View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by IntechOpen

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

15

Artificial Intelligence Methods in Fault Tolerant Control

Luis E. Garza Castañón and Adriana Vargas Martínez
Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM)
Monterrey, México

1. Introduction to Fault Tolerant Control

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.
- Prevention to develop a single fault in to a system failure.
- The use of information redundancy to detect faults instead of adding more hardware.
- The use of reconfiguration in the system components to accommodate a fault.
- FTC admits degraded performance due to a fault but maintain the system availability.
- 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.

Since a few years ago, emerging FTC techniques have been proposing new controller designs capable to tolerate system malfunctions and maintain stability and desirable performance properties. In order to achieve its objectives, two main tasks have to be considered on an active FTC system: fault detection and diagnosis and controller reconfiguration. The main purpose of fault detection and diagnosis is to detect, isolate and identify the fault, determining which faults affect the availability and safety of the plant. The controller reconfiguration task accommodates the fault and re-calculates the controller parameters in order to reduce the fault effects.
Although several schemes of FTCS have been proposed, most of them are closely related to a general architecture. (Blanke et al., 1997) introduce an approach for the design of an FTC system, shown in figure 1, which included three operational levels: single sensor validation, fault detection and isolation using analytical redundancy, and an autonomous supervision and reconfiguration system. The single sensor validation level involves the control loop with actuators, sensors, the controller and the signal conditioning and filtering. The second level (FDI) is composed of detectors and effectors that will perform the remedial actions. And finally, the supervision level deals with state-event logic in order to describe the logical state of controlled objects.

Fig. 1. Architecture for Fault Tolerant Autonomous Control Systems proposed by (Blanke, 1997).

A slightly different architecture is presented in (Karsai et al, 2003). They introduce a scheme of Fault-Adaptive Control Technology (FACT), centered on model-based approaches for fault detection, fault isolation and estimation, and controller selection and reconfiguration for hybrid systems (see figure 2). Hybrid models derived from hybrid bond graphs are used to model the continuous and discrete system dynamics. The supervisory controller, modeled as a generalized finite state automaton, generates the discrete events that cause reconfigurations in the continuous energy-based bond graph models of the plant. Fault detection involves a comparison between expected behaviors of the system, generated from the hybrid models, with actual system behavior.

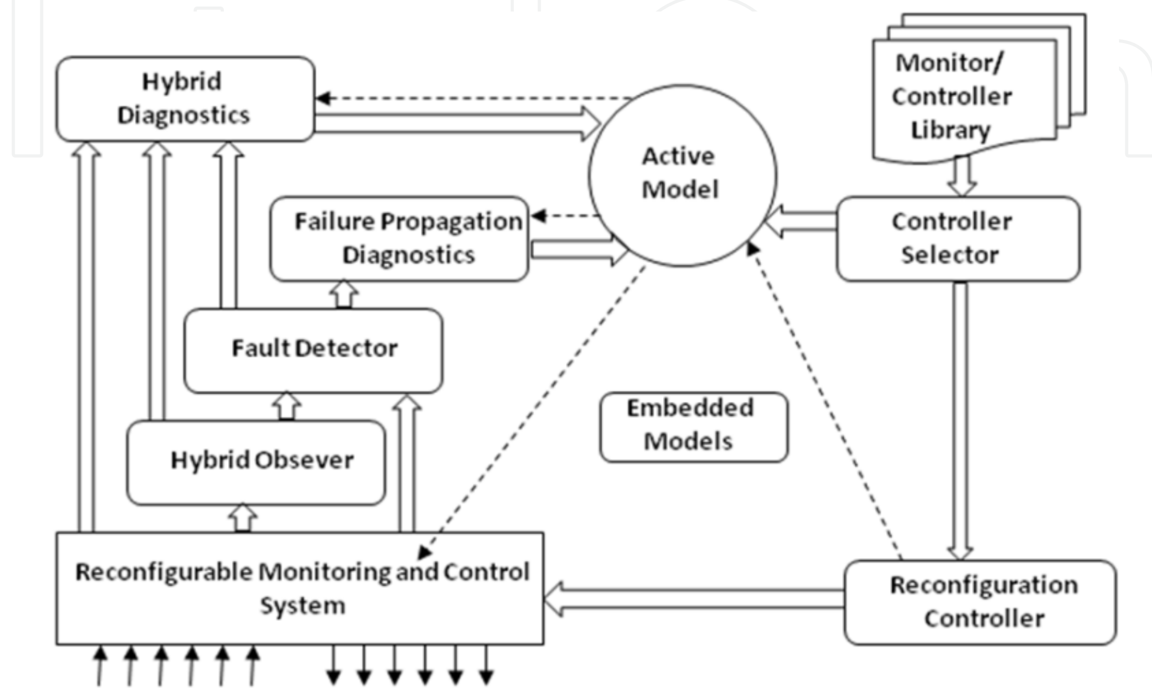

Fig. 2. Architecture for Fault-Adaptive Tolerant Control Technology (FACT) proposed by (Karsai et al, 2003).

2. Classification of the Fault Tolerant Control Methods

Some authors have proposed different classifications for the FTC methods (Blanke et al., 2003; Eterno et al., 1985; Farrel et al., 1993; Lunze & Richter, 2006; Patton, 1997; Stengel, 1991). The classification shown in figure 3 includes all the methods explained by these authors. We can also find a recent and very complete survey of FTC methods and applications in (Zhang & Jiang, 2008).

Regarding the design methods, fault tolerant control can be classified into two main approaches: active or passive. In Active Fault Tolerant Control (AFTC), if a fault occurs, the control system will be reconfigured using some properties of the original system in order to maintain an acceptable performance, stability and robustness. In some cases degraded system operations have to be accepted (Blanke et al., 2001; Patton, 1997; Mahmoud et al., 2003). In Passive Fault Tolerant Control (PFTC) the system has a specific fixed controller to counteract the effect and to be robust against certain faults (Eterno et al., 1985).

To implement the AFTC approach two tasks are needed: fault detection and isolation and controller reconfiguration or accommodation. FDI means early detection, diagnosis, isolation, identification, classification and explanation of single and multiple faults; and can be accomplished by using the following three methodologies (Venkatasubramanian et al., 2003a, 2003b, 2003c):

Quantitative Model-Based: requires knowledge of the process model and dynamics in a mathematical structural form. Also, the process parameters, which are unknown, are calculated applying parameter estimation methods to measured inputs and outputs signals of the process. This approach uses analytical redundancy that can be obtained by implementing Kalman filters, observers and parity space.

Qualitative Model-Based: Are based on the essential comprehension of the process physics and chemical properties. The model understanding is represented with quality functions placed in different parts of the process. This methodology can be divided in abstraction hierarchies and causal models. Abstraction hierarchies are based on decomposition and the model can establish inferences of the overall system behavior from the subsystems law behavior. This can be done using functional or structural approaches. Causal models take the causal system structure to represent the process relationships and are classified in diagraphs, fault trees and qualitative physics.

