Journal of Physics: Conference Series **659** (2015) 012034 doi:10.1088/1742-6596/659/1/012034

Design of sensor and actuator multi model fault detection and isolation system using state space neural networks

Andrzej Czajkowski
Institute of Control and Computation Engineering, University of Zielona Góra,
ul. Podgórna 50, 65–246 Zielona Góra, Poland

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

Abstract. This paper deals with the application of state space neural network model to design a Fault Detection and Isolation diagnostic system. The work describes approach based on multi-model solution where the SIMO process is decomposed into simple models (SISO and MISO). With such models it is possible to generate different residual signals which later can be evaluated with simple thresholding method into diagnostic signals. Further, such diagnostic signals with the application of Binary Diagnostic Table (BDT) can be used to fault isolation. All data used in experiments is obtain from the simulator of the real-time laboratory stand of Modular Servo under Matlab/Simulink environment.

1. Introduction

Recently, it has been observed an increasing development of the Fault Diagnosis (FD) methods for the Fault Tolerant Control (FTC) system desing purposes [3]. It is straightly connected to the advantages of the systems which can maintain current performance as close to the desirable one, and preserve stability conditions in the presence of faults [17]. Faults and equipment failures directly affect the performance of the control system and can result in large ecomomic losses and violation of the safety regulations. During the fault tolerant control system design, the basic problem is the early detection and isolation of possible faults. The proper isolation of the faults is crucial during FTC system design. The reason behind that statement is that the different control strategy needs to be applied in case of sensor faults and completely different behaviour is needed when actuator fault occurs. The main objective here is to have FDI methods as fast and as accurate as possible. Such assumptions are crucial to further design a control system which can provide the acceptable corrections of the control signal in case of fault. The paper focuses on two aspects of the FDI design, first is stage of system modelling and second, the application of the process models to residual signal generation and evaluation. One of the tools which are heavily exploited in this task are Artificial Neural Networks (ANN) [2, 14, 15]. In this work to construct the models of the system the so called State Space Neural Networks (SSNN) are applied. Such neural model of the system was successfully applied to Model Predictive Control of Twin Rotor Aerodynamical System in the previous work of the author [4]. The SSNN model is then used to carry out the fault detection by analysing the residual and fault isolation is done with the application of Binary Diagnostic Table. The methodology presented in the paper is tested on the example of a Modular Servo in case of both actuator and sensor faults.

The paper is organized as follows, Section 2 presents a general desciption of the Modular Servo System and provides information about faulty scenarios considered. The state space neural networks are described in Section 3. Section 4 presents a FDI algorithm, while experimental results are included in Section 5.

2. Modular Servo

In this work a Modular Servo System (MSS) laboratory stand is considered [7]. The main concept of the MSS is to create a rapid and direct path from control system design to hardware implementation. The MSS supports the real-time design and implementation of advanced control methods using MATLAB/Simulink enviroment. The MSS consists of the Inteco servomechanism and open-architecture software environment for real-time control experiments. A scheme of considered DC motor is presented in Fig. 1.

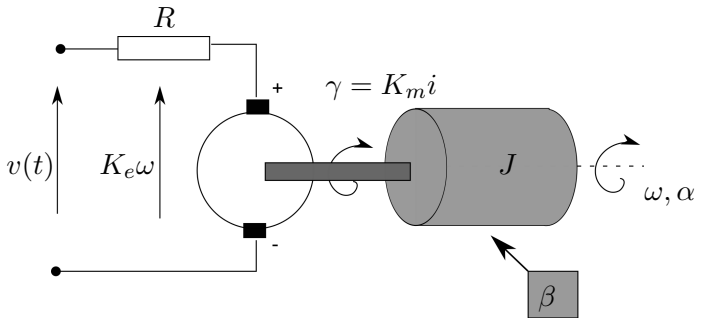

Figure 1. Diagram of DC motor.

The system is described by two classical equations, the electrical one:

$$v(t) = Ri(t) + K_e\omega(t) \tag{1}$$

and mechanical as follows:

$$J\dot{\omega}(t) = K_m i(t) - \beta\omega(t) \tag{2}$$

where $v(t)$ is the input voltage, $i(t)$ is the armature current, $\omega(t)$ is the angular velocity of the rotor, R is the resistance of armature winding, J is the moment of inertia of the moving parts, β is the damping coefficient due to viscous friction, $K_e\omega(t)$ is the back EMF and $K_M i(t)$ is the electromechanical torque.
The system used in this paper is described by the following nonlinear state space equations:

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= c(u - g(x_2)) \end{aligned} \tag{3}$$

where state variable x_1 and x_2 are the angle α (position of the shaft) and angular velocity ω, respectively. The control applied is in PWM signal form so it is assumed that dimensionless control signal u is scaled input voltage $u(t) = v(t)/v_{max}$ and it satisfies $|u(t)| \leq 1$. The function g is the inverted steady state characteristics of the system, which needs to be determined experimentally.

