

Andrzej CZAJKOWSKI, Krzysztof PATAN
 INSTITUTE OF CONTROL AND COMPUTATION ENGINEERING
 UNIVERSITY OF ZIELONA GÓRA, ul. Podgórna 50, 65-246 Zielona Góra

Robust fault detection and accommodation of the boiler unit using state space neural networks

MSIE Andrzej CZAJKOWSKI

Received M.Sc. degree in computer science from the University of Zielona Góra, Poland in 2009. Currently, he is a lecturer at the Institute of Control and Computation Engineering, University of Zielona Góra. His research interests include artificial neural networks, especially state space neural models, fault detection and accommodation and fault tolerant control. He has published a number of scientific papers in conference proceedings such as: Advanced Control and Diagnosis (ACD) and International PhD Workshop (OWD).



e-mail: A.Czajkowski@issi.uz.zgora.pl

D.Sc.IE Krzysztof PATAN

Krzysztof Patan is an associate professor at the Institute of Control and Computation Engineering, University of Zielona Góra. His research include artificial neural networks, modelling of nonlinear systems, fault diagnosis, fault tolerant control, optimization and experimental design. He has published more than 95 technical papers, 17 of which in international journals and 1 monograph Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes (Springer, 2008).



e-mail: K.Patan@issi.uz.zgora.pl

Abstract

The paper deals with application of state space neural network models to fault detection and accommodation of a boiler unit. The work describes two aspects. The first one is the fault detection. In this paper three methods for fault diagnosis, namely: simple and adaptive threshold as well as more robust method which is model error modelling, are described and compared. The second part of the paper presents the approach to fault accommodation based on the so-called instantaneous linearization of the already trained nonlinear state space model of the system. With the obtained linear model it is possible to derive a new control law of the boiler unit in order to eliminate the fault effect in the case of faults. All data used in experiments are collected from the boiler unit simulator implemented in Matlab/Simulink.

Keywords: state space neural networks, uncertainty, model error modelling, fault detection and accommodation, boiler unit.

Odporna detekcja i kompensacja uszkodzeń układu zbiornika przepływowego za pomocą sztucznych sieci neuronowych w przestrzeni stanów

Streszczenie

Artykuł dotyczy zastosowania modelu sztucznej sieci neuronowej w przestrzeni stanów do wykrywania i kompensacji uszkodzeń w układzie sterowania zbiornikiem przepływowym. Do wykrycia uszkodzenia zostały zaproponowane i doświadczalnie przetestowane trzy metody. Dwie pierwsze metody czyli progowanie proste oraz adaptacyjne polegają na obserwacji sygnału rezydium i podejmowaniu decyzji przy przekroczeniu zadanego dopuszczalnego progu przez wartość tego sygnału. Trzecia metoda opiera się na zastosowaniu dodatkowego modelu dynamicznego do modelowania błędu modelu podstawowego w celu określenia zakresu niepewności jego pracy. W przypadku przekroczenia tego zakresu, można uznać, że wystąpiło uszkodzenie. Drugim podjętym przez autorów tematem jest problem kompensacji wykrytego uszkodzenia. W pracy opisuje się podejście oparte na tzw. chwilowej linearyzacji nauczonego w trybie off-line nieliniowego modelu systemu. Na podstawie zlinearyzowanego modelu możliwe jest wyznaczenie nowego prawa sterowania w celu wyeliminowania wpływu uszkodzenia w przypadku wystąpienia awarii. Wszystkie dane wykorzystywane do celów doświadczalnych są zbierane z symulatora zbiornika zrealizowanego w pakiecie Matlab/Simulink.

Słowa kluczowe: Model neuronowy w przestrzeni stanów, niepewność, modelowanie błędu modelu, detekcja i kompensacja uszkodzeń, układ walczaka.

1. Introduction

The increasing requirements for high levels of system performance and reliability in the presence of unexpected changes of system functions cause that Fault Tolerant Control (FTC) systems has received the increasing attention in the last years [2].

Sensor or actuator faults, product changes, material consumption may affect the controller performance [2, 6] and can result in large economic losses and even violation of the safety regulations [3, 6]. To date, many different FTC schemes were investigated [2, 12, 17]. The existing FTC methods can be divided into two groups: passive and active approaches [9, 17]. Passive approaches are designed to work with presumed failure modes and their performance tends to be conservative, especially in the case of unanticipated faults. In contrast, active methods react to the occurrence of system faults on-line and attempt to maintain the overall system stability and performance even in the case of unanticipated faults.

In spite of the fact that many effective control schemes have been elaborated to date, including the optimal, predictive, adaptive control, etc., many industrial installations and plants still use the standard PID controllers. In such cases, the fault tolerant system should be designed based on the existing control with a PID controller. In this paper an active fault tolerant control scheme is proposed. It uses a robust state space model of the system and nonlinear state observers. Both components are carried out by means of artificial neural networks. Additionally, based on the neural model of the system, the fault detection block is designed. This block is used to timely switch the modified control law in order to compensate a fault effect. The modified control law is derived using the so-called instantaneous linearization of the already trained nonlinear state space model of the system [8]. The similar approach was considered in the previous work of the authors[4], however, the uncertainty of the state space neural model as well as the robust fault detection have not been considered yet.

The paper is organized as follows. In Section 2 the main idea of fault compensation is briefly portrayed based on the state space representation of the process. Section 3 presents details about state space neural networks and considerations on the uncertainty. To realize a robust model of the process a model error modelling technique was used. Based on the robust model the fault detection can be easily realized. The nonlinear state observer, required to estimate a fault function, is introduced in Section 4 along with the methodology how to calculate the augmented control law. In this framework the so-called instantaneous linearization technique is applied. Boiler unit specification and a set of simulated faulty scenarios are given in Section 5. Section 6 reports experimental results including the selection of a neural model, the fault detection and fault compensation.

