, pp. 431-447.

Rafajłowicz E. (1996): *Algorithms of Experimental Design with Implementations in MATHEMATICA* – Warsaw, Poland: Akademicka Oficyna Wydawnictwo PLJ. (in Polish)

Rice H. J., and Xu K. Q. (1996): *Linear path identification of general non-linear systems.* – Mechanical Systems and Signal Processing, Vol. 10, No. 1, pp. 55-63.

Runkler T. A. (1997): *Selection of appropriate deffuzzification methods using application specific properties.* – IEEE Trans. on Fuzzy Systems, Vol. 5, No. 1, pp. 72-79.

Rutkowska D. (2003): *Hybrid methods in artificial intelligence* – In. Cierniak R. (Ed.), Evolution or Revolution: Modern Computing Techniques, Częstochowa, Poland: Politechnika Częstochowska, pp. 41-52. (in Polish)

Rutkowska D. (2002): *Neuro-Fuzzy Architectures and Hybrid Learning.* – Heidelberg, New York: Physica-Verlag, Springer-Verlag Company.

Rutkowska D., Nowicki L. and Rutkowski L. (2000): *Neuro-fuzzy architectures with various implication operators.* In: Sinčák P, Vaščák J., Kvasnička V., Mesiar R. W. P. (Eds.), The State of the Art in Computational Intelligence – New York, USA: Springer-Verlag Company.

Rutkowska D. and Zadeh L. (Eds.) (2000): *Neuro-Fuzzy and Soft Computing.* – Special issue Int. J. Appl. Math and Comp. Sci., Vol. 10, No. 4.

Rutkowska D. (1997): *Intelligent Computational Systems.* – Warsaw, Poland: Akademicka Oficyna Wydawnicza PLJ. (in Polish)

Rutkowska D., Piliński M. and Rutkowski L. (1997): *Neural Networks, Genetic Algorithms and Fuzzy Systems.* – Warsaw, Poland: Wydawnictwo Naukowe PWN. (In Polish)

Rutkowski L. (2004a): *Flexible Neuro-Fuzzy Systems. Structures, Learning and Performance Evaluation.* – Boston, USA: Kluwer Academic Publishers.

Rutkowski L. (2004b): *New Soft Computing Techniques for System Modelling, Pattern Classification and Image Processing.* – Berlin, Heidelberg, Germany: Springer-Verlag.

Rutkowski L. and Cpałka K. (2003): *Flexible neuro-fuzzy systems.* – IEEE Trans. on Neural Networks, Vol. 14, No. 3, pp. 554-574.

Rutkowski L. and Cpałka K. (2002): *Flexible fuzzy-neural systems* – Proc. 14th Polish Control Conf., KKA'02, Zielona Góra, Poland, pp. 813-818. (in Polish)

Rutkowski L. and Starczewski J. (2002): *New methods for modelling uncertainty in fuzzy systems* – Proc. 14th Polish Control Conf., KKA'02, Zielona Góra, Poland, pp. 885-888. (in Polish)

Seliger R. and Frank P. (2000): *Robust observer-based fault diagnosis in non-linear uncertain systems.* – In: Patton R. J., Frank P., Clark R. N. (Eds.), Issues of Fault Diagnosis for Dynamic Systems, – Berlin, Germany: Springer-Verlag.

Setnes M. (2000): *Supervised fuzzy clustering for rule extraction.* – IEEE Trans. on Fuzzy Systems, Vol 8, No. 4, pp. 416-424.

Shi Y. and Mizumoto M. (2000): *Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method.* – Fuzzy Sets and Systems, Vol 112, pp. 51-63.

Söderström T. and Stoica P. (1994): *System identification.* – London, UK: Prentice Hall.

Szulim R. (2004): *Knowledge acquisition method for aiding the performance of a complex technological process.* Ph.D. thesis, University of Zielona Góra, Faculty of Electrical Engineering and Telecommunications. (in Polish)

Tadeusiewicz R. (1993): *Neural Networks.* – Warsaw, Poland: Akademicka Oficyna Wydawnicza RM. (in Polish)

Takagi T. and Sugeno M. (1985): *Fuzzy identification of systems and its application to modelling and control.* – IEEE Trans. on Systems, Man and Cybernetics, Vol. 15, No. 1, pp. 116-132.

Walter E. and Piet-Lahanier H. (1990): *Estimation of parameter bounds from bounded-error data: A survey.* – Mathematics and Computers in Simulation, Vol. 32, pp. 449-468.