Process History-Based: uses a considerable amount of the process historical data and transform this data into a priori knowledge in order to understand the system dynamics. This data transformation is done using qualitative or quantitative methods. The quantitative methods are divided in expert systems (solves problems using expertise domain) and trend modeling (represents only significant events to understand the process). Quantitative methods can be statistical (use PCA, DPCA, PLA, CA) and non statistical (neural networks) to recognize and classify the problem.

After the detection and isolation of the fault, a controller reconfiguration or accommodation is needed. In controller accommodation, when a fault appears, the variables that are measured and manipulated by the controller continue unaffected, but the dynamic structure and parameters of the controller change (Blanke et al., 2003). The fault will be accommodated only if the control objective with a control law that involves the parameters and structure of the faulty system has a solution (Blanke et al., 2001). In order to achieve fault accommodation, two approaches can be used: adaptive control and switched control. Adaptive control means to modify the controller control law to handle the situation where the system's parameters are changing over time. It does not need a priori information about the parameters limits. The goal is to minimize the error between the actual behavior of the system and the desirable behavior. In the other hand, switched control is determined by a bank of controllers designed for specifics purposes (normal operation or fault) that switch from one to another in order to control a specific situation (Lunze & Richter, 2006).

Meanwhile, controller reconfiguration is related with changing the structure of the controller, the manipulated and the measured variables when a fault occurs (Steffen, 2005). This is achieved by using the following techniques:

Controller Redesign. The controller changes when a fault occurs in order to continue achieving its objective (Blanke et al., 2003). This can be done by using several approaches: pseudo inverse methods (modified pseudo inverse method, admissible pseudo inverse method), model following (adaptive model following, perfect model following, eigen structure assignment) and optimization (linear quadratic design, model predictive control) (Caglayan *et al.*, 1988; Gao & Antsaklis, 1991; Jiang, 1994; Lunze & Richter, 2006; Staroswiecki, 2005).

Fault Hiding Methods. The controller continues unchanged when a fault is placed, because a reconfiguration system hides the fault from the controller. This method can be realized using virtual actuators or virtual sensors. (Lunze & Richter, 2006; Steffen, 2005).

Projection Based Methods. A controller is designed a priori for every specific fault situation and replaces the nominal controller if that specific fault occurs. This can be done by a bank of controllers and a bank of observers (Mahmoud et al., 2003).

Learning Control. This methodology uses artificial intelligence like neural networks, fuzzy logic, genetic algorithms, expert systems and hybrid systems which can learn to detect, identify and accommodate the fault (Polycarpou & Vemuri, 1995; Stengel, 1991; Karsai et al, 2003).

Physical Redundancy. This is an expensive approach because it uses hardware redundancy (multiple sensor or actuators) and decision logic to correct a fault because it switches the faulty component to a new one. An example of this is the voting scheme method (Isermann et al., 2002; Mahmoud et al., 2003).

On the other hand, passive FTC is based on robust control. In this technique, an established controller with constant parameters is designed to correct a specific fault to guarantee stability and performance (Lunze & Richter, 2006). There is no need for online fault information. The control objectives of robust control are: stability, tracking, disturbance rejection, sensor noise rejection, rejection of actuator saturation and robustness (Skogestad & Postlethwaite, 2005). Robust control involves the following methodologies:

H_∞ controller. This type of controller deals with the minimization of the H-infinity-norm in order to optimize the worst case of performance specifications. In Fault Tolerant Control can be used as an index to represent the attenuation of the disturbances performances in a closed loop system (Yang & Ye, 2006) or can be used for the design of robust and stable dynamical compensators (Jaimoukha et al., 2006; Liang & Duan, 2004).

Linear Matrix Inequalities (LMIs). In this case, convex optimization problems are solved with precise matrices constraints. In Fault Tolerant Control is implemented to achieve robustness against actuator and sensor faults. (Zhang et al., 2007).

Simultaneous Stabilization. In this approach multiple plants must achieve stability using the same controller in the presence of faults. (Blondel, 1994).

Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization. This methodology is implemented in Fault Tolerant Control to parameterize stabilizing controllers in order to guarantee system stability. YJBK in summary is a representation of the feedback controllers that stabilize a given system (Neimann & Stoustrup, 2005).

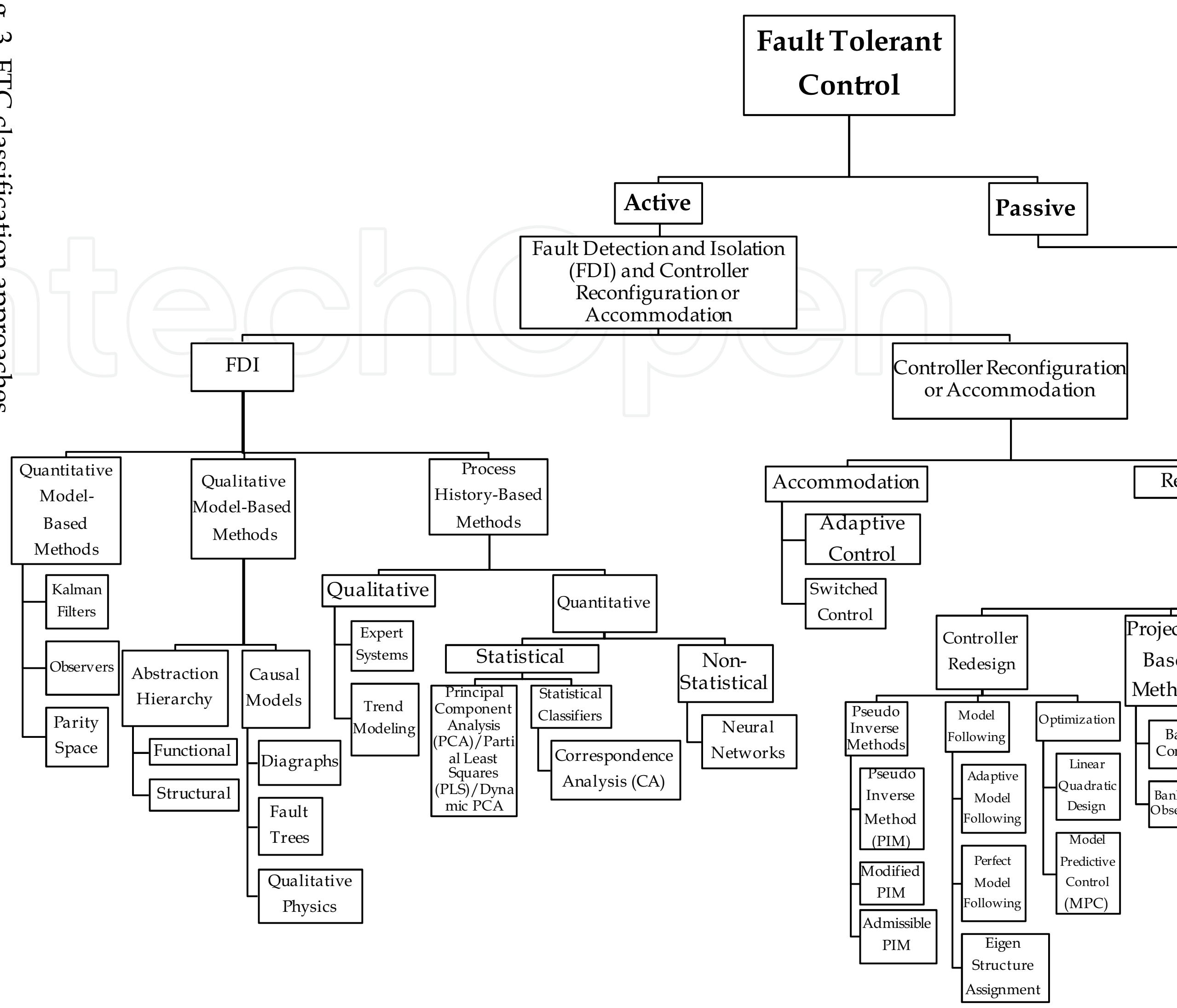

Fig. 3. FTC classification approaches

IntechOpen

3. 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 multiobjective optimization problems frequently found on FTC.