2. Problem formulation

Let us consider a nonlinear dynamic system governed by the following state equation:

$$\mathbf{x}(k+1) = g(\mathbf{x}(k), \mathbf{u}(k)) + f(\mathbf{x}(k), \mathbf{u}(k)), \quad (1)$$

where $g(\cdot)$ is a process working under the normal operating conditions, $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ is the control and $f(\cdot)$ represents a fault affecting the process. An unknown fault function $f(\cdot)$ is a function of both the state and input. Thus, it is possible to model a wide range of possible faults, not only additive ones. When the process works under the normal operating conditions the fault function $f(\cdot)$ is equal to zero. As, in general, the state vector is not available, to approximate the fault function one need to design:

(i). the model of the healthy process in the form:

$$\bar{\mathbf{x}}(k+1) = \bar{g}(\bar{\mathbf{x}}(k), \mathbf{u}(k)), \quad (2)$$

where $\bar{\mathbf{x}}$ is the state vector of the model, \bar{g} - the model of the process working at the normal operating conditions,
 (ii). the state observer:

$$\hat{\mathbf{x}}(k+1) = \hat{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{y}(k)). \quad (3)$$

where $\hat{\mathbf{x}}$ is the estimated state vector, \hat{g} - the mapping realized by the observer, \mathbf{y} - output of the process.

Then, the unknown fault function can be approximated as:

$$\hat{f} = \hat{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{y}(k)) - \bar{g}(\bar{\mathbf{x}}(k), \mathbf{u}(k)). \quad (4)$$

The fault effect occurring in the control system can be compensated/eliminated by a proper determination of the auxiliary input \mathbf{u}^{fa} based on the estimated fault function \hat{f} . This additional control is added to the control $\mathbf{u}(k)$ calculated by the standard controller used. As a result, one can determine the augmented control law \mathbf{u}^{fc} as follows:

$$\mathbf{u}^{fc}(k) = \mathbf{u}(k) + \mathbf{u}^{fa}(k), \quad (5)$$

The details how it can be done are presented later on in the paper.

3. Robust state space neural network model

A very important class of dynamic neural networks is the State Space Neural Network (SSNN) depicted in Fig. 1. The output of the hidden layer is fed back to the input layer through a bank of unit delays. The number of unit delays determines the order of the model. In general, a user decides how many neurons are used to produce feedback. Let $\mathbf{u}(k) \in \mathbb{R}^n$ be the input vector, $\bar{\mathbf{x}}(k) \in \mathbb{R}^q$ - the output of the hidden layer at time k , and $\bar{\mathbf{y}}(k) \in \mathbb{R}^m$ - the output vector.

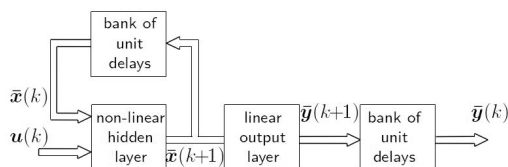


Fig. 1. Block scheme of the state space neural network
 Rys. 1. Schemat blokowy sieci neuronowej w przestrzeni stanów

The state space representation of the neural model considered is described by the equations

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \bar{g}(\bar{\mathbf{x}}(k), \mathbf{u}(k)) \\ \bar{\mathbf{y}}(k) &= \mathbf{C}\bar{\mathbf{x}}(k) \end{aligned}, \quad (6)$$

where $\bar{g}(\cdot)$ is a nonlinear function characterizing the hidden layer, and \mathbf{C} represents synaptic weights between hidden and output neurons. Introducing the weight matrix between the input and hidden layers \mathbf{W}^u and the matrix of recurrent links \mathbf{W}^x , the representation (6) can be rewritten in the following form:

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= h(\mathbf{W}^x \bar{\mathbf{x}}(k) + \mathbf{W}^u \mathbf{u}(k)) \\ \bar{\mathbf{y}}(k) &= \mathbf{C}\bar{\mathbf{x}}(k) \end{aligned}, \quad (7)$$

where h stands for the activation function of the hidden neurons. In most cases the hyperbolic tangent activation function is selected giving pretty well modelling result. For the state space model the outputs of hidden neurons which are fed back are, in general, unknown during training. Therefore, state space neural models can be trained only by minimizing the simulation error. If state measurements are available, the training can be carried out much easier, using the series-parallel identification scheme similarly as for the external dynamic approach (the feed forward network with tapped delay lines). In spite of this inconvenience, state space models have a number of advantages, contrary to other recurrent neural networks:

- The number of states (model order) can be selected independently of the number of hidden neurons. In the recurrent networks, e.g. Williams-Zipser, Elman, locally recurrent networks, the number of neurons directly influences the model order, which significantly impedes the modelling phase.
- Model states are easily accessible from the outside (they feed the network input). This property can be useful when state measurements are available at some time instants.
- State space neural models are useful in the fault tolerant control framework as they can be used to determine the approximation of a fault effect. As the fault effect can be represented in the state space one can handle different kinds of faults including multiplicative and additive ones.

3.1. Model uncertainty

One cannot expect that the designed model of the plant is a faithful replica of plant dynamics. Additionally, disturbances and noise acting upon the system should be taken into account. Therefore, the uncertainty of the model should be considered, which leads to the robust identification. The robust identification procedure should deliver not only a model of a given process, but also a reliable estimate of the uncertainty associated with the model. One of the possible solutions is to identify the process without robustness considerations first, and then consider robustness as an additional step [9, 14]. After that, one can estimate the uncertainty of the model by analysing residuals evaluated from the inputs. The uncertainty is a measure of unmodelled dynamics, noise and disturbances. Identification of residuals provides the so-called model error model. Designing procedure is described by the following steps:

1. Using a model of the process, compute the residual $r = y - \bar{y}$, where y and \bar{y} are the measured and model outputs, respectively;
 2. Collect the data $\{u_i, r_i\}_{i=1}^N$ and identify an error model using these data. This model constitutes an estimate of the error due to under modelling, and it is called model error model;
 3. Derive the centre of the uncertainty region as $\bar{y} + \tilde{y} \approx y$, where \tilde{y} is the output of the error model.
 4. Form the uncertainty region using statistical properties of the error model.
- Around the centre, there is formed a confidence region with the upper band

$$r_u = \bar{y} + \tilde{y} + t_\alpha v \quad (8)$$

and the lower band

$$r_l = \bar{y} + \tilde{y} - t_\alpha v \tag{9}$$

where \tilde{y} is the output of the error model at the input \mathbf{u} , t_α is $N(0,1)$ tabulated value assigned to $1-\alpha$ confidence level, v is the standard deviation of \tilde{y} . The error model can be realized using different techniques. The simplest way is to use a FIR filter, as discussed in [13], but also one can use the ARX model or its nonlinear version. The model error modelling idea was successfully applied to design robust locally recurrent neural networks [9]. In the present work, this methodology is used to design the robust state space neural model.

3.2. Fault detection

Using the state space model of the system the model-based fault detection can be realized. First, residuals should be calculated as a difference between the process output y and the model output \hat{y} . Then, to evaluate residuals and to obtain information about faults, simple thresholding can be applied. If residuals are smaller than the threshold value, a process is considered to be healthy, otherwise it is faulty. In practice, due to modelling uncertainty and measurement noise, it is necessary to assign thresholds larger than zero in order to avoid false alarms [1, 3, 9]. This operation causes a reduction in fault detection sensitivity. Therefore, the choice of the threshold is only a compromise between fault detection sensitivity and false alarm rate. Assuming that the residual is an $N(m, v)$ random variable, thresholds are assigned to the values:

$$T = m \pm \sigma v \tag{10}$$

where σ , in the most cases, is equal to 1, 2 or 3.

The better fault detection results can be obtained when adaptive thresholds are applied. The idea is to construct thresholds based on estimated statistics of the residual calculated using a moving time window of the length n . Then the thresholds are determined in the following way:

$$T(k) = m(k) \pm \sigma v(k) \tag{11}$$

For the details, the interested reader is referred to [9], Section 7.3, pages 133-134. The robust state space model introduced in Subsection 3.1 can be also effectively used to fault detection. In this case not the residual but the system output is monitored. Observing the system output, one may take a decision whether a fault occurred or not. If the system output is inside the uncertainty region defined by thresholds (8) and (9), the system is considered healthy.

The fault detection block is very important during designing fault tolerant control. Only in the case when a fault is detected, the fault accommodation procedure starts to work determining additional input \mathbf{u}^{fa} .

4. Fault accommodation

4.1. Nonlinear state observer

The proposed fault compensation scheme requires to estimate *on-line* the state vector of the plant. Thus, there is a need to design a state observer of the system considered. This can be carried out using the State Space Innovation Form (SSIF) shown in Fig. 2.

The identified SSIF model can be regarded as an extended Kalman filter for unknown nonlinear systems [8]. The SSIF neural model is represented as follows:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \hat{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{e}(k)) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k) \end{aligned} \tag{12}$$

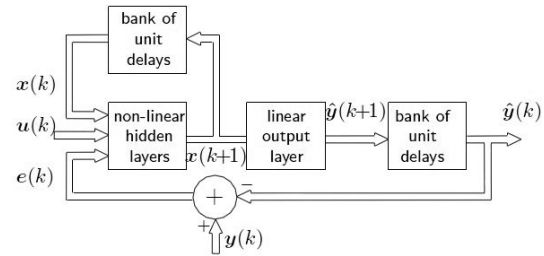


Fig. 2. Block scheme of SSIF model
Rys. 2. Schemat blokowy modelu SSIF

where $\mathbf{e}(k)$ is the error between the observer output $\hat{\mathbf{y}}(k)$ and measured system output $\mathbf{y}(k)$. Introducing weight matrices of the neural observer the equation (12) can be rewritten in the following form:

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= h(\mathbf{W}^x \hat{\mathbf{x}}(k) + \mathbf{W}^u \mathbf{u}(k) + \mathbf{W}^e \mathbf{e}(k)) \\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k) \end{aligned} \tag{13}$$

where \mathbf{W}^x is the matrix of recurrent links, \mathbf{W}^u – the input weight matrix, \mathbf{W}^e – the error weight matrix, and h is the activation function of the hidden neurons.

Then using the estimated state $\hat{\mathbf{x}}(k)$ the unknown fault function \mathbf{f} can be approximated using (4).

4.2. Deriving a control law

The accommodation of a fault means that it is necessary to change a control law so that the effect caused by a fault should be compensated. It can be carried out by adding a new component to the existing control law according to (5). The additional control $\mathbf{u}^{fa}(k)$ should be chosen in such a way that a fault effect is compensated. The problem can be easily solved for linear systems [15]. Assuming that the nominal model of the system is linear and introducing control $\mathbf{u}^{fa}(k)$, the process (1) can be rewritten in the following way:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}(\mathbf{u}(k) + \mathbf{u}^{fa}(k)) + \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)). \tag{14}$$

To completely compensate the fault effect, the fault model should be as close as possible to the nominal model, therefore

$$\mathbf{B}\mathbf{u}^{fa}(k) + \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = 0 \tag{15}$$

then

$$\mathbf{u}^{fa}(k) = -\mathbf{B}^- \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)). \tag{16}$$

where \mathbf{B}^- represents the pseudoinverse of the control matrix. Taking into account that \mathbf{f} is unknown it can be replaced with its approximation given by $\hat{\mathbf{f}}$. Finally, using (4) one obtains

$$\mathbf{u}^{fa}(k) = -\mathbf{B}^- (\hat{g}(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{y}(k)) - \bar{g}(\bar{\mathbf{x}}(k), \mathbf{u}(k))) \tag{17}$$

4.3. Model linearisation

In order to use control law (17) to accommodate faults in the control system, the state space neural model (7) have to be linearized first. In this paper the instantaneous linearization is applied. The idea is very simple. At each sample time a linear model is extracted from the state space neural model. Linearization can be carried out by expanding the model into the Taylor series around the current operating point $(\mathbf{x}, \mathbf{u}) = (\mathbf{x}(\tau), \mathbf{u}(\tau))$.