The system can be classified as a multivariable (SIMO) because it has two measurable state variables and one control variable. This allows to simulate three different fault scenarios which are described in the Table 1.

Table 1. Specification of faulty scenarios.

Fault	Description	Type
f_1	DC motor power loss	multiplicative (0.7)
f_2	Tacho-Generator failure	additive (-30)
f_3	Angle positioner failure	additive (-30)

3. State Space Neural Network

The crucial part in FDI design process is the model of the plant which is used for residual generation. In this work the State Space Neural Network (SSNN) which are a very important class of dynamic neural networks are used [16, 12]. The structure of such network is depicted in Fig. 2. 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.

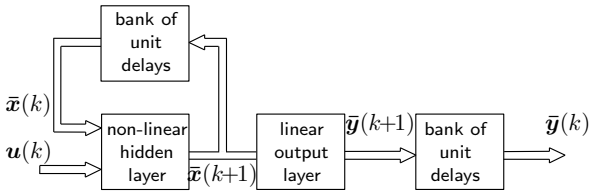

Figure 2. Block scheme of the state space neural network.

Let $\boldsymbol{u}(k) \in \mathbb{R}^n$ be the input vector, $\bar{\boldsymbol{x}}(k) \in \mathbb{R}^q$ - the output of the hidden layer at time k, and $\bar{\boldsymbol{y}}(k) \in \mathbb{R}^m$ - the output vector. The state space representation of the neural model is described by the equations

$$\begin{aligned} \bar{\boldsymbol{x}}(k+1) &= \bar{g}(\bar{\boldsymbol{x}}(k), \boldsymbol{u}(k)) \\ \bar{\boldsymbol{y}}(k) &= \boldsymbol{C}\bar{\boldsymbol{x}}(k) \end{aligned}, \tag{4}$$

where $\bar{g}(\cdot)$ is a nonlinear function characterizing the hidden layer, and $\boldsymbol{C}$ represents synaptic weights between hidden and output neurons. Introducing the weight matrix between input and hidden layers $\boldsymbol{W}^u$ and the matrix of recurrent links $\boldsymbol{W}^x$, the representation (4) can be rewritten in the following form:

$$\begin{aligned} \bar{\boldsymbol{x}}(k+1) &= h(\boldsymbol{W}^x\bar{\boldsymbol{x}}(k) + \boldsymbol{W}^u\boldsymbol{u}(k)) \\ \bar{\boldsymbol{y}}(k) &= \boldsymbol{C}\bar{\boldsymbol{x}}(k) \end{aligned}, \tag{5}$$

where $h(\cdot)$ stands for the activation function of the hidden neurons. In the most cases, the hyperbolic tangent activation function is selected giving pretty good modelling results. For the state space model the outputs of hidden neurons which constitute feedbacks 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 feedforward network with tapped delay lines). In spite of this inconvenience, state space models have a number of advantages, contrary to other recurrent neural networks ([13], [18]):

- The number of states (model order) can be selected independently from 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 kind of faults including multiplicative and additive ones.

Presented advantages of State Space Neural Networks make such structure very interesting and promising in the solving the fault compensation problem. Also the class of nonlinear state space models is strongly evaluated in different scientific approaches as a nominal model (e.g [9]).

4. Fault Detection and Isolation

In this paper the FDI approach is based on the residual evaluation. To achieve the goal of early detection and proper localisation of faults it is needed to design a process model. In this paper both the sensor and actuator faults are taken into consideration. To distinguish one group from another it is proposed to use a multi-model solution. To solve this problem let us consider a nonlinear dynamic system governed by the following state equation:

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

where $g(\cdot,\cdot)$ is a process working at the normal operating conditions, $\boldsymbol{x}(k)$ is a state vector and $\boldsymbol{u}(k)$ is the control. In case of the considered servomechanism system, the output $\boldsymbol{y}(k)$ consists of two variables – the position angle α and angular velocity ω. For diagnostic purposes it is proposed to decompose this SIMO system into the following SISO and MISO neural models:

$$\begin{aligned} \boldsymbol{x}_{ua}(k+1) &= g_{ua}(\boldsymbol{x}_{ua}(k), u(k)) \\ y_{ua}(k) &= \boldsymbol{C}\boldsymbol{x}_{ua}(k). \end{aligned} \tag{7}$$

which models the relation between voltage input u and angle position of the rotor shaft y_{ua}, through identified with SSNN function g_{ua}. This model further will be refereed as model 1 or input-angle model (uamodel).

$$\begin{aligned} \boldsymbol{x}_{uv}(k+1) &= g_{uv}(\boldsymbol{x}_{uv}(k), u(k)) \\ y_{uv}(k) &= \boldsymbol{C}\boldsymbol{x}_{uv}(k) \end{aligned} \tag{8}$$

which models the relation between voltage input u and angular velocity of the rotor y_{uv}, through identified with SSNN function g_{uv}. This model further will be refereed as model 2 or input-velocity model (uvmodel).

