Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 1, 225–237
DOI: 10.2478/v10006-012-0017-6

NONLINEAR MODEL PREDICTIVE CONTROL OF A BOILER UNIT: A FAULT TOLERANT CONTROL STUDY

KRZYSZTOF PATAN, JÓZEF KORBICZ

Institute of Control and Computation Engineering
University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland
e-mail: {k.patan,j.korbicz}@issi.uz.zgora.pl

This paper deals with a nonlinear model predictive control designed for a boiler unit. The predictive controller is realized by means of a recurrent neural network which acts as a one-step ahead predictor. Then, based on the neural predictor, the control law is derived solving an optimization problem. Fault tolerant properties of the proposed control system are also investigated. A set of eight faulty scenarios is prepared to verify the quality of the fault tolerant control. Based of different faulty situations, a fault compensation problem is also investigated. As the automatic control system can hide faults from being observed, the control system is equipped with a fault detection block. The fault detection module designed using the one-step ahead predictor and constant thresholds informs the user about any abnormal behaviour of the system even in the cases when faults are quickly and reliably compensated by the predictive controller.

Keywords: recurrent neural networks, process model, predictive control, fault detection, boiler unit.

1. Introduction

Due to the increasing requirements for high levels of system performance and reliability in the presence of unexpected changes of system functions, Fault Tolerant Control (FTC) systems have received increased attention in the last years (Blanke *et al*., 2006; Noura *et al*., 2009; Ducard, 2009; Zhang, 2007; Staroswiecki *et al*., 2007; Theillol *et al*., 2008; Sourander *et al*., 2009; Bonivento *et al*., 2004). Sensor or actuator faults, product changes and material consumption may affect the controller performance (Korbicz *et al*., 2004; Blanke *et al*., 2006). In safety-critical systems there are always defined requirements for a safe back-up in the case of failures. FTC systems may be a more effective alternative for providing the system back-up than equipment redundancy (e.g., duplication of even triplication of actuators). The demand of fault tolerant control comes also from economics. The costs of production losses due to a fault can be enormous, just like the costs of unnecessary energy consumption. From that point of view there is a strong encouragement to prolong production plant operation despite faults until a scheduled maintenance date.

The main objective of an FTC system is to maintain the current performance of the system as close as possible to the desirable one, and to preserve stability conditions in the presence of faults. The existing FTC methods can be divided into two groups: passive and active ones (Zhang, 2007). Passive approaches are designed to work with presumed failure modes. This means that they are robust against a known set of faults and their performance tends to be conservative, especially in the case of unanticipated faults. Such strategies need the existence of neither a fault detection system nor controller reconfiguration mechanisms. On the other hand, active methods react to the occurrence of system faults online and attempt to maintain the overall system stability and performance even in the case of unanticipated faults. In this case, a mandatory element is a fault diagnosis system, which provides the information about the presence of a fault, its localization and size. Additionally, the active fault tolerant control requires a reconfigurable controller and a controller reconfiguration mechanism.

An active control method that seems to be suitable for FTC is Model Predictive Control (MPC), earlier known as receding horizon control. Indeed, in MPC, the representation of both fault and control objectives is relatively simple. Some faults can be represented by modifying constraints in the MPC algorithm while other faults can be handled through modification of a system model (Joosten and Maciejowski, 2009; Camacho and Bordóns, 2004). MPC is a control method in which the

current control signal is achieved by solving online, at each sampling instant, a finite horizon optimal control problem. The optimization yields an optimal control sequence, while the first element of this sequence is used as the current control. The increasing popularity of model predictive control is due mainly to its versatility regarding coping with constraints imposed on the state and input.

Recently, FTC systems for a boiler unit were proposed by Patan (2009a; 2009b). These approaches are based on the so-called online fault approximator realized by means of a locally recurrent neural network. Consequently, using a signal representing the evolution of a fault, the control law is reconfigured in such a way as to compensate the fault effect. Unfortunately, the main difficulty related to such an approach is proper derivation of the augmented control law. The problem is especially observable in the case of nonlinear systems.

This paper presents an alternative approach where fault tolerance is achieved through a predictive control scheme. As motivated by Maciejowski (1998), predictive control has an inherent "daisy-chaining" property which allows adapting the control law to changing working conditions of the plant. Contrary to a majority of papers on fault tolerant control, which assume linear models, this manuscript uses a nonlinear model of the plant designed by means of a recurrent neural network. As significant faults push the system far from equilibrium conditions, the application of a nonlinear model is highly recommended. Furthermore, a common assumption in the academic FTC literature is that actuator and sensor faults are treated as additive disturbances on inputs and outputs. In fact, many abnormal situations cannot be modelled in this way. In the present study different types of faults are examined, not only additive ones.

The paper is organized as follows. In Section 2, basics about model predictive control and its neural network representation are described. Boiler unit specification and a set of simulated faulty scenarios are given in Section 3. Experiments including the selection of a neural model, training data and controller parameters as well as fault detection and fault compensation are provided in Section 4. Section 5 includes conclusions and final remarks.

2. Model predictive control

Model predictive control is one of the modern advanced control strategies (Maciejowski, 2002; Camacho and Bordóns, 2004). This class of control techniques has succeeded in many practical applications (Camacho and Bordóns, 2004; Breger and How, 2006; Chilin *et al.*, 2010). As a result, predictive control has received considerable attention in the last years. Predictive control algorithms are able to consider constraints imposed on both controls and process outputs (Mayne *et al.*, 2000). Moreover, the actual control is derived taking into account internal changes and interactions in the controlled process. This ability can be used to obtain a fault tolerance property of the control system. Indeed, there are many examples of fault tolerant behaviour of MPC (Maciejowski, 1998; Joosten and Maciejowski, 2009).

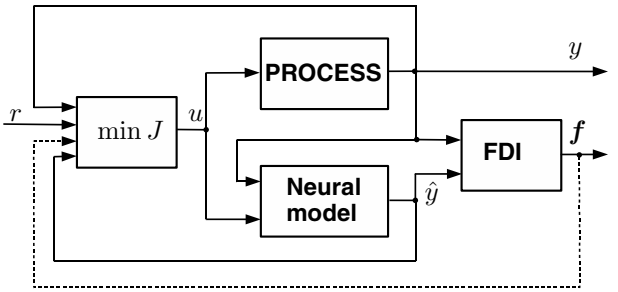

Fig. 1. Block scheme of nonlinear model predictive control with a fault diagnosis block.

MPC is a specific control strategy which uses a model of the process to derive the control signal by minimizing some objective function over a finite receding horizon. The process model is used to predict future plant outputs based on past and current outputs as well as future control signals. These are calculated through the minimization of the cost function taking into account constraints. In order to design MPC for a given problem, two elements are of crucial importance: the model of a process working in normal operating conditions and the robust optimization procedure. The block scheme of a nonlinear model predictive control is shown in Fig. 1. As this scheme uses a process model, it can be easily equipped with the Fault Detection and Isolation (FDI) block. Model-based fault diagnosis methods are widely described in the literature, (e.g., Korbicz *et al.*, 2004; Isermann, 2006; Patan, 2008). The information provided by the FDI block (localization and size of faults) can be used during the optimization procedure in order to redefine the constraints imposed on the state and inputs.

