Fault Tolerant Control Using Model Reference Adaptive Controller Method

Santosh S. Raghuwanshi
Assistant Professor
Dept. of Electrical & Electronics
PIES, Indore, India

Yamini Mokhariwale
Assistant Professor
Dept. of Electrical & Electronics
PIES, Indore, India

Ankita Singh
Assistant Professor
Dept. of Electrical & Electronics
PIES, Indore, India

ABSTRACT

Stimulated by the growing demand for improving system performance and reliability, fault-tolerant system design has been receiving significant attention. This paper proposes a new fault-tolerant control methodology using model reference adaptive controller method based on the learning capabilities of neural networks or fuzzy systems. Moreover, the fault-tolerance ability of the adaptive controller has been further improved by exploiting information estimated from a fault-diagnosis unit designed by interfacing multiple models with an expert supervisory scheme.

Keywords

Fault-tolerant control; Adaptive control; Neural/fuzzy control; Fault diagnosis; Artificial Intelligence Methods

1. INTRODUCTION

An increasing demand on products quality, system reliability, and plant availability has allowed that engineers and scientists give more attention to the design of methods and systems that can handle certain types of faults. In addition, the global crisis creates more competition between industries and plant shutdowns are not an option because they cause production losses and consequently lack of presence in the markets; primary services such as power grids, water supplies, transportation systems, and communication and commodities production cannot be interrupted without putting at risk human health and social stability.

On the other hand, modern systems and challenging operating conditions increase the possibility of system failures which can cause loss of human lives and equipments; also, some dangerous environments in places such as nuclear or chemical plants, set restrictive limits to human work. In all these environments the use of automation and intelligent systems is fundamental to minimize the impact of faults.

The most important benefit of the Fault Tolerant Control (FTC) approach is that the plant continues operating in spite of a fault, no matter if the process has certain degradation in its performance. This strategy prevents that a fault develops into a more serious failure. In summary, the main advantages of implementing an FTC system are (Blanke et al., 1997):

Plant availability and system reliability in spite of the presence of a fault.

i) Prevention to develop a single fault in to a system failure.
ii) The use of information redundancy to detect faults instead of adding more hardware.
iii) The use of reconfiguration in the system components to accommodate a fault.
iv) FTC admits degraded performance due to a fault but maintain the system availability.
v) Is cheap because most of the time no new hardware will be needed.

Some areas where FTC is being used more often are: aerospace systems, flight control, automotive engine systems and industrial processes. All of these systems have a complex structure and require a close supervision; FTC utilizes plant redundancy to create an intelligent system that can supervise the behavior of the plant components making these kinds of systems more reliable.

2. ARTIFICIAL INTELLIGENCE METHODS

The use of AI in fault tolerant control has been suggested in the past (Bastani & Chen, 1988). Methods such as Neural Networks (NNs), Fuzzy Logic and Neuro-Fuzzy Systems, offer an advantage over traditional methods (state observers, statistical analysis, parameter estimation, parity relations, residual generation, etc) because can reproduce the behavior of non linear dynamical systems with models extracted from data. This is a very important issue in FTC applications on automated processes, where information is easily available, or processes where accurate mathematical models are hard to obtain. In the other hand, AI optimization tools such as Genetic Algorithms (GAs) provide a powerful tool for multi objective optimization problems frequently found on FTC.

2.1 Neural Networks

Artificial Neural Networks (ANNs) are mathematical models that try to mimic the biological nervous system. An artificial neuron have multiple input signals *x1*, *x2*, …,*xn* entering the neuron using connection links with specific weights w1, w2, …, wn or $\sum_{i=1}^{n} WnXi$ named the net input, and also have a firing threshold b, an activation function f and an output of the neuron that is represented by y=f($\sum_{i=1}^{n} WnXi - b$). The firing threshold b or bias can be represented as another weight by placing an extra input node x0 that takes a value of 1 and has a w0=-b. (Nguyen et al., 2002). This can be represented in the figure 1.