$$\begin{aligned} \boldsymbol{x}_{va}(k+1) &= g_{va}(\boldsymbol{x}_{va}(k), \omega(k)) \\ y_{va}(k) &= \boldsymbol{C}\boldsymbol{x}_{va}(k) \end{aligned} \tag{9}$$

which models the relation between measured angular velocity ω and angle position of the rotor shaft y_{va}, through identified with SSNN function g_{va}. This model further will be refereed as model 3 or velocity-angle model (vamodel).

$$\begin{aligned} \boldsymbol{x}_{uav}(k+1) &= g_{uav}(\boldsymbol{x}_{uav}(k), u(k), \alpha(k)) \\ y_{uav}(k) &= \boldsymbol{C}\boldsymbol{x}_{uav}(k) \end{aligned} \tag{10}$$

which models the relation between voltage input u together with measured angular velocity ω and angle position of the rotor shaft y_{uav}, through identified with SSNN function g_{uav}. This model further will be refereed as model 4 or input-angle-velocity model (uavmodel).

$$\begin{aligned} \boldsymbol{x}_{uva}(k+1) &= g_{uva}(\boldsymbol{x}_{uva}(k), u(k), \omega(k)) \\ y_{uva}(k) &= \boldsymbol{C}\boldsymbol{x}_{uva}(k) \end{aligned} \tag{11}$$

which models the relation between voltage input u together with measured angle position of the rotor shaft α and angular velocity of the rotor y_{uva}, through identified with SSNN function g_{uva}. This model further will be refereed as model 5 or input-velocity-angle model (uvamodel).

The models are realised by the means of the SSNN and are summarised in the Table 2.

Next, the angular velocity ω and angle position α can be measured as the system outputs and

name	inputs	outputs
model 1 (uamodel)	u	α
model 2 (uvmodel)	u	ω
model 3 (vamodel)	ω	α
model 4 (uavmodel)	u,α	ω
model 5 (uvamodel)	u,ω	α

Table 2. Diagnostic state space models

used for the residual calculation as follows:

$$r_1(k) = \alpha(k) - y_{ua}(k) \tag{12}$$

$$r_2(k) = \omega(k) - y_{uv}(k) \tag{13}$$

$$r_3(k) = \alpha(k) - y_{va}(k) \tag{14}$$

$$r_4(k) = \omega(k) - y_{uav}(k) \tag{15}$$

$$r_5(k) = \alpha(k) - y_{uva}(k) \tag{16}$$

To evaluate calculated residuals to fault detection and isolation, in this paper a simple thresholding method is applied [1]. The diagnostic signals s_i for each model are defined as follows:

$$s_i = \begin{cases} 0 : r_i > Tl_i \ \wedge \ r_i < Tu_i \\ 1 : r_i < Tl_i \ \vee r_i > Tu_i \end{cases}, \tag{17}$$

where $i \in \{1, 2, 3, 4, 5\}$ and Tu_i, Tl_i are upper and lower thresholds respectively. The thresholds are calculated using the so-called 3σ method using collected data during work of the system in nominal conditions.

Finally, the BDT can be created and applied for fault detection and isolation [8]. The designed BDT is presented in Table 3.

When all diagnostic signals are equal to the assumed level for the specific fault, then the fault alarm is raised. If any diagnostic signal is equal 1 then a unknown fault is signalled.

Table 3. Binary Diagnostic Table

	s_1	s_2	s_3	s_4	s_5
f_1	1	1	0	1	1
f_2	0	1	1	1	1
f_3	1	0	1	0	1

5. Experimental results

5.1. Neural Modelling

The first step in the design of fault diagnosis system is the process modelling. The proper model is crucial for residual generation and evaluation. As was shown in section 4 in this paper, the approach based on multiple models is used. The models described in section 4 were designed by the means of the SSNN. To design the training data multiple sinusoids were mixed with random signals. Such control signal gives a very rich data which characterise the whole operating range of the plant. The angular velocity operating range was equal to ±200 and angle was assumed as ±2000 rads. The collected outputs signals from encoder sensor and tacho-generator were normalised before the training process. The obtained data were divided into training and testing sets and resampled from 0.01s to 0.1s. During normal work of FDI system models outputs are linearly interpolated to match the sampling of the plant. The networks were trained for 100 epochs using Levenberg-Marquardt algorithm and the selection of the proper network structure is based on the criterion of the Sum of Squared Errors (SSE) index. The tested structures of specific models and training results are shown in table 4. The best structures and noted with frames.

Table 4. Selection of neural network structures.