3.1 Neural Networks

Artificial Neural Networks (ANNs) are mathematical models that try to mimic the biological nervous system. An artificial neuron have multiple input signals $x_1, x_2, \ldots, x_n$ entering the neuron using connection links with specific weights $w_1, w_2, \ldots, w_n$ or $\sum_{i=1}^{n} w_n x_i$ 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 $y = f\left(\sum_{i=1}^{n} w_i x_i - b\right)$. The firing threshold b or bias can be represented as another weight by placing an extra input node x_0 that takes a value of 1 and has a w_0=-b. (Nguyen et al., 2002). This can be represented in the figure 4.

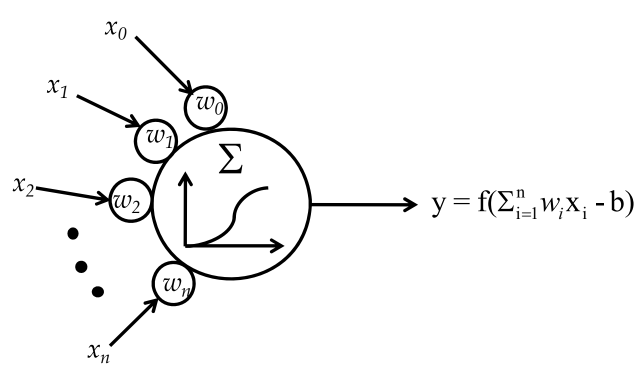

Fig. 4. Artificial Neuron.

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.

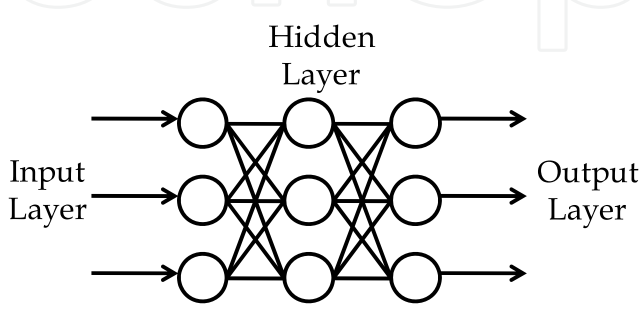

Fig. 5. Multi-layer neural network.

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 feedforward 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 backpropagation 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).

3.1.1 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 (y_p) and the reference model (y_{rm}) 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 figure 6.

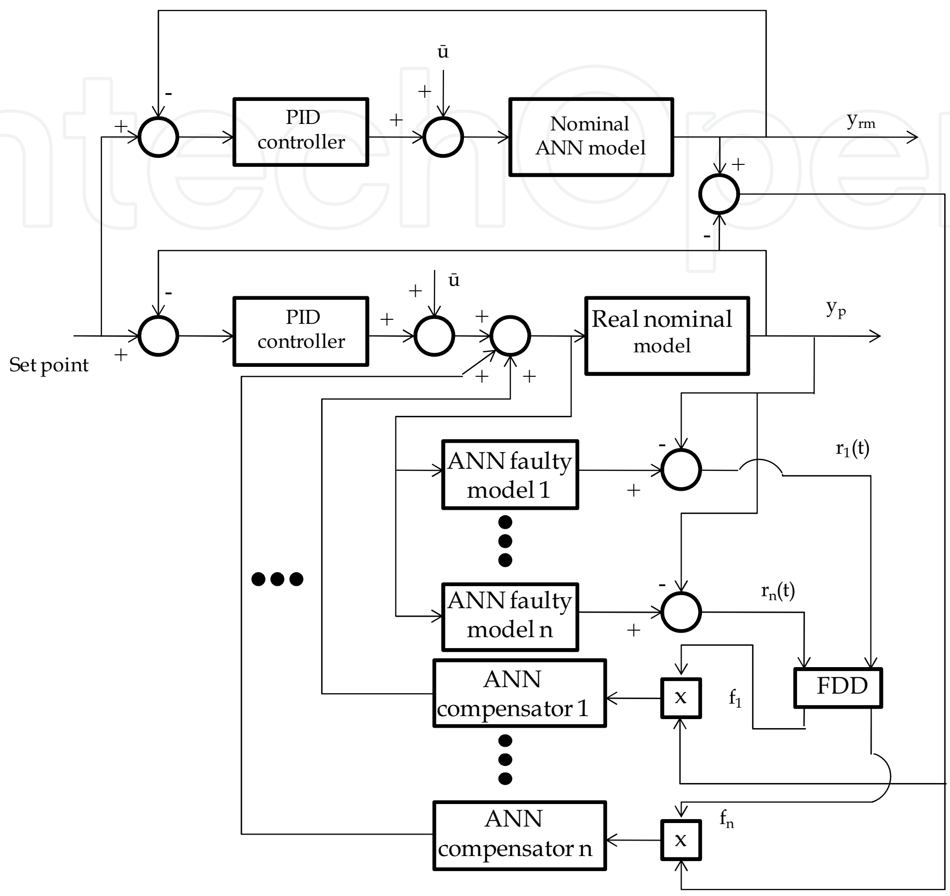

Fig. 6. Multi-ANN faulty models FTC scheme (Gomaa, 2004).

(Wang & Wang, 1999) proposed a Neural Network-FTC where the nominal system is controlled by a Pseudolinear Neural Network (PNN) based one-step-ahead controller that uses a modified gradient approach. When the system is with no fault, a PNN model connected in parallel to the plant model can be trained and used for the design of the control algorithm. The PNN model is helpful for the detection of a fault. In addition, when a fault is present, a residual signal is generated and an extra neural network based fault compensation loop is imported in order to provide the closed loop stability. 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 7.
(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. 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.

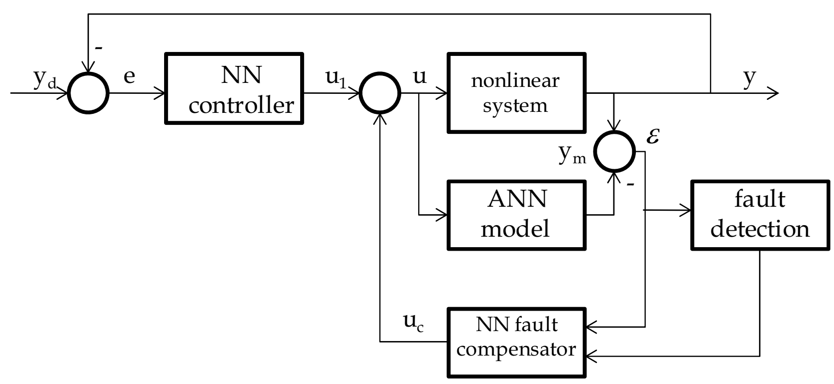

Fig. 7. Neural Network FTC scheme proposed by (Wang & Wang, 1999).