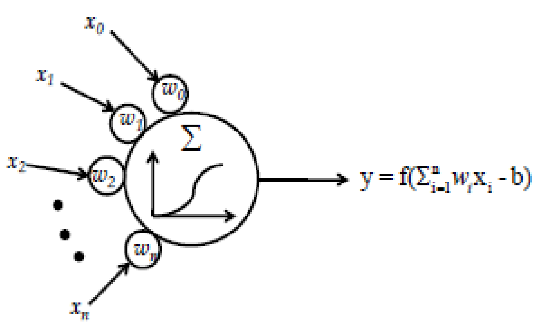

Fig.1 Artificial Neuron

A neural network with more than one input layer of neurons, a middle layer called the hidden layer and an output layer is named a multi-layer neural network.

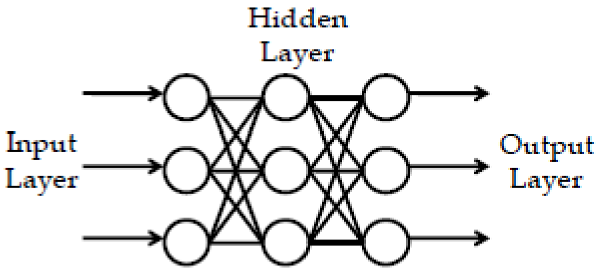

Fig.2 Multi-layer neural network

A neural network can have a feedback or a feed forward structure. In the feedback structure the information can move back and forward. In the feed forward structure, the information moves only forward from the input nodes through the outputs nodes with no cycles in the network (Ruan, 1997). The neural networks need to be trained from examples, in a process called supervised learning. Once a successfully training is done, the neural network is ready if and only if the networks reproduce the desired outputs from the given inputs. The most common methodology for this kind of learning is the back propagation algorithm, where the weights of the neural network are determined by using iteration until the output of the network is the same as the desired output (Rumelhart et al., 1986). In addition, unsupervised learning uses a mechanism for changing values of the weights according to the input values, this mechanism is named self-organization. An example of this algorithm is the Hebbian learning algorithm (Ruan, 1997).

2.1.1 Neural Networks in Fault Tolerant Control

Artificial neural networks have been applied in fault tolerant control because they are helpful to identify, detect and accommodate system faults. The application of ANNs to FTC can be divided in three groups. The first group includes neural networks used as fault detectors by estimating changes in process models dynamics (Polycarpou & Helmicki, 1995; Patton et al., 1999; Polycarpou, 2001; Gomaa, 2004). The second group includes neural networks used as controllers (Wang & Wang, 1999; Pashilkar et al., 2006), and the third group integrates neural networks which performs both functions: fault detection, and control (Perhinschi et al., 2007; Yen & DeLima 2005). (Polycarpou & Helmicki, 1995) proposed a construction of automated fault detection and accommodation architecture that uses on-line approximators and adaptive-learning schemes. The online approximator is a neural network model that monitors changes in the system dynamics due to a failure. (Patton et al., 1999) use a scheme of neural network to detect and isolate a fault in two steps: residual generation and decision making. In the first step a residual vector characterizes the fault and then the second step process the residual vector information in order to locate the fault and the time of occurrence. Once the residual is trained, qualitative knowledge of the plant can be added. This combination of qualitative and quantitative approached is helpful to decrease the number of false alarms in the fault decision making step. (Polycarpou, 2001) proposed a methodology for fault accommodation of a multivariable nonlinear dynamical system using a learning approach that monitors and approximates any abnormal behavior using neural networks and adaptive nonlinear estimation. When a fault occurs the neural network is used to estimate the nonlinear fault function supplying a framework for fault identification and accommodation. The neural network at the beginning of the monitoring stage is capable of learning the modeling errors in order to improve the system robustness. (Gomaa, 2004) recommended a fault tolerant control approach based on multi-ANN system faulty models. The nominal plant is nonlinear and is vulnerable to faults. A feedforward neural network is trained as the nominal model; two PID controllers are used, one for the nominal plant and the other for the neural network imitating the nominal plant (reference model). Both PIDs controllers were tuned using genetic algorithms. If there exist a difference between the nominal plant (yp) and the reference model (yrm) a nonzero residual is generated. Then, depending on the magnitude of the residual an ANN faulty model and its respective compensation path are selected to repair the fault and improve the system operating conditions. This can be observed in fig. 3.
This last neural network is a two layers perceptron network and its weights are updated using the modified gradient approach. This FTC system is shown in figure 4. (Pashilkar et al. 2006) proposed a neural controller that improves the fault tolerant potential of a fighter aircraft during landing. The faults are caused by severe winds or stuck control surfaces and can be divided in single faults (aileron or elevator stuck) or double fault (aileron and elevator stuck). This neural network controller employs a feedback error learning method with a dynamic radial basis function neural network.