		Model 1		Model 2		Model 3		Model 4		Model 5	
Model order	**No. neurons**	**Training SSE**	**Testing SSE**	**Training SSE**	**Testing SSE**	**Training SSE**	**Testing SSE**	**Training SSE**	**Testing SSE**	**Training SSE**	**Testing SSE**
1	2	126.2	279.4	1.58	1.80	0.18	61.21	1.58	1.80	0.1849	86.9
1	3	92.5	297.2	0.742	1.9e+04	0.0057	0.20	1.35	1.87	0.049	0.080
1	4	198.6	278.2	0.786	7.1e+03	0.0048	0.0256	0.26	1.54	0.1279	103.4
1	5	140.6	220.8	0.582	1.01	0.0037	0.11	0.108	0.220	0.0042	0.0593
1	6	124.3	198.0	0.652	3.3e+03	0.0041	0.12	0.107	0.221	0.2944	68.5
2	2	87.1	122.6	2.30	3.13	0.13	14.74	2.19	3.55	0.1525	19.5
2	3	0.77	1.47	0.0384	0.0729	0.0014	0.013	1.95	3.19	0.0103	6.55
2	4	0.64	0.72	0.0295	0.057	6.9e-04	0.0070	1.42	2.61	0.0017	0.0179
2	5	0.73	1.40	0.0299	0.0494	5.5e-04	6.7e-04	0.024	0.075	5.1e-04	0.0976
2	6	1.24	87.01	0.0397	0.0961	8.6e-04	9.6e-04	0.023	0.12	0.0010	0.0291
3	3	86.0	102.23	2.01	2.62	0.034	23.02	0.12	0.66	3.6993	216.1
3	4	1.12	14.40	0.28	0.402	11.35	252.13	1.64	3.36	1.4086	283.7
3	5	84.8	105.08	0.13	0.23	3.4e-04	0.0068	0.58	2.16	0.1877	95.24
3	6	8.3e+3	8.2e+03	3.34	2.2e+03	0.0018	0.0164	1.39	1.62	0.4683	162.05

Summarising the results, the best structures used in further experiments were shown in the Table 5.

5.2. Fault Management

The designed models of the plant allows to carry out the FDI experiments in which system is controlled by PID controller, tuned by the manufacturer of the plant. The control task was

Table 5. Selected neural network structures

	Model order	No neurons
model 1	2	4
model 2	2	5
model 3	2	5
model 4	2	5
model 5	2	4

to follow the reference angle signal, which changes between two values. The models are used for residual generation and simple thresholding method is used to evaluate residuals to obtain diagnostic signals. The calculated thresholds together with means and standard deviations are shown in Table 6. Three faults scenarios – one actuator and two sensor which are described in

Table 6. Thresholds calculation results.

	mean	σ	T_l	T_u
model 1	0.0132	7.8625	-23.5743	23.6006
model 2	0.2975	3.2198	-9.3619	9.9568
model 3	-0.0235	0.0165	-0.0730	0.0260
model 4	0.4268	3.2956	-9.4601	10.3137
model 5	-0.0563	0.2087	-0.6823	0.5697

Table 1, were considered. As shown in Figs. 3, 4 5 and 6 the actuator fault f_1 is very minimal and do not influence the work of the system as the PID controller compensate it easily (faults of greater magnitude should be diagnosed with even higher efficiency). In case of tacho-generator fault f_2 it does not influence the work of the control system but in the case of encoder fault f_3 which is used as the control feedback the fast detection and appropriate action is crucial.

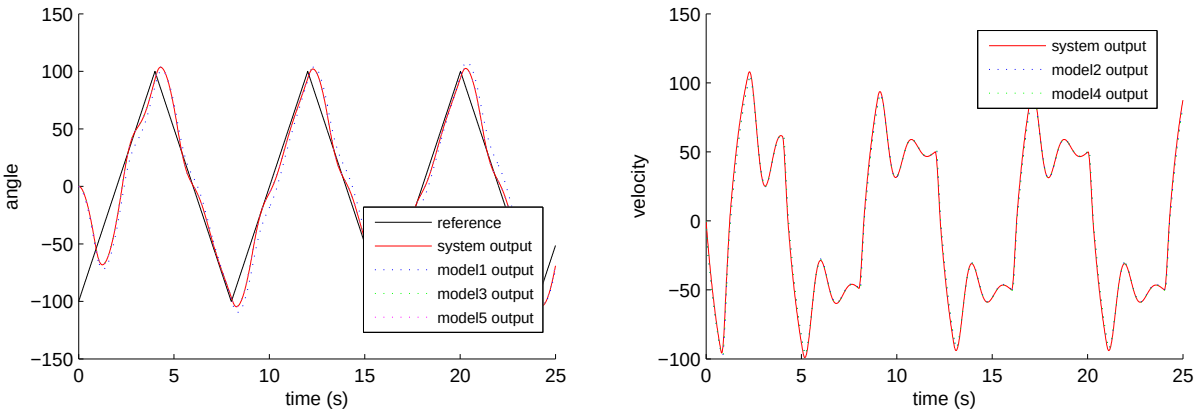

Figure 3. Outputs of the system and models in nominal conditions.