Walter E. and Pronzato L. (1997): *Identification of Parametric Models from Experimental Data.* – Berlin, Germany: Springer-Verlag.

Wang L. X. and Mendel J. M. (1992): *Back-propagation fuzzy system as nonlinear dynamic system identifiers.* – Proc. 1st IEEE Conf. on Fuzzy Systems, San Diego, USA, pp. 1409-1418.

Witczak M. (2003): *Identification and Fault Detection of Non-linear Dynamic Systems.* – Zielona Góra, Poland: Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Lecture Notes in Control and Computer Science, Vol. 1.

Witczak M. and Mrugalski M. (2003): *Designing GMDH neural networks with the bounded-error approach: Application to robust fault detection.* – Proc. Int. Conf. Methods and Models in Automation and Robotics, MMAR 2003, Międzyzdroje, Poland, pp. 1377-1382.

Yager R. R. and Filev D. P. (1994): *Essentials of Fuzzy Modelling and Control.* – New York, USA: John Willey and Sons.

Yager R. R. and Filev D. P. (1994): *Generation of fuzzy rules by mountain clustering.* – Journal of Intelligent and Fuzzy Systems, Vol. 2, No. 3, pp. 209-219.

Yen J. and Wang L. (1999): *Simplifying fuzzy rule-based models using orthogonal transformation methods.* – IEEE Trans. on Systems, Man and Cybernetics-Part B, Vol. 29, No. 1, pp. 13-24.

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

Lecture Notes in Control and Computer Science

Editor-in-Chief: Józef KORBICZ

Vol. 9: Marek Kowal
Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems
116 p. 2005 [83-89712-88-1]

Vol. 8: Wojciech Paszke
Analysis and Synthesis of Multidimensional System Classes Using Linear Matrix Inequality Methods
188 p. 2005 [83-89712-81-4]

Vol. 7: Piotr Steć
Segmentation of Colour Video Sequences Using the Fast Marching Method
110 p. 2005 [83-89712-47-4]

Vol. 6: Grzegorz Łabiak
Wykorzystanie hierarchicznego modelu współbieżnego automatu
w projektowaniu sterowników cyfrowych
168 p. 2005 [83-89712-42-3]

Vol. 5: Maciej Patan
Optimal Observation Strategies for Parameter Estimation of Distributed Systems
220 p. 2004 [83-89712-03-2]

Vol. 4: Przemysław Jacewicz
Model Analysis and Synthesis of Complex Physical Systems Using Cellular Automata
134 p. 2003 [83-89321-67-X]

Vol. 3: Agnieszka Węgrzyn
Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego
125 p. 2003 [83-89321-54-8]

Vol. 2: Grzegorz Andrzejewski
Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych
109 p. 2003 [83-89321-53-X]

Vol. 1: Marcin Witczak
Identification and Fault Detection of Non-Linear Dynamic Systems
124 p. 2003 [83-88317-65-2]

Prace Naukowe z Automatyki i Informatyki

Przewodniczący: Józef KORBICZ

Vol. 9: Marek Kowal
Optimization of Neuro-Fuzzy Structures in Technical Diagnostics Systems
116 p. 2005 [83-89712-88-1]

Vol. 8: Wojciech Paszke
Analysis and Synthesis of Multidimensional System Classes Using Linear Matrix Inequality Methods
188 p. 2005 [83-89712-81-4]

Vol. 7: Piotr Steć
Segmentation of Colour Video Sequences Using the Fast Marching Method
110 p. 2005 [83-89712-47-4]

Tom 6: Grzegorz Łabiak
Wykorzystanie hierarchicznego modelu współbieżnego automatu
w projektowaniu sterowników cyfrowych
168 s. 2005 [83-89712-42-3]

Tom 5: Maciej Patan
Optimal Observation Strategies for Parameter Estimation of Distributed Systems
220 s. 2004 [83-89712-03-2]

Tom 4: Przemysław Jacewicz
Model Analysis and Synthesis of Complex Physical Systems Using Cellular Automata
134 s. 2003 [83-89321-67-X]

Tom 3: Agnieszka Węgrzyn
Symboliczna analiza układów sterowania binarnego z wykorzystaniem wybranych metod analizy sieci Petriego
125 s. 2003 [83-89321-54-8]

Tom 2: Grzegorz Andrzejewski
Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych
109 s. 2003 [83-89321-53-X]

Tom 1: Marcin Witczak
Identification and Fault Detection of Non-Linear Dynamic Systems
124 s. 2003 [83-88317-65-2]