3.2 Fuzzy Logic

Fuzzy logic is a research field of fuzzy set theory; this methodology uses heuristic and mathematical tools to approximate reasoning in order to achieve knowledge processing when lack of information exists. In addition, this approach tries to mimic the control logic rules used by persons.
In Fuzzy systems, a mathematical model of the system is not needed because the desired input and output relationships are defined by “if” and “then” rules. Also, due to the lack of system information it is not possible to precise the exact value of a variable “x” but it is possible to know the membership function. The membership function is created to describe range of values that the variable can have, this is known as input fuzzification (Passino & Yurkovich, 1997; Ruan, 1997; Nguyen, 2002). This membership function can have different shapes that are defined according to a specific problem; the shapes forms are: triangular, Gaussian trapezoidal, sigmoidal S and sigmoidal Z functions.
Once the membership functions of variables are established, the inference rules “if (antecedent)-then (consequent)” are set. Then an inference method is used to quantify each premise, this is an application of a fuzzy operator in the “if” part of the rule and can be the math function “or” – “and”, once this is done, the inference method fire rules, which is known as the implication method. Two inference methods are commonly used: Mamdani method, where the “if” and “then” parts are fuzzy expressions, and the Takagi-Sugeno

method where the "if" part of the rule is a fuzzy expression, but the "then" part of the rule is a deterministic expression (Passino & Yurkovich, 1997; Nguyen, 2002). After applying the chosen inference method, the output fuzzy set is form. This can be shown in the next rules:

If ***(0.4 or 0.2)*** **then** ***input (0.4)***

If ***max (0.4, 0.2)*** **then** ***input (0.4)***

Finally***:*** **If** ***(0.4)*** **then** ***input (0.4)***

The above means that if the antecedent part of the rule is partially true which means that have a value minor than one, then the output fuzzy set will be truncated by the implication method, as showed in the next figure:

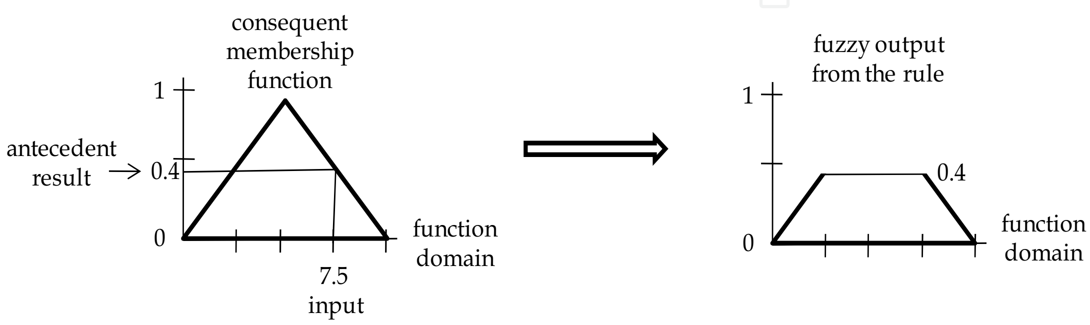

Fig. 8. Implication method representation.

The last step in the inference method is the aggregation of all the fuzzy outputs from the rules, as can be seen in the next figure:

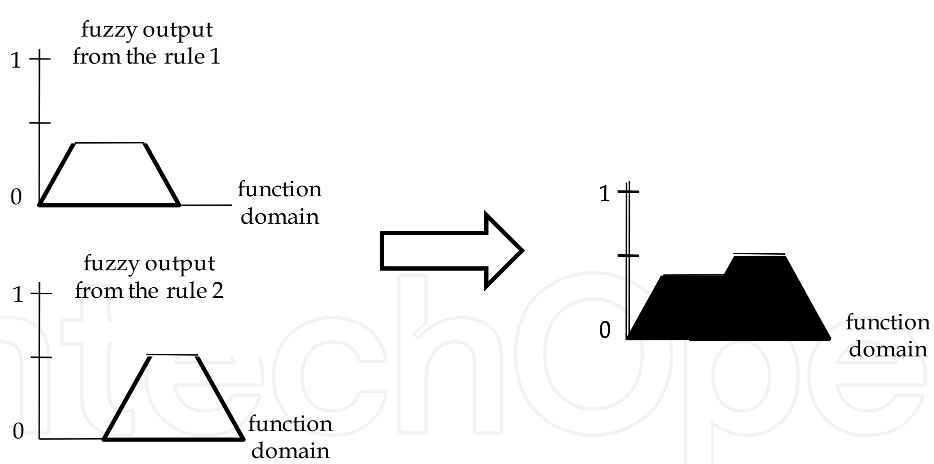

Fig. 9. Fuzzy outputs aggregation representation.

Finally, the fuzzy controller implements defuzzification, where the input of the defuzzification process is a fuzzy set and the output is just a number. This can be done using several methods as: center of area method also known as the center or gravity or centroid method, the height-center of area, the max criterion, the first of maxima and the middle of maxima method. The deffuzification result will be the output of the fuzzy system (Passino & Yurkovich, 1997; Nguyen, 2002). This is showed in the following figure:

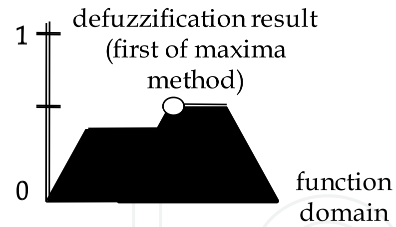

Fig. 10. Defuzzification representation.

In summary, the main advantages of Fuzzy Logic are: robustness, easy controller reconfiguration, works well with multi input multi output systems and adequate to model complex nonlinear systems.

3.2.1 Fuzzy Logic in Fault Tolerant Control

Fuzzy models are capable of handling nonlinear dynamic systems and the nonlinearities of the faults. In the following we describe several works reported with fault detectors and/or controllers based on fuzzy logic.

(Kwong et al., 1995) presented an expert supervision approach of Fuzzy learning system for fault tolerant control in aircrafts. This approach, shown in figure 11, uses a fuzzy model reference learning controller (FMRLC) to reconfigure the aircraft nominal controller in order to accommodate actuator failures without needing specific information about the failures. This approach also demonstrated that the FMRLC can be improved using fault detection and identification information to acquire an adaptive system that can reconfigure its performance level when a failure occurs.