All of faults were introduced at the 14th second of simulation. The residuals were calculated and together with thresholds are presented in Figs. 7 and 8.

In all three scenarios as one can see the diagnostic system worked quite efficiently and it was easy to detect and isolate fault. The performance indexes in the form of false detection rate (r_{fd}), time of fault detection (t_{dt}), true fault detection rate (r_{td}), time of fault isolation (t_{ti}), false

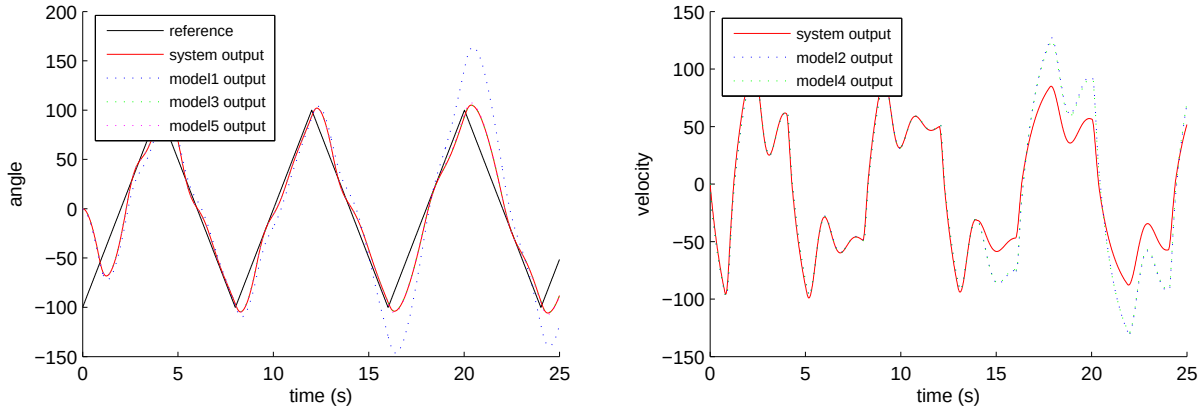

Figure 4. Outputs of the system and models in case of f_1.

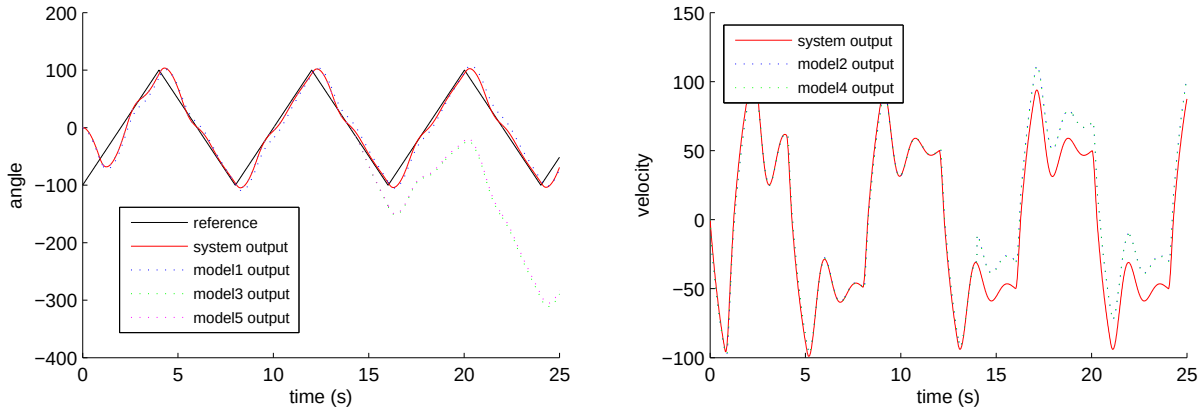

Figure 5. Outputs of the system and models in case of f_2.