$\mathbf{u}(\tau)$), rejecting the nonlinear components. As a result, the linear state space model in the form

$$\begin{aligned} \bar{\mathbf{x}}(k+1) &= \mathbf{A}\bar{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{D} \\ \bar{\mathbf{y}} &= \mathbf{C}\bar{\mathbf{x}}(k) \end{aligned} \quad (18)$$

is obtained, where $\mathbf{A} = h'\mathbf{W}^x$, $\mathbf{D} = \mathbf{x}(\tau) - \mathbf{A}\bar{\mathbf{x}}(\tau - 1) - \mathbf{B}\mathbf{u}(\tau - 1)$, $\mathbf{B} = h'\mathbf{W}^u$. Symbol h' represents the first derivative of the activation function. If one uses a hyperbolic tangent as the activation function, this derivative can be simply calculate as

$$h' = 1 - \tanh^2. \quad (19)$$

This is a very useful property as the linearization can be performed very quickly not burdening significantly the simulations carried out in real-time.

5. Boiler unit

The object considered in this work is a laboratory installation developed at the Institute of Automatic Control and Robotics of the Warsaw University of Technology [5]. The installation is dedicated for investigations of diagnostic methods of industrial actuators and sensors [10]. The whole system consists of a boiler, a storage tank, a control valve with a positioner, a pump and transducers to measure process variables. The system is controlled by the standard PID controller. The boiler has the form of a horizontally placed cylinder, which introduces strong nonlinearity into the static characteristic of the system. The scheme of the boiler unit with process variables marked is presented in Fig. 3.

In turn, the specification of process variables is shown in Table 1.

Tab. 1. Specification of process variables
Tab. 1. Charakterystyka zmiennych procesu

Variable	Specification	Range
CV	control value	0-100 %
dP	pressure difference on valve V_1	0-275 kPa
P	pressure before valve V_1	0-500 kPa
F_1	flow (electromagnetic flowmeter)	0-5 m ³ / h
F_2	flow (Vortex flowmeter)	0-5 m ³ / h
L	water level in boiler	0-0.5 m

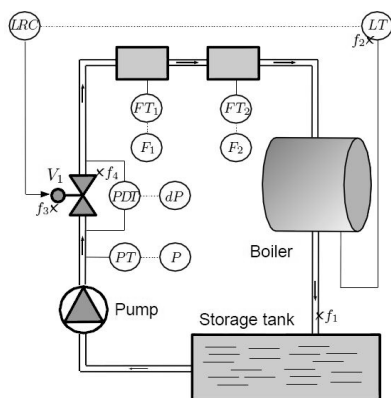


Fig. 3. Boiler unit and possible faults placement
Rys. 3. Układ walczaka wraz z lokalizacją możliwych uszkodzeń

The boiler unit together with the control system was implemented in Matlab/Simulink. Simulations were performed with the sample time equal to 0.05. The simulation model of the

boiler unit renders it possible to generate a number of faulty situations. The specification of faults is presented in Table 2.

Tab. 2. Specification of faulty scenarios
Tab. 2. Charakterystyka możliwych scenariuszy uszkodzeń

Fault	Description	Type
f_1	output choking	partly closed (50%)
f_2	level transducer failure	additive (-0.05)
f_3	positioner fault	multiplicative (0.7)
f_4	valve head or servo-motor fault	multiplicative (0.8)

Faults in different parts of the installation, including sensor, actuator and component faults (see Fig. 3 for faults placement) are proposed. Moreover, the scenarios considered are different, including additive as well as multiplicative faults. Thus, the proposed set of faults makes it possible to examine fault tolerance properties of the investigated fault accommodation procedure.

6. Experiments

6.1. Boiler modelling

The first step in the fault tolerant control design is the process modelling. To build a proper model, the training data describing the process under normal operating conditions is required. The input signal should be as much informative as possible. The training data was collected in the open loop control, where the input signal in the form of random steps with levels from the interval (0, 100) was used. Each step lasted 240 seconds. The obtained data was analysed using Discrete Fourier Transform (DFT). DFT Spectral distribution of the analysed input including 100000 samples is shown in Fig. 4. The cut-off frequency was found as $\omega_n = 0.0067$. This means that the data can be sampled with the period equal to 75s. The boiler unit is the process with slow dynamics. Filling the boiler to the level equal to 0.25m using the maximal input flow lasts approximately 300s. On this basis one can guess that distinct frequencies are lower than 0.02Hz (i.e. components with the period greater than 50s). However, during selection of the sampling time, one should keep in mind the reaction time of the fault detection block.

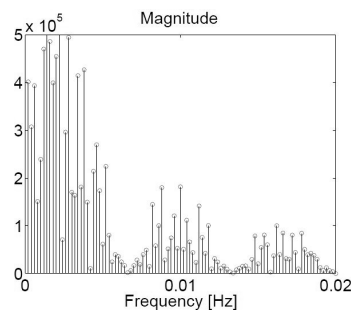


Fig. 4. DFT spectrum of random steps (100000 samples)
Rys. 4. Widmo DTF losowego sygnału skokowego (100000 próbek)

From this point of view the sampling period should be relatively short to make it possible fast detection of faults. Thus, the sampling period was set to 5s as the relatively good compromise. The training set consists of 500 samples.

The neural network state space innovation form (13) was trained for 100 epochs using Levenberg-Marquardt algorithm. The model input was the control value (CV) and the model output was the level in the boiler (L). The neural network (13) is the state observer of the system.