In the following sections the components of the nonlinear model predictive control scheme shown in Fig. 1 are portrayed.

2.1. Optimization criterion. The predictive control considered in this work is a modification of Generalized Predictive Control (GPC) (Clark *et al.*, 1987). The idea behind GPC is to minimize, et each iteration, the following criterion:

$$J=\sum_{i=N_1}^{N_2}(r(k+i)-\hat{y}(k+i))^2+\rho\sum_{i=1}^{N_u}(\Delta u(k+i-1))^2 \tag{1}$$

with respect to N_u future controls,

$$\boldsymbol{u}(k)=\left[\begin{array}{ccc} u(k) & \ldots & u(k+N_u-1) \end{array}\right]^T, \tag{2}$$

and subject to the constraints

$$\Delta u(k+i) = 0, \quad N_u \leq i \leq N_2 - 1, \tag{3}$$

where $r(k+i)$ is the future reference signal, $\hat{y}(k+i)$ is the prediction of future outputs, $\Delta u(k+i-1) = u(k+i-1) - u(k+i-2)$, N_1 is the minimum prediction horizon, N_2 is the prediction horizon, N_u is the control horizon, and finally ρ represents a factor penalizing changes in the control signal. For nonlinear systems the optimization problem (1) has to be solved at each sample time giving a sequence of future controls $\boldsymbol{u}(k)$, where the first element is taken to control the process. The distinct characteristic of MPC is the receding horizon. The control signal is derived in such a way as to achieve the desired behaviour of the control system in the subsequent N_2 time steps. Another important property is the control horizon $N_u < N_2$. Only the first N_u future controls are determined. From that point the control is assumed to be constant. For practical reasons, taking into account the computation burden, N_u is kept as small as possible, e.g., equal to 1 or 2.

2.2. Neural modelling. In the criterion (1), $\hat{y}(k+i)$ denotes the minimum variance i-step ahead prediction. The predictor can be obtained in two ways:

(i) by instantaneous linearization of a nonlinear model of a plant,

(ii) by successive recursion of a one-step ahead nonlinear model.

In the first case a unique solution of (1) exists and the future control signals can be calculated directly. Unfortunately, the instantaneous linearization technique suffers from some drawbacks. It relies on the linearized model, which may have a limited validity in certain regimes of the operating range. In the second case the control system uses a nonlinear model of the plant, thus the minimization of the objective function (1) has to be carried out through an iterative procedure. This kind of control system is called Nonlinear Predictive Control (NPC). The one-step ahead prediction is described by the formula

$$\hat{y}(k+1) = f(y(k), \ldots, y(k-n+1), u(k), \ldots, u(k-m+1)), \tag{4}$$

where n and m are numbers representing past outputs and inputs, respectively. The i-step ahead prediction of the plant output is then calculated by successive recursion according to

$$\hat{y}(k+i) = f(y(k+i-1), \ldots, y(k+i-n), u(k+i-1), \ldots, u(k+i-m)). \tag{5}$$

It is assumed that the measurement of the output is available up to time k. Therefore, one should substitute

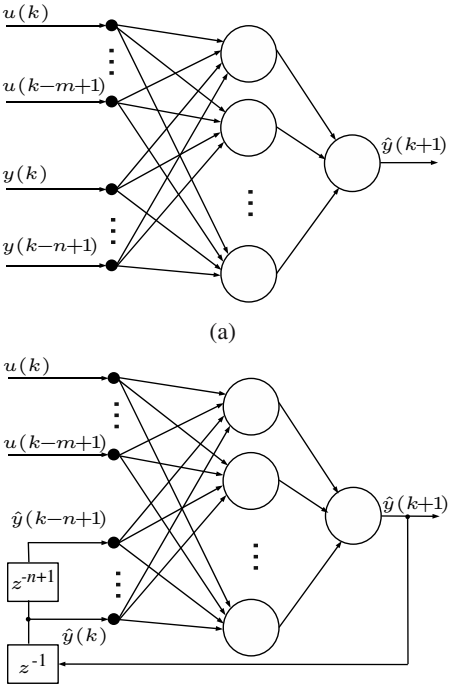

Fig. 2. Dynamic model realization: using a feedforward network (a), using a recurrent network (b).

predictions for actual measurements since these do not exist,

$$y(k+i) = \hat{y}(k+i), \quad \forall i > 1. \tag{6}$$

A nonlinear function f can be realized by an artificial neural network. Artificial neural networks provide an excellent mathematical tool for dealing with nonlinear problems (Haykin, 1999; Patan, 2008). They have an important property according to which any continuous nonlinear relation can be approximated with arbitrary accuracy using a neural network with a suitable architecture and weight parameters. Another attractive property is the self-learning ability. A neural network can extract the system features from historical training data using the learning algorithm, requiring little or no *a priori* knowledge about the process. This provides the modelling of nonlinear systems with great flexibility (Haykin, 1999).

The most frequently used neural network to model dynamic nonlinear mappings is the multilayer perceptron with tapped delay lines. This kind of neural network, which is the direct realization of (4), is depicted in Fig. 2(a). The training of such a network leads to the so-called serial-parallel identification model (Patan, 2008). The use of real plant outputs avoids many of the analytical difficulties encountered, assures neural model stability and simplifies the identification procedure. However, in order to design the i-step ahead predictor, one needs to

feed the network with past predicted outputs according to (5) and (6), and the feedforward network trained using the series-parallel identification model may be insufficient.

The neural realization of (5) is presented in Fig. 2(b). In this case the neural network is of the recurrent type and the training is carried out through the parallel identification model (Patan, 2008). However, recurrent networks suffer from stability problems as well as quite complicated training algorithms and, in general, designing an acceptable recurrent neural model is not an easy task. In the case when $i > n$, the neural network structure is something inbetween the feedforward and recurrent networks presented in Fig. 2, as it uses both measured past outputs (up to time k) and predicted outputs (for time instants $k+1, \ldots, k+i$). However, from the practical point of view, the training of such a hybrid structure is carried out in the same way as for the recurrent network depicted in Fig. 2(b).