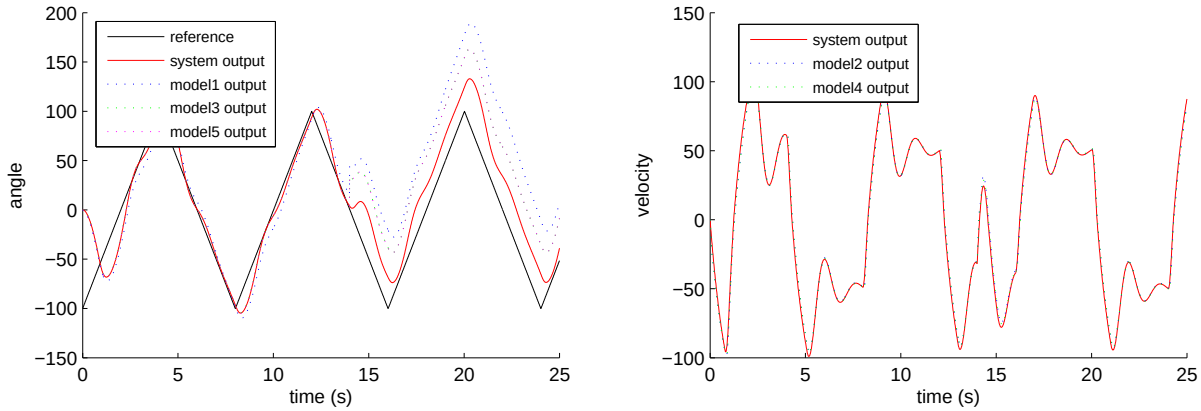

Figure 6. Outputs of the system and models in case of f_3.

isolation rate (r_{fi}) and true isolation rate (r_{ti}) were calculated according to formulas defined in [13]. Indexes are shown in Table 7. The obtained results are quite satisfying, especially in case of sensor faults where fault detection is instant and isolation rate is over 99%. Such high

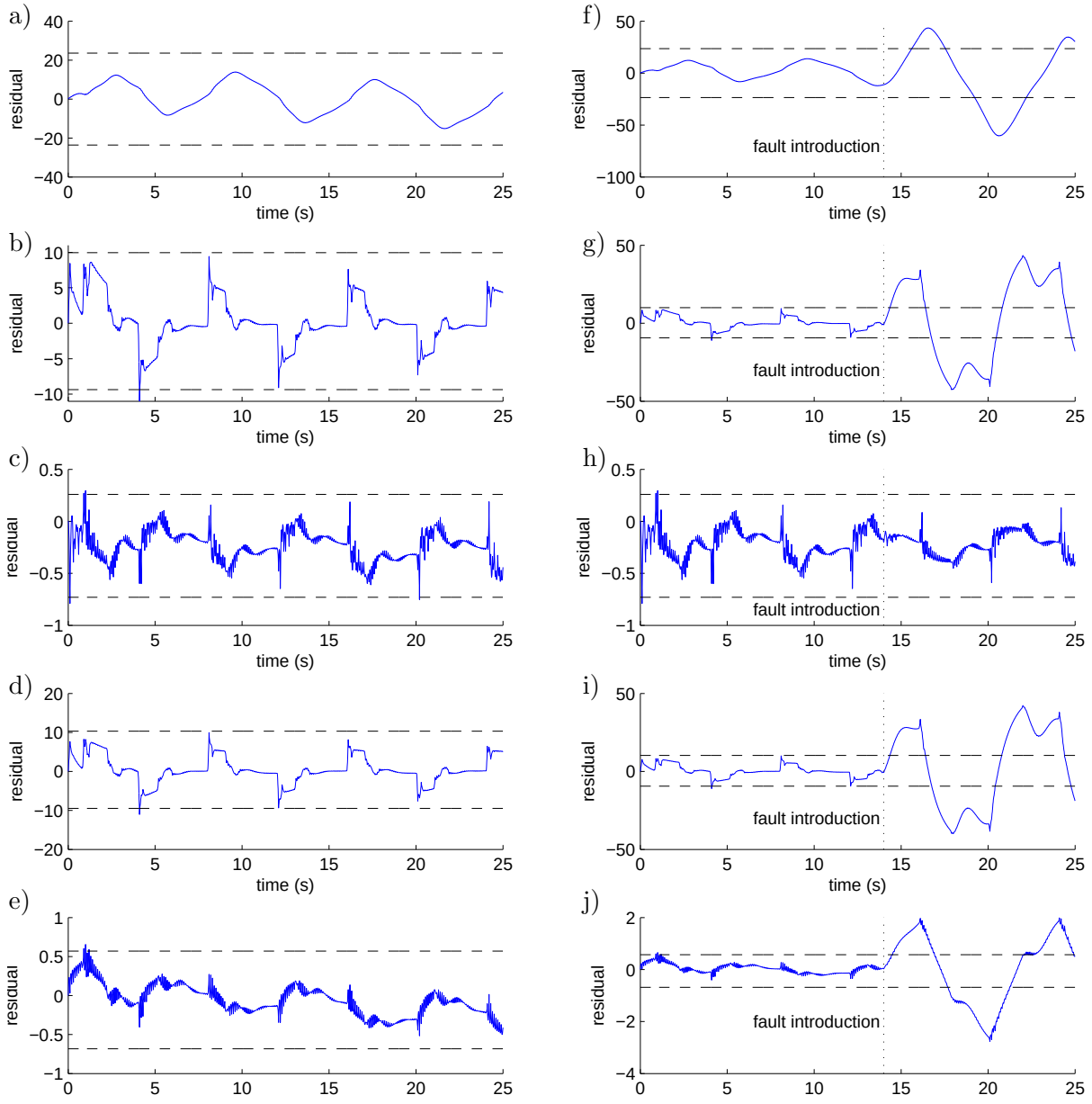

Figure 7. Residuals generated and thresholds with model 1 (a), model 2 (b), model 3 (c), model 4 (d) and model 5 (e) in nominal conditions and with model 1 (f), model 2 (g), model 3 (h), model 4 (i) and model 5 (j) in case of fault f_1.