Tab. 3. Selection of neural network structure
Tab. 3. Wybór struktur sieci neuronowych

Network structure	Number of neurons	Model order	SSE	FPE
1	4	1	12.4918	0.000628
2	4	2	45.0943	0.0023
3	4	3	32.3772	0.0016
4	7	1	35.2229	0.0018
5	7	2	10.6031	0.000533
6	7	3	32.5675	0.0016
7	15	1	25.4094	0.0013
8	15	2	13.0330	0.000656
9	15	3	15.7853	0.000794

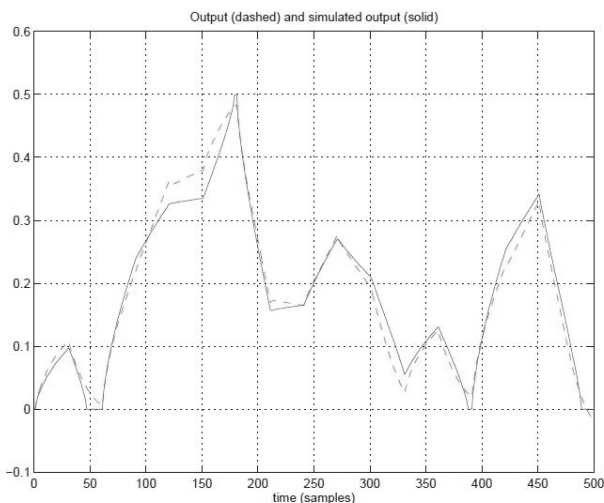


Fig. 5. Modelling results – training set: process (dashed), model (solid)
Rys. 5. Wynik modelowania – zestaw uczący: proces (linia przerywana), model (linia ciągła)

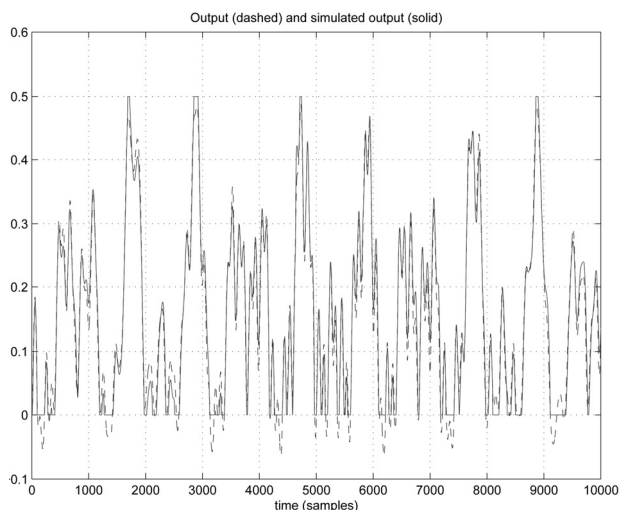


Fig. 6. Modelling results – testing set: process (dashed), model (solid)
Rys. 6. Wynik modelowania – zestaw testujący: proces (linia przerywana), model (linia ciągła)

However, by setting $e(k) = 0$ the observer is converted into the state space model (7). Different structures of the neural model were tried by changing the number of hidden neurons as well as the order of the model. The best model was selected using Sum of Squared Errors (SSE) index and Final Prediction Error (FPE) information criterion. The results of model development are presented in Table 3. Each network configuration was trained 10

times and quality indexes were averaged. The best results (marked with frames) were achieved for the second order neural model consisting of seven hidden neurons with the hyperbolic tangent activation function. For this model both quality indexes have the lowest values. The quality of the modelling for the best neural model is presented in Fig. 5. In turn, the testing of the best model is shown in Fig. 6. In both cases one can see that the output of the model follows the process output almost immediately, which proves pretty good generalization abilities of the model.

6.2. Uncertainty modelling

To build the error model, three different autoregressive models were used: the classical ARX model, Nonlinear ARX (NLARX) based on a wavelet neural network [16] and Neural Network ARX (NNARX) based on a multilayer perceptron [8]. The modelling procedure was carried out according to the algorithm presented in Subsection 3.1. The different number of input (n_a) and output (n_b) delays was tested. The quality of the obtained models is listed in Table 4. All models were compared taking into account SSE index. The best result is marked with the box. As one can see, the best model was the linear ARX with $n_a = 15$ and $n_b = 5$. Nonlinear ARX models based on neural network achieved worse results, however, the NLARX is slightly worse than the linear ARX. To definitely judge about the quality of NLARX, further investigations should be carried out checking, for example, how the type of a wavelet used influences the modelling quality.

Tab. 4. Results of error modeling
Tab. 4. Wyniki modelowania błędu modelu

Error model	delays		SSE
	n_a	n_b	
ARX	1	5	13,333
	5	1	5,763
	5	15	6,3187
	15	5	5,5946
	15	15	6,256
NLARX	1	5	45,205
	5	1	20,408
	5	15	6,1227
	15	5	6,3858
	15	15	6,381
NNARX	1	5	15441
	5	1	42,632
	5	15	53,005
	15	5	206,2735
	15	15	50,488

6.3. Fault detection

The first examined method was the simple thresholding (10). Such a method of fault detection is easy to implement and in many cases gives satisfactory results. Based on the residual ($r = y - \bar{y}$) consisting of 3000 samples, the mean value and the standard deviation of the residual were calculated: $m = 0.0016$, $\nu = 0.0093$.

Table 5 includes the thresholds values generated according to (10) with different σ levels. Using the threshold generated for $\sigma = 1$ one can achieve the fastest fault, on the other hand for $\sigma = 3$ the probability of false alarms is very low but the detection time is relatively long.

The second tested decision making method was the adaptive thresholding. In this case the statistical parameters of the residual are calculated using past observations of the residual.