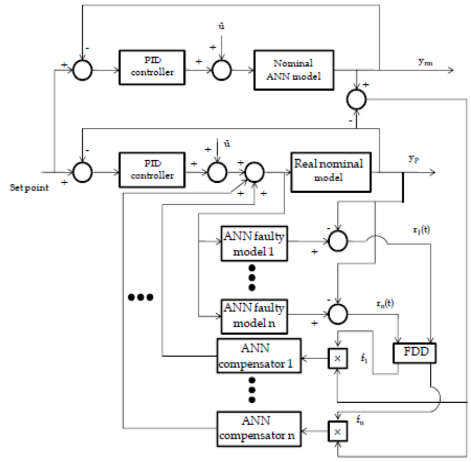

Fig.3. Multi-ANN faulty models FTC scheme (Gomaa, 2004).

The neural network uses on-line training and not a-priori training. This kind of controller helped to improve the capability of handling large faults and also helps to achieve the desired requirements. (Perhinschi et al., 2007) presented a methodology for detection, identification and accommodation of sensor and actuator failures inside fault tolerant control laws. The fault detection and identification uses neural estimators. The accommodating control laws design for the actuator fault is done using nonlinear dynamic inversion with neural network augmentation. Whereas the accommodation of sensor fault is accomplished by changing the failed sensor output for neural estimates calculated in the detection and identification process. This approach can handle sensor and actuator faults successfully. It uses membership functions to describe the mathematical model of process. (Yen & DeLima, 2005) presented a neural network trained on-line with a global dual heuristic programming architecture. This approach has also a supervision structure made from decision logic. This supervision level is very efficient to identify the controller faults in early stages and can supply new values to improve the convergence utilizing dynamic model bank information.

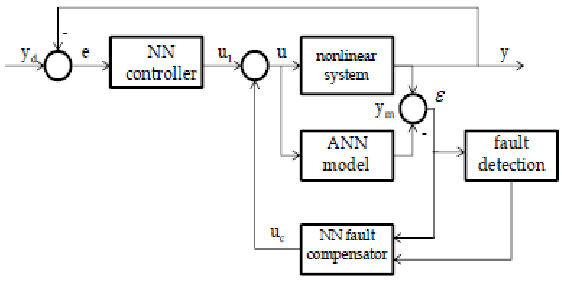

Fig.4. Neural Network FTC scheme proposed by (Wang & Wang, 1999)

2.2 Genetic Algorithms

Genetic Algorithms (GAs) are searching and optimizing algorithms motivated by natural selection evolution and natural genetics (Goldberg, 1989). The simplest GA follows the next steps: Generate a random initial population of chromosomes, calculate the fitness of every chromosome in the population, apply selection, crossover and mutation and replace the actual population with the new population until the required solution is achieved. The main advantages of GAs are: powerful computational effect, robustness, fault tolerance, fast convergence to a global optimal, capability of searching in complex landscape where the fitness function is discontinuous, can be combined with traditional optimization techniques (Tabu search) and have the ability to solve problem without needing human experts (Goldberg, 1989; Mitchell, 1996; Ruan, 1997).