2.3. Control law update. When a nonlinear neural network is applied as a process model, output predictions are nonlinear in the control inputs. This constitutes a complex nonlinear programming problem, which should be solved in real-time while the optimization procedure should assure fast convergence and numerical robustness. In general, the control law update at each sample time i is represented as follows:

$$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \eta^{(i)} \boldsymbol{h}^{(i)}, \tag{7}$$

where $\boldsymbol{u}^{(i)}$ is the current iterate of the future control inputs sequence, $\eta^{(i)}$ represents the step size and $\boldsymbol{h}^{(i)}$ is the search direction. The numerical algorithm suitable to solve the problem should be characterized by fast convergence, real-time processing and numerical robustness. From this point of view, gradient based methods suffer from very slow convergence. On the other hand, global optimization techniques are too complex to satisfy real-time processing. Hence, second-order optimization algorithms seem to be a reasonable choice. The standard Newton based algorithm cannot be applied due to problems with assuring positive definiteness of the Hessian. In turn, the popular Levenberg–Marquardt method cannot guarantees rapid convergence as it is not a small residual problem (Nørgaard *et al*., 2000). In this paper a modification of the Newton and Levenberg–Marquardt methods proposed by Nørgaard *et al*. (2000) is applied. The search direction $\boldsymbol{h}^{(i)}$ at the i-th sample time is derived in the following way:

$$\boldsymbol{h}^{(i)} = -\tilde{\boldsymbol{H}}^{-1} \boldsymbol{g}^{(i)}, \quad \tilde{\boldsymbol{H}} = \boldsymbol{H}^{(i)} + \lambda^{(i)} \boldsymbol{I}, \tag{8}$$

where $\boldsymbol{H}^{(i)}$ is the Hessian calculated as

$$\boldsymbol{H}^{(i)} = \left. \frac{\partial^2 J}{\partial \boldsymbol{u}^2} \right|_{\boldsymbol{u}=\boldsymbol{u}^{(i)}}, \tag{9}$$

$\boldsymbol{g}^{(i)}$ is the gradient vector represented as

$$\boldsymbol{g}^{(i)} = \left. \frac{\partial J}{\partial \boldsymbol{u}} \right|_{\boldsymbol{u}=\boldsymbol{u}^{(i)}}, \tag{10}$$

and $\lambda^{(i)} \boldsymbol{I}$ is the term proposed in the framework of the Levenberg–Marquardt algorithm. The parameter λ should be selected in such a way as to assure $\tilde{\boldsymbol{H}}$ to be positive definite. If $\tilde{\boldsymbol{H}}$ is not positive definite, λ should be increased until it is. A detailed description of the algorithm can be found in the work of Nørgaard *et al*. (2000).

3. 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. The installation is dedicated for the investigation of diagnostic methods of industrial actuators and sensors (Patan, 2009a; 2009b). 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 boiler has the form of a horizontally placed cylinder, which introduces

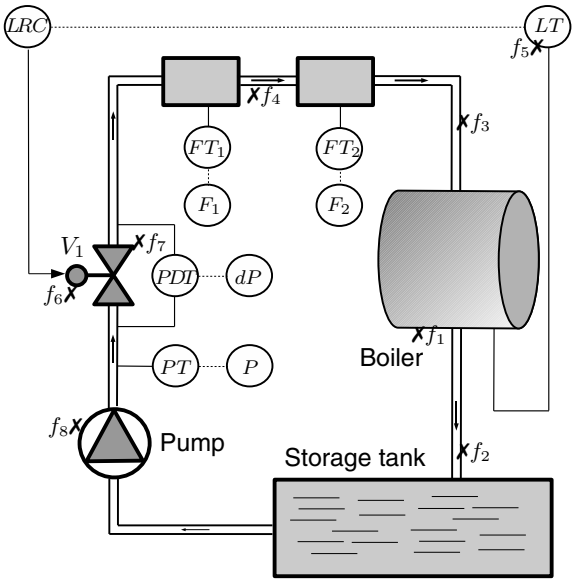

Fig. 3. Boiler unit and possible faults placement.

Table 1. Specification of process variables.

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

Table 2. Specification of faulty scenarios.

Fault	Description	Type
f_1	leakage from boiler	additive (-0.05)
f_2	outflow choking	partly closed (50%)
f_3	loss of internal pipe diameter	partly closed (50%)
f_4	leakage from pipe	additive (-1)
f_5	level transducer failure	additive (-0.05)
f_6	positioner fault	multiplicative (0.7)
f_7	valve head or servo-motor fault	multiplicative (0.8)
f_8	pump productivity reduction	multiplicative (0.8)

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. The objective of the control system is to keep the required level of the water in the boiler. During experiments, three reference signals were used:

1. the constant value
$$r(k) = 0.25, \tag{11}$$
2. the sinusoidal signal of the form
$$r(k) = \begin{cases} 0 & \text{for } k < 120, \\ \dfrac{1}{20}\sin\left(\dfrac{2\pi k}{600}\right) + \dfrac{1}{4} & \text{otherwise,} \end{cases} \tag{12}$$
3. random steps with levels from the interval $(0, 0.5)$ (each step lasts 240 seconds).

The boiler unit together with the control system was implemented in Matlab/Simulink. Simulations are 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. Faults in different parts of the installation are proposed, including sensor, actuator and component faults (see Fig. 3 for faults placement). Moreover, the scenarios considered are different, including additive as well multiplicative faults. Thus, the proposed set of faults makes it possible to examine fault tolerance properties of the investigated predictive controller in the widest range possible. The purpose of this paper is twofold: to develop the control scheme for the boiler unit based on NPC and to examine fault tolerant properties of the controller.

4. Experiments

4.1. Training data. The first step in the controller design is the process modelling. To build a proper model, the training data describing the process under normal operating conditions the required. The input signal should be as much informative as possible. It means that it should be persistently exciting of a certain order, i.e., it should contain sufficiently many distinct frequencies. The training data was collected in the open loop control, and the following input signals were used:

1. random steps with levels from the interval $(0, 100)$ (each step lasted 240 seconds),
2. PseudoRandom Binary Signal (PRBS), simulated as a sum of five sinusoids with large amplitudes limited to the range $(0, 100)$,
3. the sum of sinusoids with the frequencies 0.00066Hz, 0.0008Hz, 0.0016Hz, 0.002Hz and 0.005Hz with the amplitude equal to 0.05 each, and with the mean value of 0.25.

Each signal was analysed using the Discrete Fourier Transform (DFT). DFT spectral distributions of analysed inputs, 100000 samples each, are presented in Figs. 4(a)–(c). The boiler unit is a process with slow dynamics. Filling the boiler to the level equal to 0.25 m using maximal input flow lasts approximately 300 seconds. On this basis one can guess that distinct frequencies are lower than 0.02 Hz (i.e., components with the period greater than 50 seconds). Comparing spectra of these three input signals it is evident that the most informative one is the signal in the form of random steps. The DFT can be also used to specify the length of the training sequence. The DFT spectrum of random steps containing 10000 samples is shown in Fig. 4(d). This result clearly shows that a sequence of the length 10000 is not informative enough to be used as the training one (cf. Fig. 4(a)), and much longer sequences should be selected. On the other hand, taking into account training time, training sequences should be as short as possible. Then, the selection of the training sequence is only a compromise between the number of samples and persistent excitation properties. Taking into account these considerations, the random step sequence containing 50000 samples was selected as the training input signal.