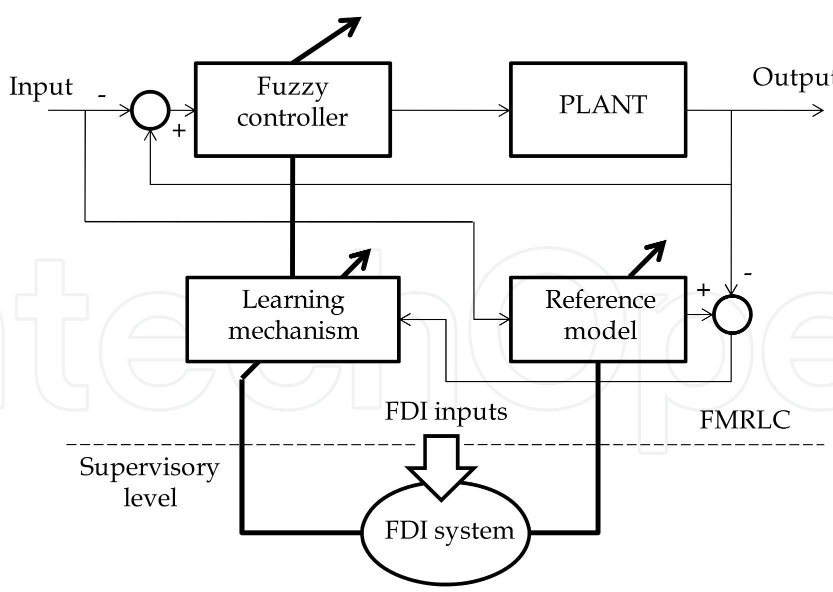

Fig. 11. Supervisory FMRLC scheme proposed by (Kwong et al., 1995).

(Ballé et al., 1998) presented a model-based adaptive control and reconfiguration approach based on fault detection and diagnosis implemented to a heat exchanger. The fault detection and adaptive tasks are based on Takagi-Sugeno (T-S) fuzzy model of the process and the

fault diagnosis task uses a self-organizing fuzzy structure. In this approach an on-line adaptation scheme is used for reconfiguration when the measurements are erroneous because of the sensor faults. If all the input signal and the disturbances can be measured the model is used for model-based predictive control. When a sensor fault occurs, the signals are not longer used for prediction. Instead of these measurements, a reduced model of the system process is required to adapt these erroneous signals as external disturbances. The T-S fuzzy model is also used for residuals generation in order to detect and isolate sensor faults. Then the decision making is sustained by identifying particular patterns of the residual that are link to different faults. This is done by using a fuzzy classification tree that can learn sample data patterns. The next figure shows the explained approach:

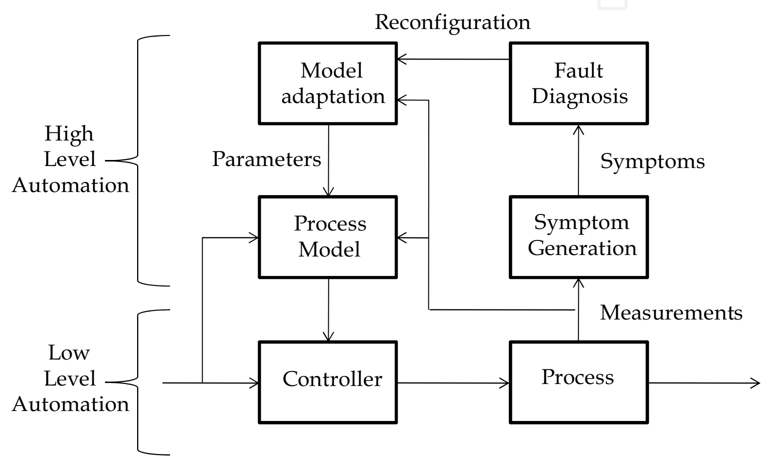

Fig. 12. Model-based control, diagnosis and reconfiguration scheme proposed by (Ballé al., 1998).

(Patton et al., 1999) proposed the use of Takagi-Sugeno fuzzy observers for nonlinear dynamic systems. In this methodology the plant is represented by a T-S fuzzy model and a fuzzy observer estimate the system state vector. The fuzzy model is locally observable in order to design the fuzzy observer. A linear time invariant observer is created with every real of the fuzzy model. Then, the whole complete observer's dynamic is a weighted sum of the individual observers. A residual is generated from the observers and is compared against the fuzzy model output. If the residual is nonzero a fault is present in the system.
(Holmes & Ray, 2001) proposed a damage mitigating control system in which the main objective is to accomplish optimal performance of the system with a high level of reliability, availability, component durability and maintainability. This scheme was implemented in a two-tier structure. A lineal robust sample data controller was proposed for the lower tier to track a reference trajectory vector. On the other hand, for the upper tier, a fuzzy logic based damage controller was implemented. This controller adjusts the system dynamic performance with the structural durability of critical components. This scheme demonstrated that a substantial gain in structural durability of critical components can be reached without significant performance degradation.
(Diao & Passino, 2001) implemented an intelligent Fault Tolerant Control scheme using a hierarchical learning structure for the design of the plant. This learning structure was a Takagi-Sugeno Fuzzy system. Then, the fault tolerant control system is based on stable

adaptive fuzzy/neural control that has online learning properties which are used to acquire the unknown dynamics developed as consequence of system faults.
(Diao & Passino, 2002) presented a fault tolerant control scheme using adaptive estimation and control methods based on the learning properties of fuzzy systems and neural networks. An online approximation-based stable adaptive neural/fuzzy control was implemented in an input-output feedback time-varying nonlinear system. Also, the adaptive controller enhanced the fault tolerance capability by using the estimated information from the fault diagnosis unit, chosen by a multiple model interface with a supervisory expert approach.
(Oudghiri et al., 2008) proposed a Fault Tolerant scheme for lateral vehicle dynamics implementing static output feedback control with sliding mode observers in order to enhance the vehicle stability and handling in the presence of sensor fault. This approach uses three blocks, one for the fault and detection, a second for the static output feedback controller and a third for a switcher. The lateral vehicle dynamics are represented in a nonlinear two degrees of freedom vehicle model implemented in a Takagi-Sugeno fuzzy model. The fault detection and isolation block uses a bank of observers' constructed using sliding mode in order to calculate the system state vector. Then, a set of diagnostic signal residuals are produced, associating the measured and the estimated outputs. After doing this, the sensor in which the fault happened is isolated.

3.3 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).

3.3.1 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 technique implemented for large scale neural networks programmed in a single 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.

4. 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.

4.1 MRAC Controller

The Model Reference Adaptive Controller, shown in figure 13, 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.

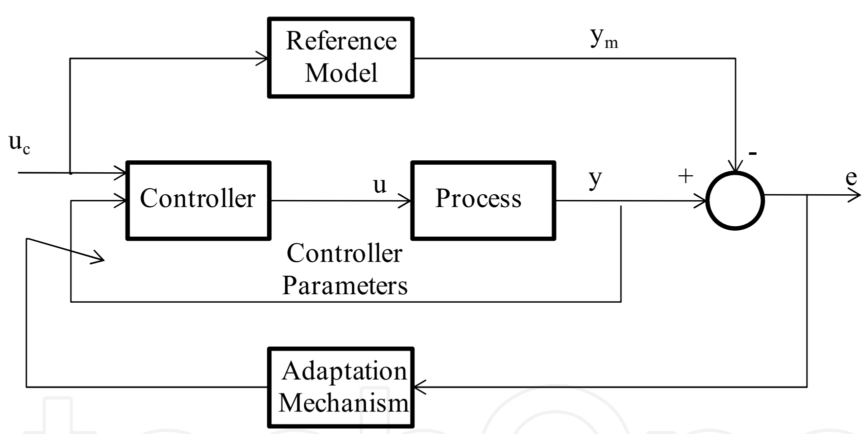

Fig. 13. MRAC scheme (Nagrath, 2006).