2.2.1 Genetic Algorithms in Fault Tolerant Control

Recently genetic algorithms have been applied in fault tolerant control as a strategy to optimize and supervise the controlled system in order to accommodate system failures. Some applications of this technique are the following: (Schroder et al., 1998) proposed a fault tolerant control technique for an active magnetic bearing. In this approach a nonlinear model of a turbo machine rotor from the rolls-royce lifted up by an active magnetic bearing was presented. This model is capable of modeling difference configuration of magnetic bearings. A multi-objective genetic algorithm was used to generate and adequate PID controller for the active magnetic bearing with different bearing configuration. Also the fault accommodation was done using a centralized fault compensation scheme. (Sugawara et al., 2003) showed a fault tolerant control approach using multi-layer neural networks with a genetic algorithm. The proposed of this approach was to develop a self recovery ship to accommodate faults without the needing of a host computer. This FTC scheme uses hardware redundancy and weight retraining using a genetic algorithm in order to reconfigure the neural network to accommodate the fault. The objective of the genetic algorithm is to reduce the error between the actual output and the desired output.

2.2.2 The Proposed Methodology, Experiments and Results

We propose a new FTC schema, where a Model Reference Adaptive Control (MRAC) is used in combination with a neural network controller, in order to achieve a better performance when faults are present in the system. We use an experimental model of a heat exchanger where abrupt and gradual faults (also called soft faults) are induced in sensors and actuators. To compare our schema, we also have made experiments with a simple MRAC and MRAC-PID structures.

3. MRAC CONTROLLER

The Model Reference Adaptive Controller, shown in fig.5, implements a closed loop controller that involves the parameters that should be optimized, in order to modify the system response to achieve the desired final value. The adaptation mechanism adjusts the controller parameters to match the process output with the reference model output.

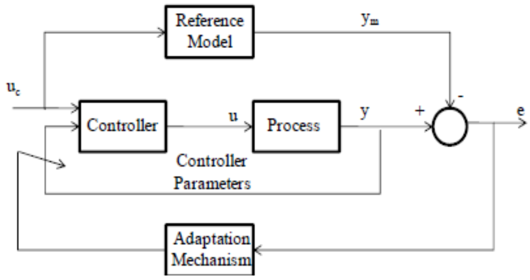

Fig.5 MRAC scheme (Nagrath, 2006)

The controller error is calculated as follows:

$$e = y_{process} - y_{reference} \quad (1)$$

To reduce the error, a cost function was used, in the form of:

$$J(\theta) = 1/2\, e^2(\theta) \quad (2)$$

The function above can be minimized if the parameters **θ** change in the negative direction of the gradient **J**, this is called the gradient descent method and is represented by:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (3)$$

where γ helps to adjust the speed of learning. The above equation is known as the MIT rule and determines how the parameter θ will be updated in order to reduce the error.

The implemented MRAC scheme in our process, shown in fig. 6, has two adaptation parameters: adaptive feed forward gain (θ_1) and adaptive feedback gain (θ_2) . These parameters will be updated to follow the reference model.

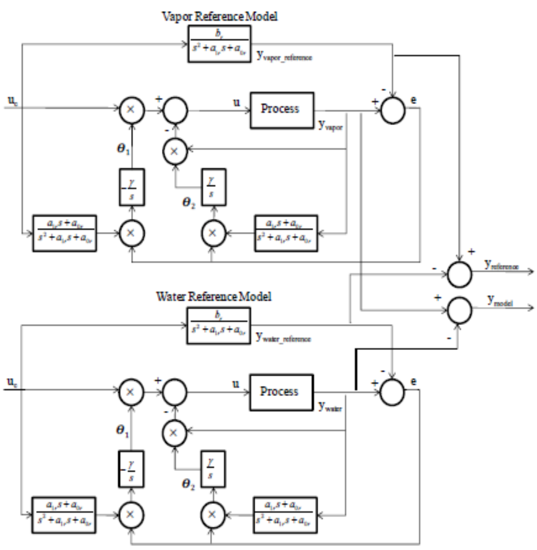

Fig. 6 Fault tolerant MRAC scheme.