4.2. Process modelling. In order to build a model of the process, two neural models were tried: NNARX (Neural Network AutoRegressive with eXogenous input), shown in Fig. 2(a), and the NNOE (Neural Networks Output Error) model, portrayed in Fig. 2(b). The model input was the control value (CV) and the model output was the level in the boiler (L). Preliminary experiments showed that NNARX was easier to train (i.e., the training error reached a low value), but the generalization properties were poor taking into account the prediction quality. In turn, the training process of the NNOE was more difficult to carry out but the prediction quality achieved acceptable.

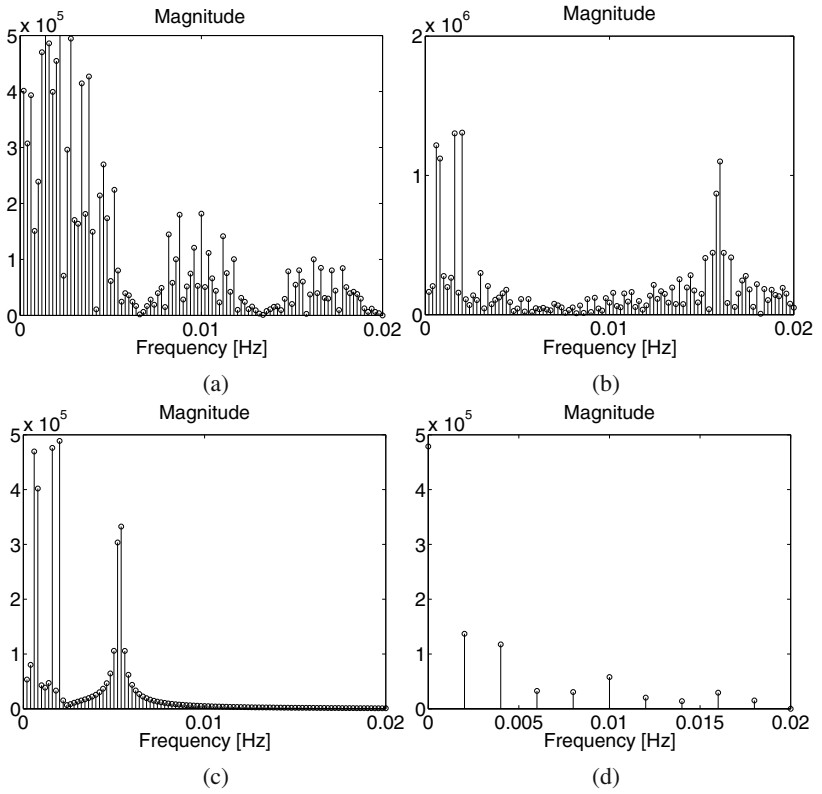

Fig. 4. DFT spectra of training signals: random steps (100000 samples) (a), PRBS (100000 samples) (b), sum of sinusoids (100000 samples) (c), and random steps (10000 samples) (d).

The testing quality of sample neural models are listed in Table 3. Both models had the same number of hidden neurons (5) and the same number of input and output delays ($m = 2$, $n = 2$). The training sequence was in the form of random steps (50000 samples). The NNARX model achieved better quality for the random steps testing sequence containing 300000 samples, but for other testing sequences its performance was worse than in the case of the NNOE model. Particularly, significant differences are observable for the constant signal. Therefore, the final model of the system was designed by means of NNOE.

The structure of the model (the number of hidden neurons, the number of past inputs and outputs) was selected experimentally using the "trial and error" method with the final prediction error information criterion to discard too complex models. The final settings are as follows: one input, one output, five hidden neurons with hyperbolic tangent activation function, one linear output neuron, the number of past inputs $m = 2$ and the number of past outputs $n = 2$. The modelling was carried out in open loop control in normal operating conditions. As the input signal, the random step function was selected in order to provide persistent excitation of the object (see Section 4.1). Using this signal, a training set containing 50000 samples was formed. The training process was carried out off-line for 100 steps with the Levenberg–Marquardt algorithm. The quality of the obtained model was checked using a testing set containing 300000 samples. The one-step ahead prediction of the selected neural model fed with random steps signal is presented in Fig. 5. The model output mimics the behaviour of the process pretty well. The sum of squared errors between the model and process outputs is equal to 157.59, and the mean squared error equal to $5.2 \cdot 10^{-4}$. In turn, the one-step ahead prediction of the neural model feeding with the sum of sinusoids described in Section 4.1 is depicted in Fig. 6. Once again it is observable that the model follows the process output almost immediately, which proves pretty good generalization abilities of the model. In this case, the sum of squared errors between the model and process outputs is equal to 204.52, and the mean squared error equal to $6.8 \cdot 10^{-4}$.

Table 3. Testing of neural models for inputs.

Model	Sum of squared errors		
	Random steps	Sum of sinusoids	Constant value
NNARX	127.7	231.12	654.03
NNOE	157.59	204.51	313.31

4.3. Control. In order to design model predictive control for the boiler unit using a neural model, the values of control parameters should be properly selected. These parameters are as follows:

1. The minimum prediction horizon N_1: As the fundamental model of the controlled process is defined in the form of the one-step ahead predictor, this parameter is usually set to 1 and this value was used during the experiments.

2. The prediction horizon N_2: Taking into account the computation burden this value should be kept as small as possible. Taking into account "slow dynamics" of the process, through the "trail and error" procedure, a value equal to 15 was selected as quite a good compromise.

3. The control horizon N_u: Following the reasons presented in Section 2.1, the value of the control horizon was set to 2.

4. The penalty factor ρ: This is the crucial parameter penalizing changes in the control signal. It significantly influences the optimization process. Many values were tried from the interval $[0, 10]$. The best control results were observed for $\rho = 0.0003$.

5. The maximum number of iterations: For each time instant the algorithm calculated the optimum value of the control signal. Taking into account real-time processing, the maximum number of iterations was set to 5. It was observed that such a value is sufficient to solve the optimization problem (1) with acceptable quality.

6. Constraints: For the problem considered, the only constraint is imposed on the value of the control signal. The control signal should have a value from the interval $[0, 100]$.

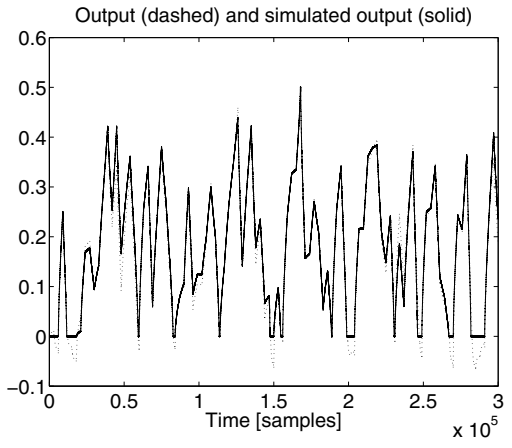

Fig. 5. Testing of the neural model on random steps.