The controller error is calculated as follows:

$$e = y_{process} - y_{reference} \tag{1}$$

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

$$J(\theta) = \tfrac{1}{2} e^2(\theta) \tag{2}$$

The function above can be minimized if the parameters $\boldsymbol{\theta}$ change in the negative direction of the gradient $\boldsymbol{J}$, this is called the gradient descent method and is represented by:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \tag{3}$$

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 figure 14, has two adaptation parameters: adaptive feedfoward gain (θ_1) and adaptive feedback gain (θ_2). These parameters will be updated to follow the reference model.

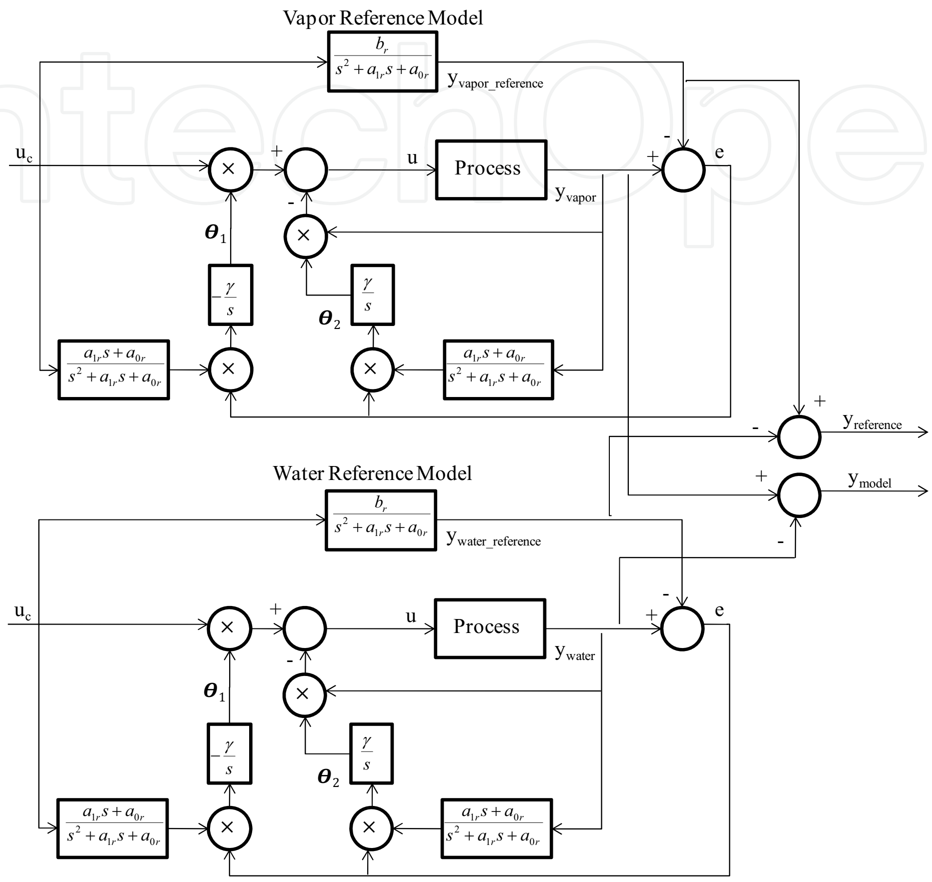

Fig. 14. Fault tolerant MRAC scheme.

4.2 PID Controller

To overcome the limitations of the simple MRAC structure, a classical PID controller, in the form of equation 4, was introduced in the feedforward part of simple MRAC scheme. The resulting structure is shown in figure 15. The desired response is introduced in terms of rise time, stabilization time and overshoot.

$$G_{PID\_controller} = K_p + K_i/s + K_d s \tag{4}$$

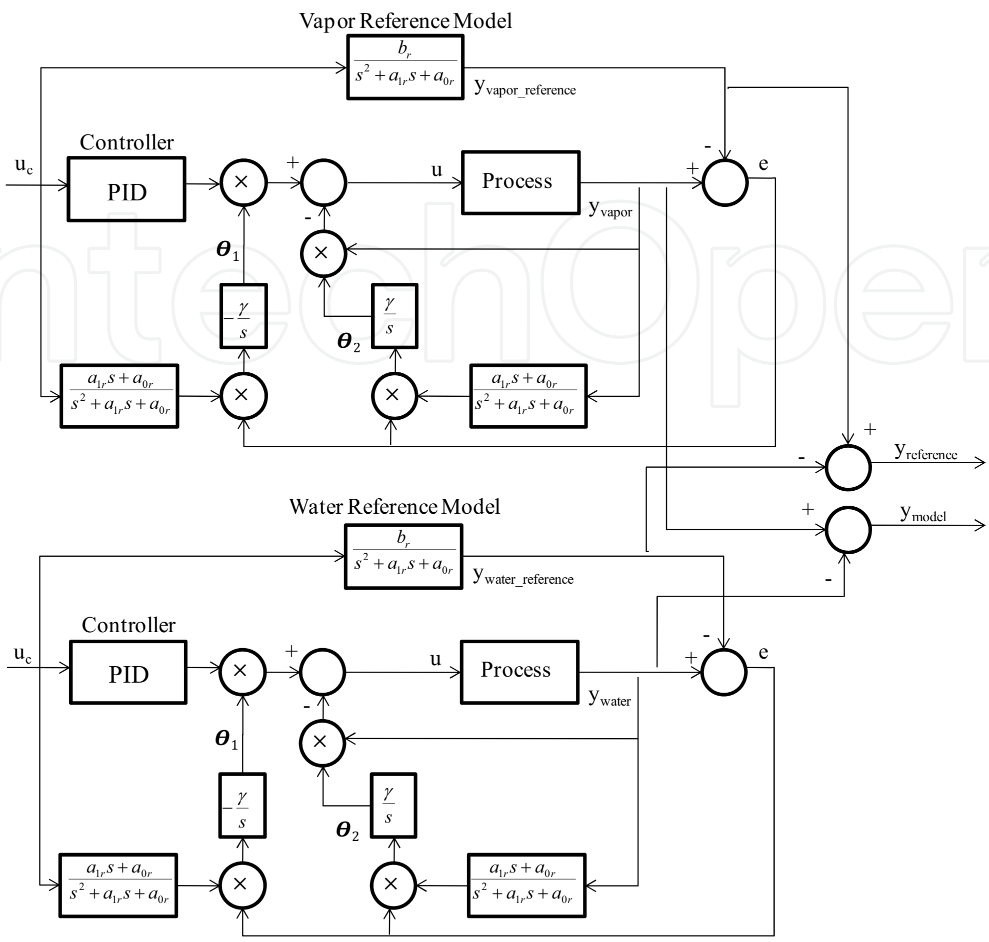

Fig. 15. Fault tolerant MRAC-PID Controller structure.