Tab. 5. Thresholds for different σ values
 Tab. 5. Wartości progów dla różnych wartości σ

σ	lower threshold	upper range
1	-0,0077	0,0109
2	-0,0170	0,0202
3	-0,0263	0,0295

Over the past n samples, one can calculate the estimated values of the mean value and standard deviation. Then a threshold can be calculated according to (11). The main problem here is to choose properly the length of the time window n . If n is selected as a too small value, the threshold adapts very quickly to any change in the residual caused by any factor, e.g. disturbances, noise or a fault. If n is too large, the threshold acts in a similar way as a constant one, and the sensitivity of decision making is decreased. In order to avoid too fast adaptation to the changing residual, it is proposed to apply the weighted sum of the current and previous residual statistics. The weighted standard deviation is calculated as follows:

$$\bar{v}(k) = \zeta v(k) + (1 - \zeta)v(k - 1), \quad (20)$$

where $\zeta \in (0,1)$ is the momentum parameter controlling the influence of the current and previous values of the standard deviation value on the threshold level. In a similar way, $\bar{m}(k)$ can be calculated as:

$$\bar{m}(k) = \zeta m(k) + (1 - \zeta)m(k - 1). \quad (21)$$

In practice, in order to obtain the expected behaviour of the threshold, it is recommended to use the value of ζ slightly lower than 1, e.g. $\zeta = 0.99$.

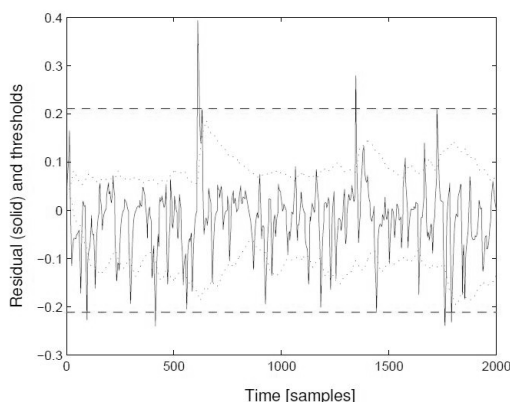


Fig. 7. Illustration of the constant (dashed) and adaptive thresholds with momentum (dotted).
 Rys. 7. Porównanie stałego (linia kreskowana) i adaptacyjnego (linia kropkowana) progowania

Fig. 7 shows the application of the constant (dashed lines) and adaptive thresholds (dotted lines) to fault detection in the boiler unit. For the adaptive thresholds the length of the time window was set to $n = 20$ and the momentum parameter $\zeta = 0.99$. It is

observable that the adaptive threshold is sensitive to changes in the residual so it enables detecting faults of small values, contrary to the constant threshold. All the presented decision making methods were compared taking into account two quality indexes: the time of fault detection t_{dt} and the number of false alarms r_{fa} [9]. The results are listed in Table 6.

The third tested decision method was the model error modelling. In order to obtain a more reliable method for threshold adaptation, one should estimate the model uncertainty taking into consideration other process variables, e.g. measurable process inputs and outputs. The robust model of the system consists of the state space model developed in Subsection 6.1 and the error model designed in Subsection 6.2. Fig. 8 illustrates the idea of the uncertainty region. The system output is marked with the solid line, the uncertainty region centre is marked with the dotted line. Using a certain significance level, confidence bands (marked with dashed lines) are generated around the centre.

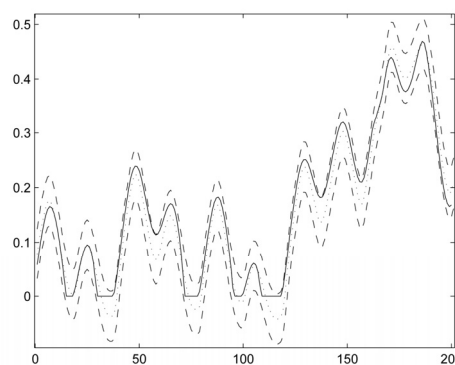


Fig. 8. Illustration of the model error modelling based decision making
 Rys. 8. Detekcja uszkodzenia z wykorzystaniem metody modelowania błędu modelu

Tab. 6. Quality of the investigated decision making methods
 Tab. 6. Porównanie jakości proponowanych metod detekcji

σ	Decision making method					
	constant threshold		adaptive threshold		model error modelling	
	t_{dt}	r_{fa}	t_{dt}	r_{fa}	t_{dt}	r_{fa}
1	20	0.5189	5	0.7600	10	0.1339
2	25	0.1333	10	0.4217	20	0.0117
3	35	0.0072	10	0.0494	25	0.0039

The best results are marked with the frames. As one can see, the best results concerning the detection time were obtained for the adaptive thresholds. However, it should be kept in mind that the sampling period is equal to 5s, then the decision making block can take a decision using 5 second length interval. Then, the detection time equal to 10s is also the acceptable result. In turn, taking into account the number of false alarms, the best results were achieved using the model error modelling technique. Taking into account both quality indexes, the best results were achieved for the model error modelling with $\sigma = 1$, which is assigned to the confidence level equal to 0.15866.

The fault detection abilities of the investigated decision making methods are shown in Figs. 9 and 10. Once again one can see that the model error modelling presents more reliable behaviour than the other considered methods.