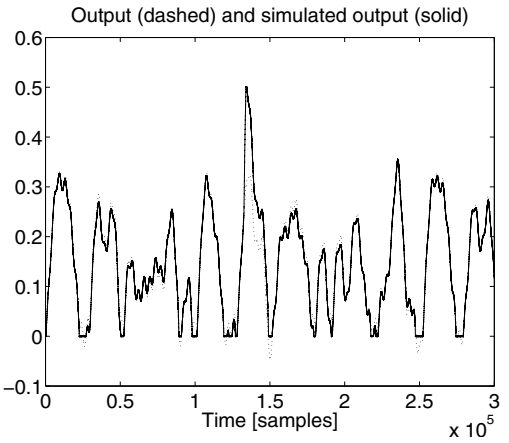

Fig. 6. Testing of the neural model on the sum of sinusoids.

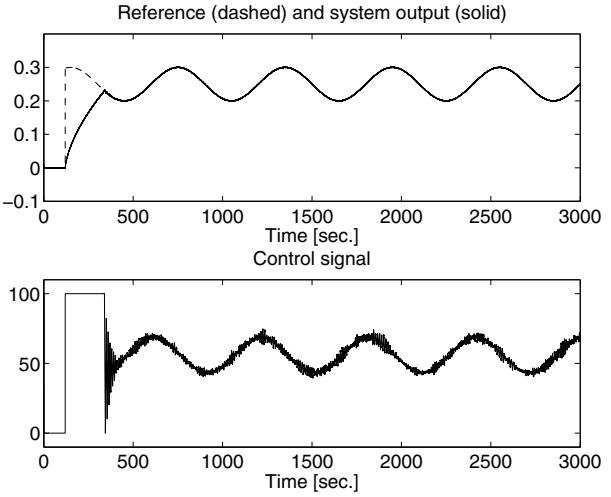

Fig. 7. Control of the boiler unit. Tracking of the sinusoidal signal (upper panel) and the control signal (lower panel).

The performance of the proposed predictive control is shown in Figs. 7 and 8. First, let us consider the behaviour of the control system in the case of the reference (12) (see Fig. 7). As the reference signal changes from 0 to 0.3, the optimization procedure calculates the control signal of a high value equal to its maximum. Then, due to decreasing the cost criterion, the control signal is decreased and, after approximately 170 seconds, the plant starts to follow the reference almost immediately (Fig. 7, the upper panel). The lower panel in Fig. 7 presents the control signal. When the output of the process reaches the reference after some decreasing oscillations, the control becomes a signal of sinusoidal character. There are small changes visible in the control signal, but the optimization procedure keeps the signal as close as possible to the sinusoidal, while tracking the reference is carried out with a pretty high quality.

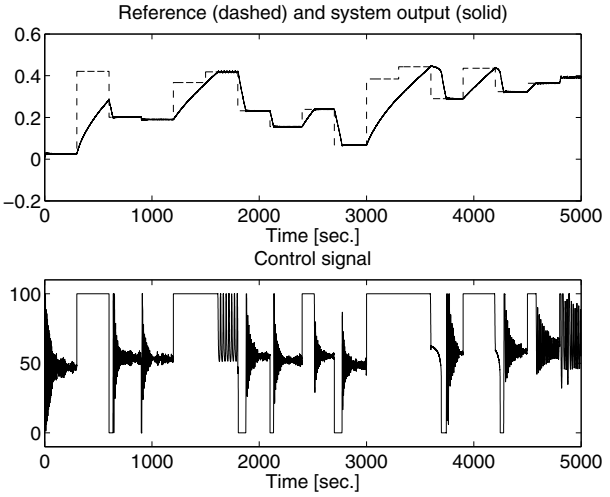

Fig. 8. Control of the boiler unit. Tracking of random steps (upper panel) and the control signal (lower panel).

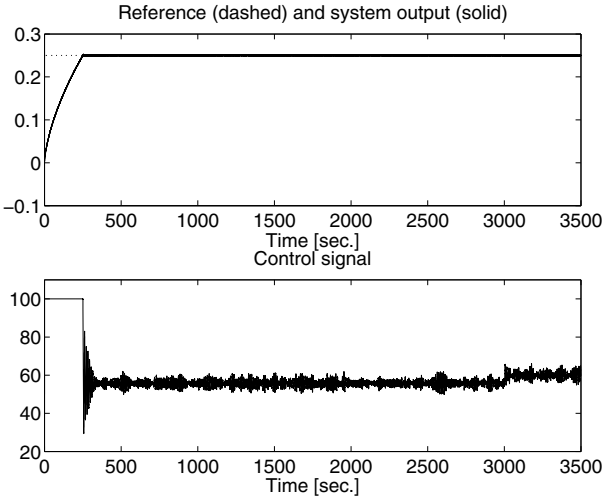

Fig. 9. Fault tolerant control: fault f_8 and the constant value as the reference.

In turn, the performance of the control system in the case of the reference signal in the form of random steps is presented in Fig. 8. Once again, one can clearly observe the performance of the predictive controller. Every time the reference changes, the controller adapts the control signal to these changes (Fig. 8, the upper panel). When the reference is increased, the predictive control works similarly as in the previously analyzed case. When the reference is significantly decreased, the controller generates the control signal of a high value, and then after some oscillations the control sets up at a certain value (see the lower panel of Fig. 8). Such behaviour is more clearly observable when reference changes are larger. This behaviour can be explained as a direct consequence of the optimization procedure, which adapts the control to the changing operating point, based on the nonlinear multimodal cost function. In general, the quality of the proposed predictive controller is much better than in the case of the classical PID one (Patan, 2009a; 2009b).

4.4. Fault tolerance. The fault tolerance property of the proposed predictive controller was investigated introducing several faults listed in Table 2 and observing the control quality. Each fault was simulated at the 3000-th second as a permanent fault.

Fault tolerant control results are presented in Table 4, where $\Delta e = \|e_l^F(k)\|^2 - \|e_l^N(k)\|^2$, $e_l^N(k)$ and $e_l^F(k)$ are tracking errors in normal operating conditions and a faulty situation, respectively, the tracking error is defined as $e_l(k) = L(k) - r(k)$. All faults were properly compensated excluding the fault f_3. In this case, even the maximum inflow cannot compensate the fault effect in any way.

Fault tolerant control in the case of the fault f_8 is presented in Figs. 9 and 10. As one can see in the upper part of Fig. 9, the fault effect is not observable in the system output. Even if the reference is changing in time (e.g., sinusoidal signal), the fault is hidden from being observed (the upper panel of Fig. 10). This is due to a very fast reaction of the predictive controller to pump productivity reduction, which immediately increased and optimally adapted the control signal to the changing working conditions of the plant (see the lower panels of Figs. 9 and 10).