4.3 Neural Network Controller

In this scheme, we substitute the PID controller by a neural network (shown in figure 16). The neural network was trained with the original process inputs as well as the desired outputs.

4.4 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 figure 17). The process has two inputs: water and steam flows controlled by pneumatic valves, and one output, the water temperature measured by a termocouple. 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.

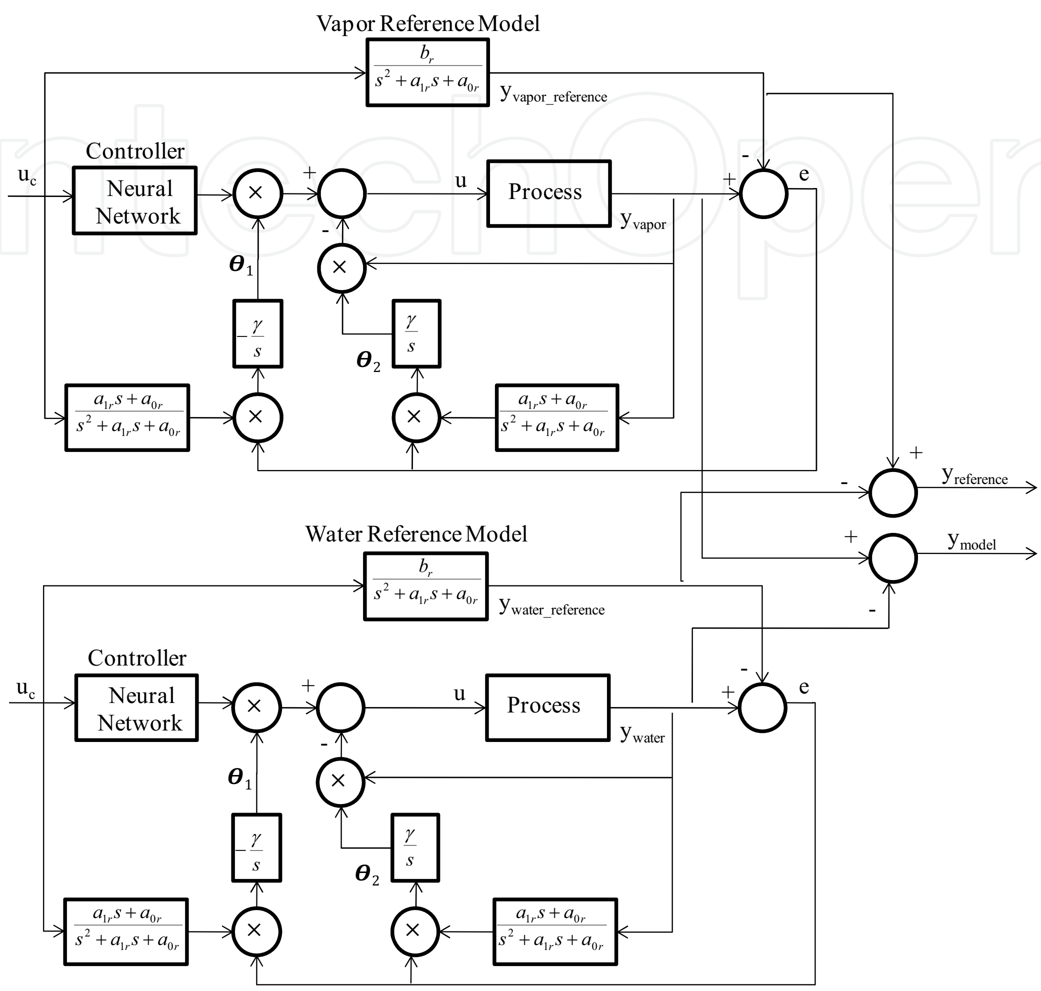

Fig. 16. Fault tolerant MRAC-Neural Network Controller structure.

With the data obtained in the PRBS test, the identification was achieved using software developed in Matlab®.


Fig. 17. Industrial heat exchanger used in the experiments.

The following model was obtained:

$$G_p = G_{vapor} - G_{water}$$
$$G_p = \frac{0.00002}{s^2 + 0.004299s + 0.00002} - \frac{0.000013}{s^2 + 0.007815s + 0.00008} \quad (5)$$

This model was used to implement the 3 different FTC Schemes mentioned above.

4.5 Results

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.

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**
MRAC-PID-Abrupt Fault	- If the fault magnitude is < 1.75, the system is robust against the fault. - If the fault magnitude is between [1.75, 1.97] the system accommodates the fault. -If the fault magnitude is > 1.97, the system becomes unstable. **- MSE=0.00036942**	- If the fault magnitude is 1 the system response varies around +/- 3.5%. This means that the system is degraded but still works. This degradation becomes smaller over time, because the system continues accommodating the fault. **- MSE=0.13283874**
MRAC-PID-Gradual Fault	- If the fault has saturation < +/- 1.75, the system is robust against the fault. - If the fault has a saturation between +/- [1.75, 1.97] the system accommodates the fault. -If the fault has saturation > 1.97, the system becomes unstable. **- MSE=0.0003694**	- 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=0.13508005**

MRAC-Neural Network-Abrupt Fault	- The system is robust against sensor faults - **MSE=0.00030043**	- 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=0.13154736**
MRAC-Neural Network-Gradual Fault	- The system is robust against sensor faults. - **MSE=0.00030043**	- If the fault saturation is +/- 1 the system response varies around +3% and - 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=0.13149647**

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

The following graphs represent a comparison between the different simulated experiments.
Figure 18 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.
Figure 19 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.