Table 7. Diagnostic performance indexes

	r_{fd}	t_{dt}	r_{td}	t_{ti}	r_{fi}	r_{ti}
no fault	0.0032	-	-	-	-	-
f_1	0.0043	37	0.9673	164	0	0.2861
f_2	0.0043	1	1	4	0	0.9982
f_3	0.0043	1	1	2	0	0.9973

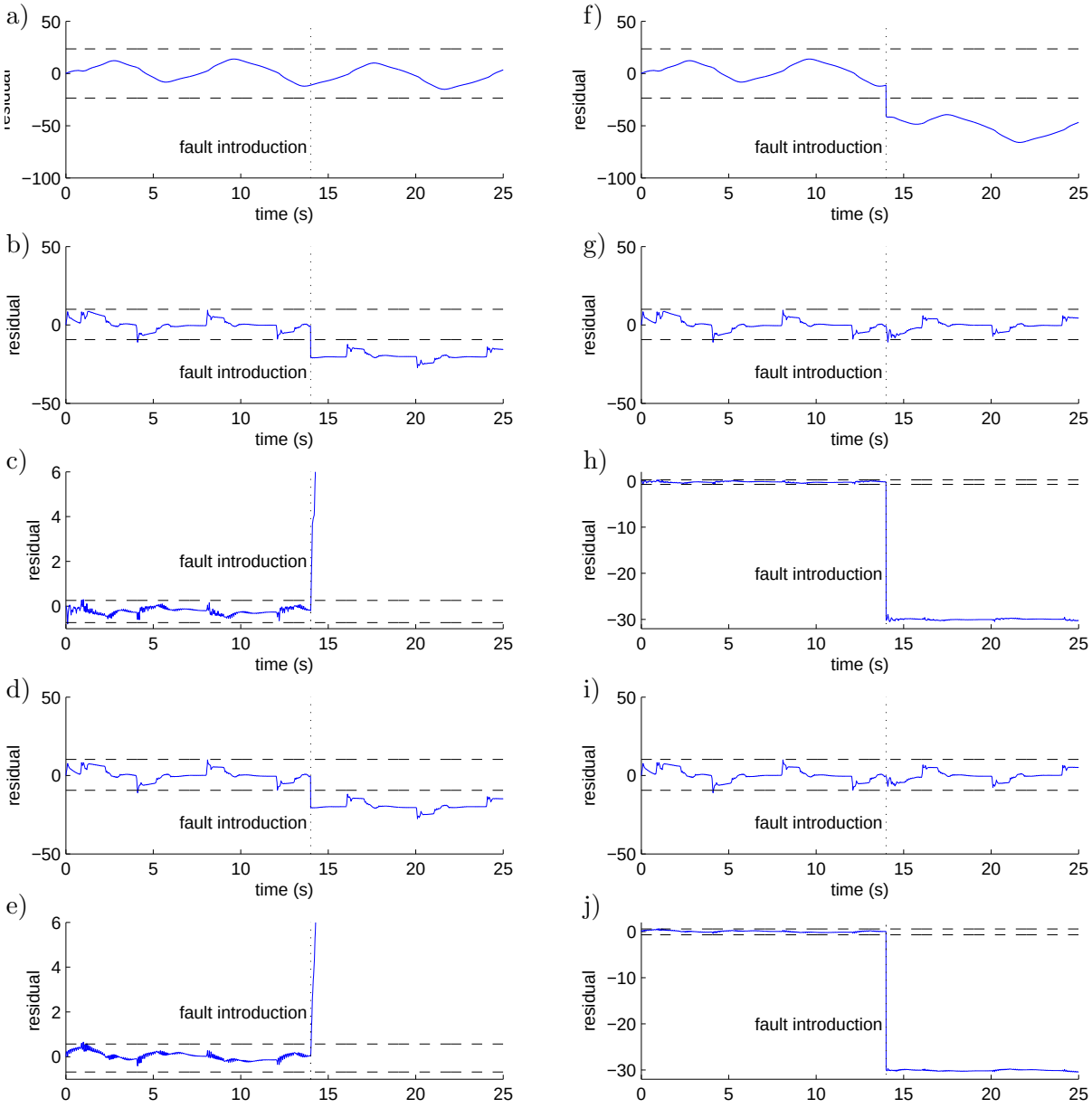

Figure 8. Residuals and thresholds generated with model 1 (a), model 2 (b), model 3 (c), model 4 (d) and model 5 (e) in case of fault f_2 and with model 1 (f), model 2 (g), model 3 (h), model 4 (i) and model 5 (j) in case of fault f_3.