In turn, fault tolerant control results in the case of the fault f_5, for different reference signals, are presented in Figs. 11 and 12. Here, effects caused by the fault are clearly observable in both cases (see the upper parts of Figs. 11 and 12). Fortunately, due to flow transducer failure, the predictive controller increased the control signal fast and after some time the fault was completely compensated (see the lower panels of Figs. 11 and 12).

In most cases, the norm of the tracking error increases, a little bit contrary to normal operation conditions. This means that the predictive controller works pretty well and the fault effect is not observable taking into accout the process output. The highest difference be-

Table 4. Fault tolerance quality measures.

Fault	$\|e_l(k)\|^2$	SEE	Δe
no fault	0.0381	0.0014	–
f_1	0.0381	0.0014	0
f_2	0.0733	0.0054	0.0352
f_3	12.1116	146.6907	12.0735
f_4	0.0843	0.0071	0.0462
f_5	2.2123	4.8941	2.1742
f_6	0.0692	0.0048	0.0311
f_7	0.0570	0.0053	0.0189
f_8	0.0395	0.0016	0.0014

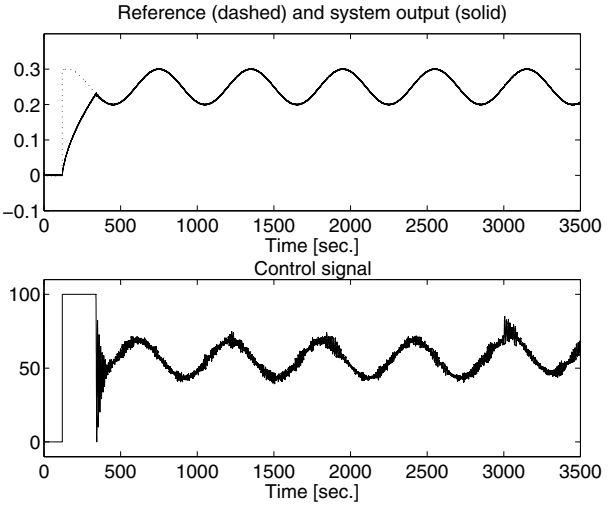

Fig. 10. Fault tolerant control: fault f_8 and the sinusoidal signal as the reference.

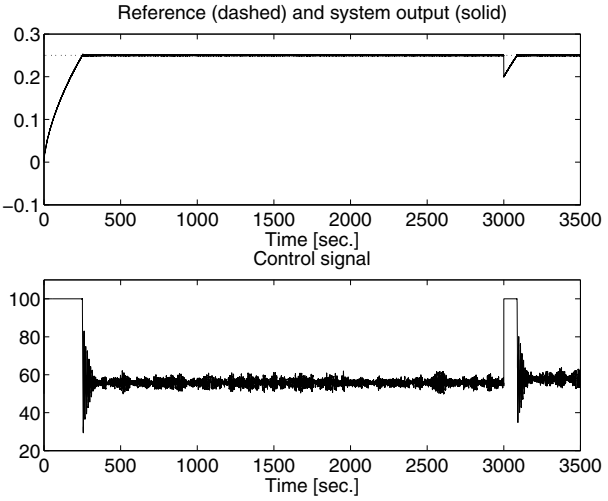

Fig. 11. Fault tolerant control: fault f_5 and the constant value as the reference.

tween norms of the tracking error calculated for normal operating conditions and a faulty situation is observable for the fault f_5 (excluding the fault f_3, which cannot be compensated). This is a sensor fault of a additive character. However, after some time the controller handled this fault in order to maintain the control objectives.

An interesting situation is observed for the fault f_1. The quality indexes calculated for this fault are exactly the same as for normal operating conditions (see the first two rows of Table 4). The explanation of this phenomenon is that the fault f_1 has a neglecting impact on the control system, i.e., the leakage from the boiler is of a very small value. The fault effect caused by f_1 is also discussed in Section 4.6.

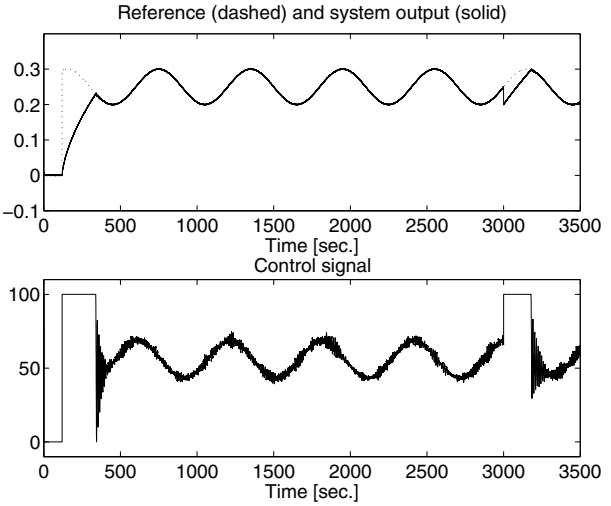

Fig. 12. Fault tolerant control: fault f_5 and the sinusoidal signal as the reference.

4.5. Fault compensation. To evaluate the fault compensation property of the proposed predictive controller, a tolerance region around the reference was defined. The tolerance region was determined taking into account a

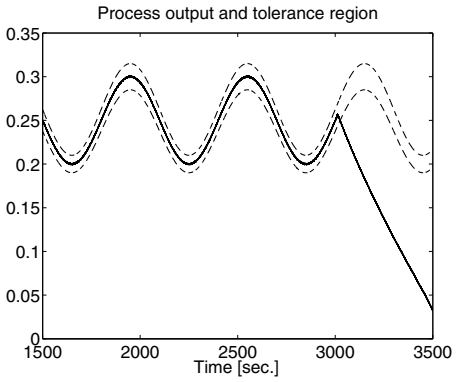

Fig. 13. Illustration of compensation time derivation: fault f_3.

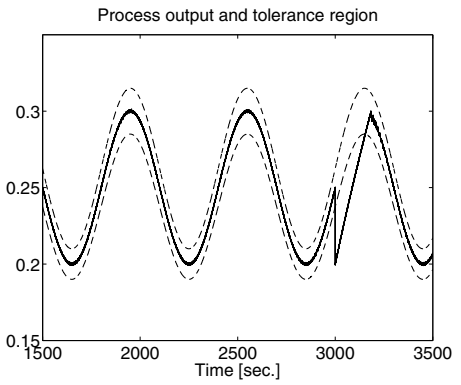

Fig. 14. Illustration of compensation time derivation: fault f_5.

$\pm 5\%$ tolerance around the reference, and thus the tolerance interval was assumed to be

$$[0.95r(k), 1.05r(k)]. \tag{13}$$

On this basis, fault compensation time was calculated as a difference between the time in which the system response is inside the interval (13) and the time of fault occurrence. An illustration of fault compensation time derivation for the fault f_3 is presented in Fig. 13. In this case the fault effect was not compensated. The output of the process exceeded the tolerance region permanently. This is due to the fact that after f_3 occurred in the system the maximal inflow to the boiler was lower than the outflow. In turn, fault compensation time derivation for the fault f_5 is shown in Fig. 14. This time the fault effect was compensated after 154.75 seconds.