3.1 Experiments

Two different faults were simulated: abrupt faults and gradual faults. In the abrupt faults case, the whole magnitude of the fault is developed in one moment of time and was simulated with a step function. On the other hand, gradual faults are developed during a period of time and are implemented with a ramp function. Both types of faults, abrupt and gradual, can be implemented in sensors (feedback), in which the properties of the process are not affected, but the sensor readings are mistaken. Also, can be implemented in actuators (process entry) in which the process properties are not affected either, but the process behavior can change or can be interrupted. We use an industrial heat exchanger to test our approach (shown in fig. 7). The process has two inputs: water and steam flows controlled by pneumatic valves, and one output, the water temperature measured by a thermocouple. Variations in water and steam flows are determined by flow transmitters. To develop the continuous model of this process, an identification experiment was performed, where a Pseudo Random Binary Sequence (PRBS) was applied to water and steam valves, and variations in water temperature were recorded. With the data obtained in the PRBS test, the identification was achieved using software developed in Matlab.

Fig. 7 Industrial heat exchanger used in the experiments

The following model was obtained:

$$G_p = G_{vapor} - G_{water}.$$

$$G_p = \frac{0.00002}{s^2 + 0.004299s + 0.00002} - \frac{0.000013}{s^2 + 0.007815s + 0.00008} \quad (4)$$

A total of six different experiments were developed in Simulink. Table 1 explains the results of each simulated experiment and shows the numerical performance of every method by using the Mean Square Error (MSE) obtained during the application of the fault.

Table 1. Results of experiments with abrupt and gradual faults simulated in the 3 different fault tolerant MRAC schemes.

Method & Fault Type	Results when a the Fault was applied In Sensor	Results when an Fault was applied In Actuator
MRAC Abrupt Fault	- If the fault magnitude is < 0.44, the system is robust against the fault. - If the fault magnitude is between [0.44,1.52] the system accommodates the fault. -If the fault magnitude is > 1.52, the system becomes unstable. - **MSE= 0.501236189**	- If the fault magnitude is 1 the system response varies around +/- 3%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.10521016**
MRAC Gradual Fault	- If the fault has saturation < +/- 0.44, the system is robust against the fault. - If the fault has a saturation between +/- [0.44, 1.52] the system accommodate the fault. -If the fault has saturation > 1.52, the system becomes unstable. **MSE=0.50777113**	- If the fault saturation is +/- 1 the system response varies around +/- 4%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. - **MSE=0.09163081**

The following graphs represent a comparison between the different simulated experiments. Fig. 8 represents system behavior when abrupt faults are simulated. The three graphs on the left column are sensor faults and the graphs from the right column are actuator faults. The sensor faults have a magnitude of 1.8 and the actuator faults a magnitude of 1. It is observed that the MRAC-Neural Network represents the best scheme because is insensitive to abrupt sensor faults and has a good performance when abrupt actuator faults are developed. Fig. 9 graphs represent system behavior when gradual faults are present on the system. The fault magnitude of the sensor fault is of 1.8 and the magnitude of the actuator fault is of 1. It can be seen also that the MRAC-Neural Networks Controller scheme is the better option because is robust to sensor faults and has a less degraded performance in actuator faults. In conclusion, the proposed MRAC-Neural Network scheme gives the best fault tolerant control scheme developed in this work.

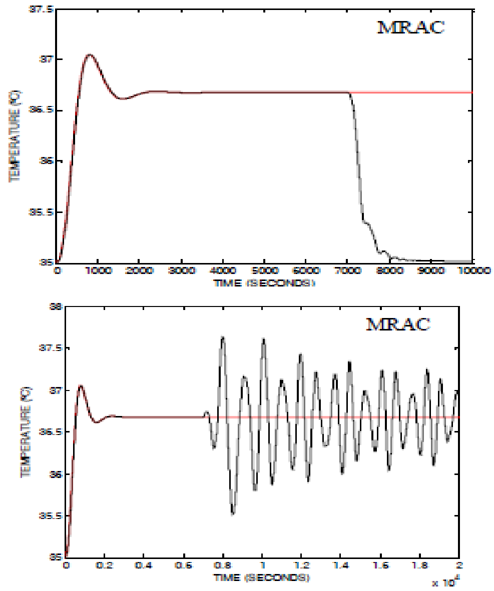

Fig. 8 Abrupt-Sensor Faults (left column) and Abrupt-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