performance is caused mostly by the model 3 (vamodel) which reacts very fast with great change of residual. The actuator fault is also immediately detected (after 0.3s) but the isolation is quite poor with only 28% rate. This problem can be addressed with the application of diagnostic submatrices or neural classifier and our future work will consider this approaches. The last conclusion concerns false alarms which rate is very low but such events can be disastrous in future design of fault tolerant control where inappropriate fault compensation action can occur. This will be addressed in future work with robust fault detection by application of the so-called MEM technique (model error modelling) used earlier by the authors in [6, 5]. What is most important the fault false isolation rate in each scenario is equal to 0 so that means the faults

are no mistaken with each other.

6. Conclusions

In this work the complete sensor and actuator fault detection and isolation diagnostic system was designed. As it was shown the SSNN can be used very easily and successfully in application of such diagnostic system. The obtained residual signals allows for very fast and accurate fault detection. The so-called methods of simple thresholding and Binary Diagnostic Tables are very important tools in the process of residual evaluation and fault isolation.

Our future work will be focused on achieving the better results by the replacement of the simple thresholding method by robust fault detection with MEM in the diagnostic system. Also, more accurate fault isolation by applying diagnostic submatrices will be investigated.

Acknowledgments

This work was supported by the National Science Centre in Poland under the grant UMO-2012/07/N/ST7/03316.

References

[1] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, Englewood Cliffs, 1993.

[2] Piotr Bilski and Jacek Wojciechowski. Artificial intelligence methods in diagnostics of analog systems. *International Journal of Applied Mathematics and Computer Science*, 24(2):271–282, 2014.

[3] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and J. Schröder. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] A. Czajkowski and K. Patan. Model predictive control of the two rotor aero-dynamical system using state space neural networks with delays. In M. Kowal eds. J. Korbicz, editor, *Intelligent systems in technical and medical diagnostics*, Advances in Intelligent Systems and Computing : Vol. 230, pages 113–124. Springer-Verlag, Berlin - Heidelberg, 2013.

[5] A. Czajkowski and K. Patan. Robust fault detection by means of echo state neural network. *Advanced and Intelligent Computations in Diagnosis and Control*, 386:341, 2015.

[6] Andrzej Czajkowski, Krzysztof Patan, and Mirosław Szymański. Application of the state space neural network to the fault tolerant control system of the plc-controlled laboratory stand. *Engineering Applications of Artificial Intelligence*, 30:168–178, 2014.

[7] INTECO. *Modular Servo System - User's Manual*. http://www.inteco.com.pl/Docs/Servo_um.pdf, 2014.

[8] J. Korbicz, J. M. Koscielny, Z. Kowalczuk, and W. (eds) Cholewa. *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer-Verlag, Berlin, 2004.

[9] Pangao Kou, Jianzhong Zhou, Changqing Wang, Han Xiao, Huifeng Zhang, and Chaoshun Li. Parameters identification of nonlinear state space model of synchronous generator. *Engineering Applications of Artificial Intelligence*, 24(7):1227 – 1237, 2011.

[10] M. Luzar, M. Witczak, and C. Aubrun. Robust $\mathcal{H}_\infty$ sensor fault diagnosis with neural networks. In *Intelligent Systems in Technical and Medical Diagnosis.*, pages 125–136. Springer, Berlin, 2013.

[11] M. Luzar, M. Witczak, and P. Witczak. Neural-network based robust FTC: Application to wind turbines. In *Artificial Intelligence and Soft Computing)*, pages 97–108. Springer, Berlin, 2014.

[12] M. Norgaard, O. Ravn, N. K. Poulsen, and L. K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, 2000.

[13] K. Patan. *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*. Springer-Verlag, Berlin, 2008.

[14] K. Patan. Neural network-based model predictive control: Fault tolerance and stability. *Control Systems Technology, IEEE Transactions on*, 23(3):1147–1155, May 2015.

[15] K. Patan and J. Korbicz. Nonlinear model predictive control of a boiler unit: a faul tolerant control study. *International Journal of Applied Mathematics and Computer Science*, Vol. 22(no 1):225–237, 2012.

[16] Ole Sørensen. *Neural networks in control applications*. PhD thesis, 1994.

[17] R.F. Stengel. Intelligent failure-tolerant control. In *Intelligent Control, 1990. Proceedings., 5th IEEE International Symposium on*, pages 548–557 vol.1, Sep 1990.

[18] Jesús M. Zamarreño Cosme and Pastora Vega. State space neural network. properties and application. *Neural Networks*, 11(6):1099–1112, 1998.