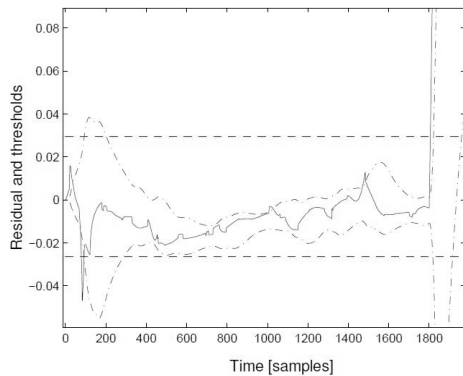


Fig. 9. Residual signal (solid), constant (dashed) and adaptive (dot-dashed) thresholds for $\sigma = 3$ in the case of a fault introduced at 1800th sample

Rys. 9. Sygnał residuum (linia ciągła), próg stały (linia kreskowana) i adaptacyjny (linia kropka-kreska) dla wartości $\sigma = 3$ w przypadku wystąpienia uszkodzenia przy 1800 próbkę

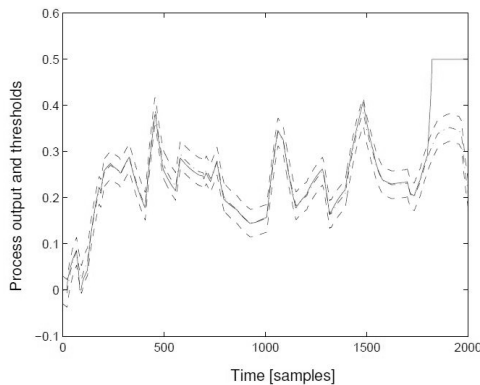


Fig. 10. Model error modelling; system output (solid) and uncertainty bands (dashed) in the case of a fault introduced at 1800th sample

Rys. 10. Modelowanie błędu modelu; odpowiedź systemu (linia ciągła) oraz progi niepewności (linia przerywana) w przypadku wystąpienia uszkodzenia przy 1800 próbkę

6.4. Fault compensation

In order to check the fault accommodation approach proposed in this paper, a number of experiments was carried out investigating the behaviour of the system control in the case of different faults introduced. The objective of the control system working with the PID controller was to keep the constant level in the boiler equal to $0.25m$. Each fault listed in Table 2 was simulated at 500th time instant. All the decision making methods investigated in the previous section were tried to design the fault detection block. Under the nominal condition, the value of the additional control u_{fa} is set to 0, but in the case of a fault its value changes according to (17), then the fault effect is compensated. Figs. 11-14 present the behaviour of the fault tolerant system in the case of faults listed in Table 2. Each chart presents the output of the healthy system (solid line), output of the faulty system without compensation (dashed line) and outputs of the compensated system (marked with different marks methods dependent).

In turn, the fault tolerant control results are given in Table 7, where J^{fc} and J^{nofc} are sums of squared tracking errors calculated for the systems with fault compensation and without fault compensation, respectively. The tracking error is defined as $e_l(k) = L(k) - y_r(k)$, where $y_r(k)$ is the reference signal.

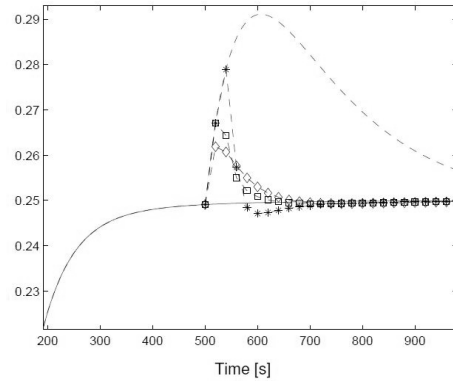


Fig. 11. System affected with the fault f_1 introduced at 500th time instant: healthy system (solid line), without FTC (dashed line), FTC with constant threshold (star marks), FTC with adaptive threshold (diamond marks), FTC with model error modelling (square marks)

Rys. 11. Wystąpienie uszkodzenia f_1 w pracy system w czasie 500 próbki: zdrowy system (linia ciągła), bez FTC (linia kreskowana), FTC z prostym progowaniem (gwiazdki), FTC z adaptacyjnym progiem (romby), FTC z modelowaniem błędu modelu (kwadraty)

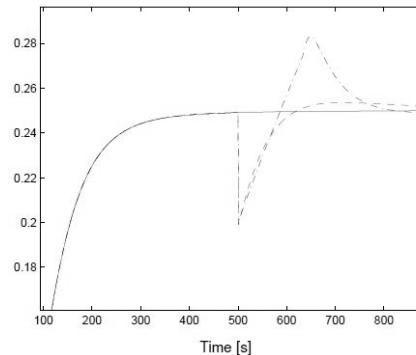


Fig. 12. System affected with the fault f_2 introduced at 500th time instant: healthy system (solid line), all decision making methods generate the same detection time and FTC (dash-dot line) behaves in the same manner for each detection method

Rys. 12. Wystąpienie uszkodzenia f_2 w pracy system w czasie 500 próbki: zdrowy system (linia ciągła), wszystkie metody detekcji wykryły uszkodzenie w tym samym czasie, przez co kompensacja z systemem ftc (linia kresko-kropokowana) w każdym wypadku wygląda tak samo

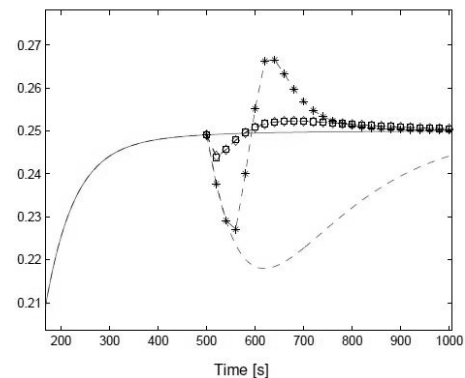


Fig. 13. System affected with the fault f_3 introduced at 500th time instant: healthy system (solid line), without FTC (dashed line), FTC with constant threshold (star marks), FTC with adaptive threshold (diamond marks), FTC with model error modelling (square marks)

Rys. 13. Wystąpienie uszkodzenia f_3 w pracy system w czasie 500 próbki: zdrowy system (linia ciągła), bez FTC (linia kreskowana), FTC z prostym progowaniem (gwiazdki), FTC z adaptacyjnym progiem (romby), FTC z modelowaniem błędu modelu (kwadraty)