Fault compensation times for each fault considered are presented in the 3-rd column of Table 5 for the reference (12), and the 5-th column of Table 5 for the reference (11), where t_c represents fault compensation time in seconds, and NC (Not Compensated) means that a fault was not compensated. As one can observe, almost all faults were immediately compensated.

4.6. Fault detection. The automatic control system can hide faults from being observed (see Table 5, the third column, positions with $t_c = 0$, or Figs. 9 and 10). Therefore, to diagnose a system condition, a fault detection and isolation algorithm is required. The previously designed one-step ahead predictor can be easily used to construct model-based fault detection system. Comparing the system output with that of the one-step ahead predictor, the so-called residual signal can be generated.

To carry out fault detection, a decision making technique based on constant thresholds is used. If the residual is smaller than a certain threshold value, a process is considered healthy, otherwise it is considered faulty. In general, two thresholds can be used to detect both increasing and decreasing trends in the residual. Thus, the thresholds can be calculated as

$$T_u = t_\alpha v + m, \quad T_l = t_\alpha v - m \tag{14}$$

Table 5. Fault detection and fault compensation measures.

Fault	Reference (12)		Reference (11)	
	t_d [s]	t_c[s]	t_d[s]	t_c[s]
f_1	ND	–	ND	–
f_2	5.9	0	1.15	0
f_3	19.45	NC	3.55	NC
f_4	12.35	0	9.1	0
f_5	0.05	154.75	0.05	64.55
f_6	69.65	0	7.7	0
f_7	42.15	0	5.7	0
f_8	33.85	0	7.7	0

where T_u and T_l represent the upper and lower threshold value, respectively, t_α is $\mathcal{N}(0,1)$ tabulated value assigned to $1-\alpha$ confidence interval, v represent standard deviation of the residual and m is mean value of the residual (Patan, 2008). The decision making of the fault f_5 using constant thresholds in the case of the reference (11) is illustrated in Fig. 15. The statistics of the residual were as follows: $m = -0.0333$, $v = 3.5 \cdot 10^{-4}$. Assuming $\alpha = 0.01$, the threshold values are $T_u = -0.0324$ and $T_l = -0.0342$. Before the time of fault occurrence (before the 3000-th time instant), the residual is inside the uncertainty region determined by upper and lower thresholds. When the fault occurred in the plant, the residual changed its value. In the case considered here the fault was immediately detected as it was a sensor fault changing significantly the measured value of the boiler level.

Figure 16 shows a similar study for the sinusoidal reference (12). In this case, the residual has larger standard deviation ($m = -0.0202$, $v = 0.002$) then the uncertainty region is wider ($T_u = -0.0155$ and $T_l = -0.0249$). In spite of that, the fault was reliably detected. Results of fault detection in the form of detection time t_d for the entire set of faults are presented in the 2-nd and 4-th columns of Table 5. The abbreviation ND (Not Detected) means that a fault was not detected by the fault diagnosis block. Detection time is defined as a period of time needed for

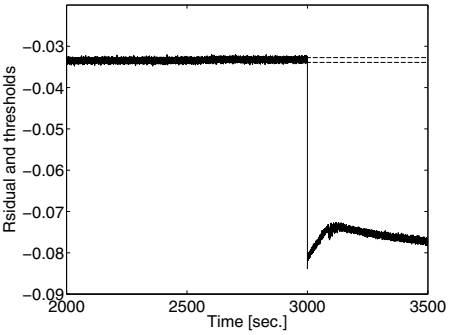

Fig. 15. Residual signal for the constant reference (11): fault f_5.

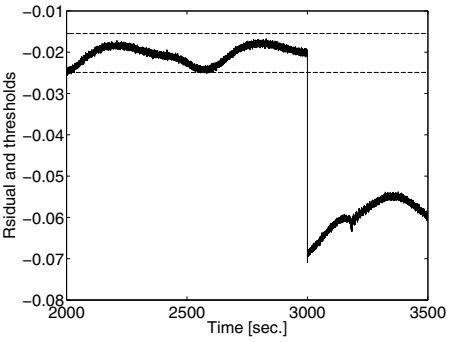

Fig. 16. Residual signal for the reference (12): fault f_5.

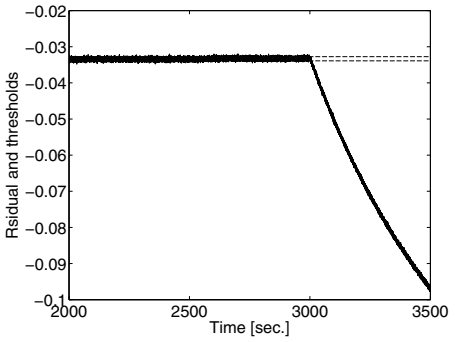

Fig. 17. Residual signal the constant reference (11): fault f_8.

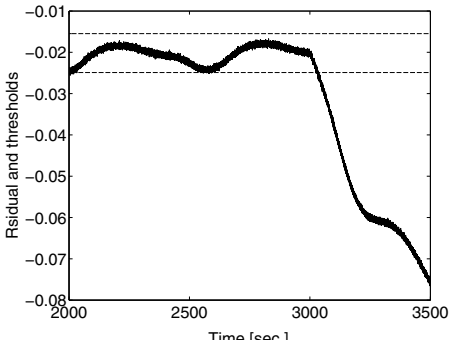

Fig. 18. Residual signal for the reference (12): fault f_8.

the detection of a fault measured from the fault start-up time to a permanent, true decision about the fault, i.e., a fault is signalled only if the residual permanently exceeds the uncertainty region. In this way, the occurrence of false alarms is minimized.

It is obvious that almost all faults were detected (excluding f_1) even in the situations when the fault effect was not observable (entries with t_c equal to zero). The only fault f_1 was not detected. This fault is of additive character but with a very small value. In fact, the fault does not affected the residual in a significant way, so it was impossible to observe the time of fault occurrence. Only a more exact model of the system makes it possible to detect this fault. Figures 17 and 18 present the residual signal for the faulty scenario f_8 in the cases of the constant and sinusoidal references, respectively. The fault f_8 is of multiplicative character, so residual changes its value not as quickly as in the case of f_5, which is the additive scenario. In general, analysing the results, it is observable that the system needs more time to detect multiplicative faults than additive ones.