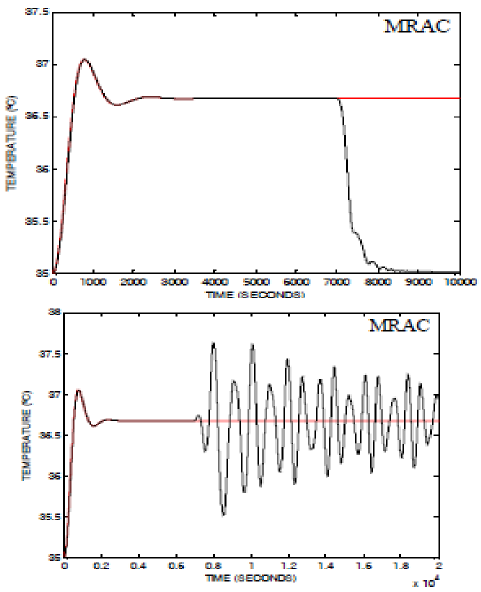

Fig. 9 Gradual-Sensor Faults (left column) and Gradual-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

4. REFERENCES

[1] Ballé, P.; Fischera, M.; Fussel, D.; Nells, O. & Isermann, R. (1998). Integrated control,diagnosis and reconfiguration of a heat exchanger. IEEE Control Systems Magazine,Vol. 18, No. 3, (June 1998) 52–63, ISSN: 0272-1708.

[2] Bastani, F., & Chen, I. (1988). The role of artificial intelligence in fault-tolerant processcontrol systems. Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems, pp. 1049-1058, ISBN:0-89791-271-3, June 1988, ACM, Tullahoma, Tennessee, United States.

[3] Blanke, M.; Izadi-Zamanabadi, R.; Bogh, R. & Lunau, Z. P. (1997). Fault tolerant control systems—A holistic view. Control Engineering Practice, Vol. 5, No. 5, (May 1997) 693–702, ISSN: S0967-0661(97)00051-8.

[4] Blanke, M., Staroswiecki, M., & Wu, N. E. (2001). Concepts and methods in fault-tolerant control. In Proceedings of the 2001 American Control Conference, pp. 2606–2620, Arlington, Virginia, ISBN: 0-7803-6495-3, June 2001, IEEE, United States.

[5] Caglayan, A.; Allen, S. & Wehmuller, K. (1988). Evaluation of a second generation reconfiguration strategy for aircraft flight control systems subjected to actuator failure/surface damage. Proceedings of the 1988 National Aerospace and Electronics Conference, pp. 520–529, May 1988, IEEE, Dayton , Ohio, United States.

[6] Diao,Y. & Passino, K. (2002). Intelligent fault-tolerant control using adaptive and learning methods. Control Engineering Practice, Vol. 10, N. 8, (August 2002) 801–817, ISSN: 0967-0661.

[7] Gao, Z. & Antsaklis, P. (1991). Stability of the pseudo-inverse method for reconfigurable control systems. International Journal of Control, Vol. 53, No. 3, (March 1991) 717–729. Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning, Addison-Wesley, ISBN: 0201157675, Reading, Massachusetts, United States.

[8] Gomaa, M. (2004). Fault tolerant control scheme based on multi-ann faulty models. Electrical, Electronic and Computer Engineering. ICEEC International Conference, Vol. , No. , (September 2004) 329 – 332, ISBN: 0-7803-8575-6.

[9] Isermann, R.; Schwarz, R. & Stölzl, S. (2002). Fault-tolerant drive-by-wire systems. IEEE Control Systems Magazine, Vol. 22, No. 5, (October 2002) 64-81, ISSN: 0272-1708.

[10] Jaimoukha, I.; Li, Z. & Papakos, V. (2006). A matrix factorization solution to the H-/H infinity fault detection problem. Automatica, Vol. 42, No. 11, 1907 – 1912, ISSN: 000-1098.

[11] Jiang, J. (1994). Design of reconfigurable control systems using eigenstructure assignments. International Journal of Control, Vol. 59, No. 2, 395–410, ISNN 00-7179.