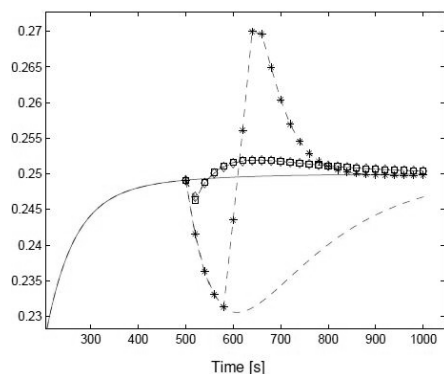


Fig. 14. System affected with the fault f_4 introduced at 500th time instant: healthy system (solid line), without FTC (dashed line), FTC with constant threshold (star marks), FTC with adaptive threshold (diamond marks), FTC with model error modelling (square marks)

Rys. 14. Wystąpienie uszkodzenia f_4 w pracy system w czasie 500 próbki: zdrowy system (linia ciągła), bez FTC (linia kreskowana), FTC z prostym progowaniem (gwiazdki), FTC z adaptacyjnym progiem (romby), FTC z modelowaniem błędu modelu (kwadraty)

Anyway, the fault accommodation scheme utilizing the adaptive thresholds or the model error modelling technique works pretty well and a fault effect is much faster compensated than using the constant threshold. Summarizing, using the robust model of the system the behaviour of the fault tolerant control may be significantly increased. All faults were quickly compensated excluding fault f_2 . In this case the system without the compensation (using only PID controller) works better.

Tab. 7. Fault tolerance quality measures

Tab. 7. Wskaźniki jakości zaproponowanego systemu FTC

Constant thresholds				
	f_1	f_2	f_3	f_4
J^{nofc}	0.3448	0.0626	0.2188	0.0756
J^{frc}	0.0234	0.1302	0.0424	0.0425
Adaptive thresholds				
	f_1	f_2	f_3	f_4
J^{nofc}	0.3448	0.0626	0.2188	0.0756
J^{frc}	0.0083	0.1302	0.0022	0.0012
Model error modelling				
	f_1	f_2	f_3	f_4
J^{nofc}	0.3448	0.0626	0.2188	0.0756
J^{frc}	0.0134	0.1302	0.0023	0.0012

7. Conclusion

The purpose of this work was to design a fault detection and accommodation system based on the robust state space neural model. The fault compensation was carried out applying the modified control law derived using the instantaneous linearization of the already trained nonlinear state space model of the system. The efficiency of the fault compensation depends on the quality of the fault detection module. This block is used to timely switch the modified control law in order to compensate a fault effect. The experiments proved that using the robust model of the system, faults could be detected vastly and reliably and the procedure of the fault accommodation could start faster. Our further research in

this area will focus on analysing the behaviour of the proposed approach in the case of disturbances acting on the system and stability analysis of the proposed fault tolerant control, which is a kind of an adaptive control scheme

This work was supported in part by the Ministry of Science and Higher Education in Poland under the grant N N514 6784 40.

8. References

- [1] Basseville M. and Nikiforov I. V. (1993): Detection of Abrupt Changes: Theory and Application, Prentice Hall, Englewood Cliffs.
- [2] Blanke M., Kinnaert M., Lunze J. and Staroswiecki M. (2006): Diagnosis and Fault-Tolerant Control, Springer, Berlin.
- [3] Chen J. and Patton R. J. (1999): Robust Model-Based Fault Diagnosis for Dynamic Systems, Kluwer, Berlin.
- [4] Czajkowski A. and Patan K. (2010): Fault detection and accommodation of the boiler unit using state space neural networks, 8th European Workshop on Advanced Control and Diagnosis, ACD 2010, Ferrara, Italy, pp. 380–385. CD-ROM.
- [5] Koj J., Zelazny M. and Kościelny J. (2005): Laboratory stands for research and didactic purposes in the area of automatic control and diagnosis, Measurements, Automatic Control, Monitoring 50(9): 261–264. special issue, in Polish.
- [6] Korbicz J., Kościelny J. M., Kowalczyk Z. and (eds) Cholewa W. (2004): Fault Diagnosis. Models, Artificial Intelligence, Applications, Springer-Verlag, Berlin.
- [7] Leszczyński M. and Syfert M. (2005): Application of fault tolerant control system for the boiler laboratory set-up, Proc. XV National Control Conference, KKA'05, Warsaw, Poland, Vol. II, pp. 175–178. in Polish.
- [8] Norgaard M., Ravn O., Poulsen N. K. and Hansen L. K. (2000): Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, London.
- [9] Patan K. (2008): Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes, Springer-Verlag, Berlin.
- [10] Patan K. (2009): Fault detection and accommodation by means of neural networks. application to the boiler unit, Proceedings of 7th International Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2009 Barcelona, Spain, 30th June – 3rd July 2009. CD-ROM.
- [11] Patan K. (2010): Local stability conditions for discrete-time cascade locally recurrent neural networks, International Journal of Applied Mathematics and Computer Science 20(1): 23-34.
- [12] Patton R. J. (1997): Fault-tolerant control: the 1997 situation (survey), Proc. IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'97, Hull, U.K., pp. 1029–1052.
- [13] Puig V. (2010): Fault diagnosis and fault tolerant control using set-membership approaches: Application to real case studies. International Journal of Applied Mathematics and Computer Science 20(4): 619-635.
- [14] Reinelt W., Garulli A. and Ljung L. (2002): Comparing different approaches to model error modelling in robust identification, Automatica 38: 787–803.
- [15] Theillol D., Cedric J. and Zhang Y. (2008): Actuator fault tolerant control design based on reconfigurable reference input, International Journal of Applied Mathematics and Computer Science 18(4): 553–560.
- [16] Zhang Q. (1997): Using wavelet networks in non parametric estimation, IEEE Transactions on Neural Networks 3: 227–236.
- [17] Zhang Y. (2007): Active fault-tolerant control systems: integration of fault diagnosis and reconfigurable control, in J. Korbicz, K. Patan and M. Kowal (Eds), Fault Diagnosis and Fault Tolerant Control, Challenging Problems of Science - Theory and Applications: Automatic Control and Robotics, Academic Publishing House EXIT, Warsaw, pp. 21–41.