Decision making about faults can be much more reliable when adaptive thresholds are applied. There are plenty of methods available, including the CUSUM algorithm, set-membership identification or model error modelling. This problem, however, is out of the scope of this paper. Another problem not raised in this work is fault isolation and identification, which is very important taking into account redefining the constraints imposed on both the system state and control. Moreover, sensor faults should be compensated differently than actuator faults. For example, in the case of the fault f_5, the control signal value should be kept the same as before fault occurrence, because such a fault does not change the water level in the boiler. Unfortunately, the control system immediately reacts to the change in the system output, which was a direct consequence of the wrongly measured water level in the boiler. Fault isolation and identification block will render it possible to compensate the measured level in the boiler without changing the control. Our further research directions will focus on these issues.

5. Concluding remarks

The paper describes the realization of nonlinear model predictive control for a boiler unit. The predictive controller was constructed using the model of the process based on the recurrent neural network while the control law was optimized using a modified version of the Newton algorithm. The proposed controller works pretty well, but some problems with selecting the proper control parameters can be encountered. Especially the penalty parameter ρ seems to be very important from the point of view of control quality. Other parameters may significantly influence the computational burden.

The proposed predictive control system possesses very good fault tolerant properties. Almost all faults considered were properly compensated, so the control system tried to maintain the performance of the system as close as possible to the desirable one. However, the predictive controller can hide faults from being observed. Therefore, the fault detection block was designed giving information about abnormal working conditions of the plant. Then, the engineer can perform proper preventing and maintenance actions. Unfortunately, the sensor fault f_5 should be treated in a different way than, for example, actuator faults. To cope with this problem, fault isolation and identification should be carried out. Based on the information provided by the fault isolation and identification block, the value wrongly measured by the transducer could be changed to a proper one without affecting the control signal. Future research will be focused on these aspects as well as on searching for an automatic procedure for selecting the parameter ρ and on stability issues of the control system in the case of faults.

Acknowledgment

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

References

Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M. (2006). *Diagnosis and Fault-Tolerant Control*, Springer, Berlin.

Bonivento, C., Isidori, A., Marconi, L. and Paoli, A. (2004). Implicit fault-tolerant control: Application to induction motors, *Automatica* **40**(3): 355–371.

Breger, L. and How, J.P. (2006). Nonlinear model predictive control technique for unmaned air vehicles, *Journal of Guidance, Control and Dynamics* **29**(5): 1179–1188.

Camacho, E.F. and Bordóns, C. (2004). *Model Predictive Control*, Springer-Verlag, Berlin.

Chilin, D., Liu, J., de la Pẽna, D.M., Christofides, P.D. and Davis, J. F. (2010). Detection, isolation and handling of actuators faults in distributed model predictive control, *Journal of Process Control* **20**(9): 1059–1075.

Clark, D.W., Mothadi, C. and Tuffs, P.S. (1987). Generalized predictive control—Part I: The basic algorithm, *Automatica* **23**(2): 137–148.

Ducard, G.J.J. (2009). *Fault-tolerant Flight Control and Guidance Systems. Practical Methods for Small Unmanned Aerial Vehicles*, Advances in Industrial Control, Springer, London.

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation,* 2nd Edn., Prentice-Hall, Upper Saddle River, NJ.

Isermann, R. (2006). *Fault Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*, Springer-Verlag, New York, NY.

Joosten, D.A. and Maciejowski, J. (2009). MPC design for fault-tolerant flight control purposes based upon an existing output feedback controller, *Proceedings of 7th International Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2009, Barcelona, Spain,* (on CD-ROM).

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

Maciejowski, J.M. (1998). The implicit daisy-chaining property of constrained predictive control, *Applied Mathematics and Computer Science* **8**(4): 695–711.

Maciejowski, J. (2002). *Predictive Control with Constraints*, Prentice-Hall, Harlow.

Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality, *Automatica* **36**(6): 789–814.

Nørgaard, M., Ravn, O., Poulsen, N. and Hansen, L. (2000). *Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag, London.

Noura, K., Theilliol, D., Ponsart, J.C. and Chamseddine, A. (2009). *Fault Tolerant Control Systems. Design and Practical Applications*, Advanced in Industrial Control, Springer, London.

Patan, K. (2008). *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*, Lecture Notes in Control and Information Sciences, Vol. 377, Springer-Verlag, Berlin.

Patan, K. (2009a). Fault detection and accommodation by means of neural networks. application to the boiler unit, *Proceedings of the 7th International Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2009, Barcelona, Spain,* (on CD-ROM).

Patan, K. (2009b). Locally recurrent networks for fault approximation and accommodation, *in* Z. Kowalczuk (Ed.), *Diagnosis of Processes and Systems*, Pomeranian Science and Technology Publishers PWNT, Gdańsk, pp. 263–270.

Sourander, M., Vermasvuori, M., Sauter, D., Liikala, T. and Jämsä-Jounela, S.-L. (2009). Fault tolerant control for a dearomatisation process, *Journal of Process Control* **19**(7): 1091–1102.

Staroswiecki, M., Yang, H. and Jiang, B. (2007). Active fault tolerant control based on progressive accommodation, *Automatica* **43**(12): 2070–2076.

Theillol, D., Cédric, 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, DOI: 10.2478/v10006-008-0048-1.

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.

Krzysztof Patan received the M.Sc. degree in electrical engineering from the Technical University of Zielona Góra, Poland, in 1996, the Ph.D. degree in machine design and exploitation from the Warsaw University of Technology, Poland, in 2000, and the D.Sc. degree in electrical engineering from the University of Zielona Góra in 2009. Currently, he is an associate professor at the Institute of Control and Computation Engineering, University of Zielona Góra. His research include artificial neural networks, especially recurrent networks, neural modelling and identification of nonlinear dynamic systems, model based fault diagnosis, fault tolerant control systems, optimization techniques and optimal experimental design. He has published more than 100 technical papers (17 in international journals). He is the author of two monographs, e.g., *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes* (Springer, 2008). Currently, he is a member of the Commission of Engineering Cybernetics of the Polish Academy of Sciences, Poznań Branch.

Józef Korbicz has been a full-rank professor of automatic control at the University of Zielona Góra, Poland, since 1994, and a corresponding member of the Polish Academy of Sciences since 2007. He currently heads the Institute of Control and Computation Engineering of his home university. His research interests include computational intelligence, fault detection and isolation, and control theory. The primary aim of his research group is to contribute to the diagnosis of dynamical systems. He has published more than 360 technical papers, 58 of which in international journals. He is a co-author of eight monographs and textbooks and a co-editor of four books, e.g., *Fault Diagnosis. Models, Artificial Intelligence, Applications* (Springer-Verlag, 2004) and *Modeling, Diagnostics and Process Control. Implementation in the DiaSter System* (Springer, 2010). He served as the IPC chairman of the IFAC Symposium *SAFEPROCESS*, Beijing, China, 2006, and of the Workshop on *Advanced Control and Diagnosis*, *ACD*, Zielona Góra, Poland, 2009. He is a senior member of the IEEE, and a member of the IFAC SAFEPROCESS TC.

Received: 28 January 2011
Revised: 4 July 2011