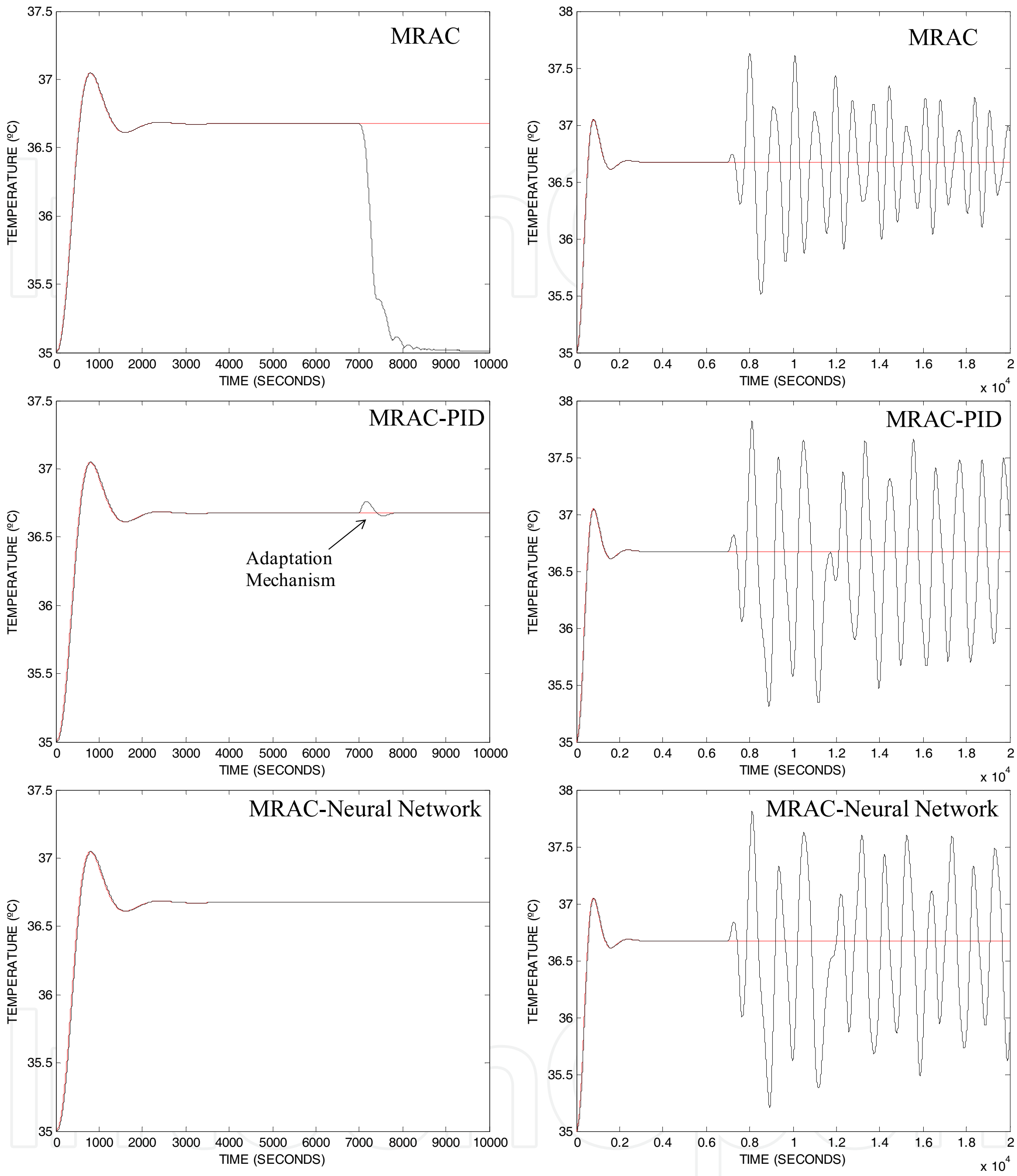

Fig. 18. Abrupt-Sensor Faults (left column) and Abrupt-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

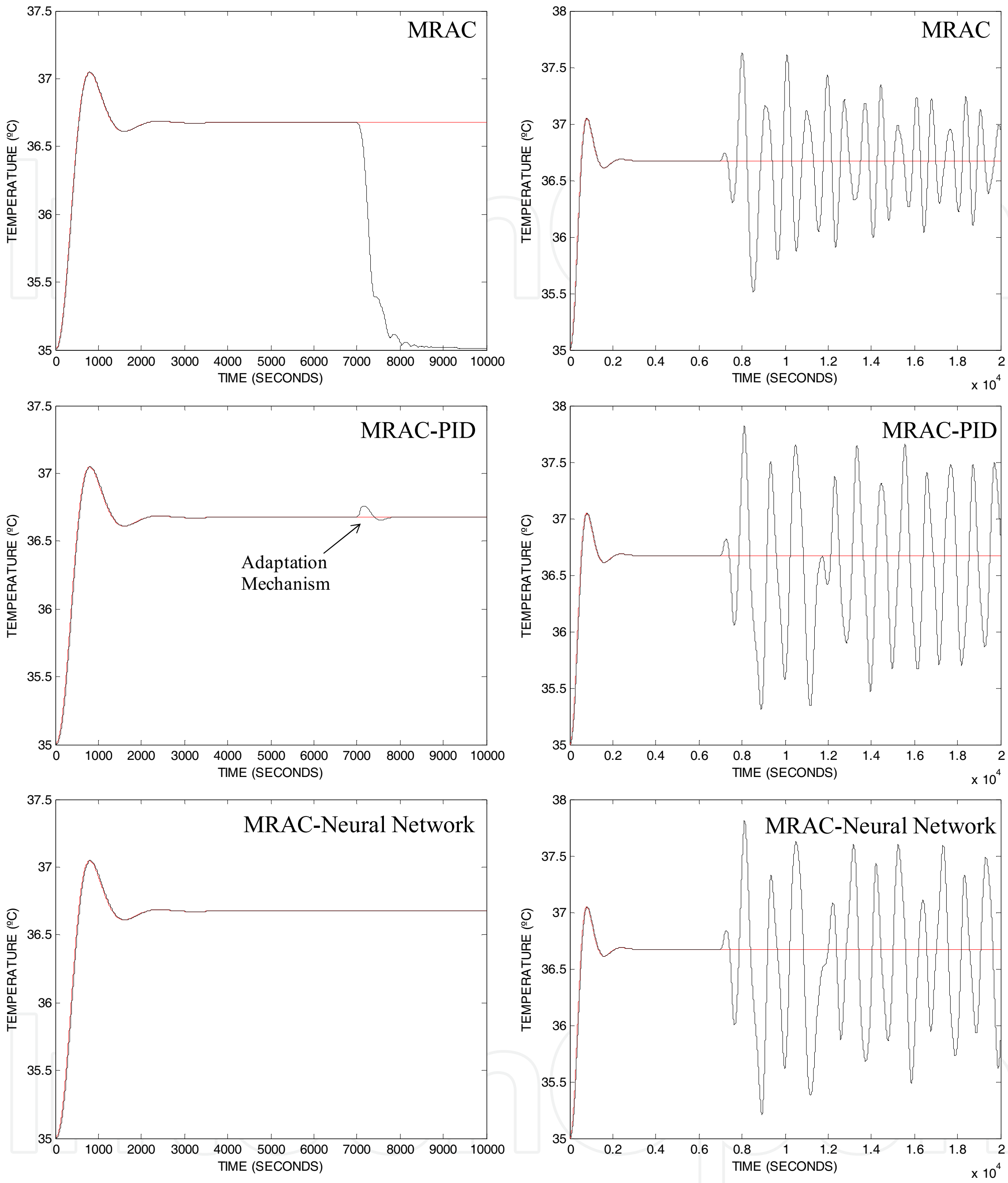

Fig. 19. Gradual-Sensor Faults (left column) and Gradual-Actuator Faults (Right column) of the three different proposed schemes, the fault started at time 7000 secs.

5. References

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.

Bastani, F., & Chen, I. (1988). The role of artificial intelligence in fault-tolerant process-control 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.

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.

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.

Blanke, M.; Kinnaert, M.; Lunze, J. & Staroswiecki, M. (2003). *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, ISBN: 3540010564 , Berlin, Germany.

Blondel, V. (1994). *Simultaneous Stabilization of Linear Systems*. Springer Verlag, ISBN: 3540198628, Heidelberg, Germany.

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.

Diao, Y. & Passino, K. (2001). Stable fault-tolerant adaptive fuzzy/neural control for turbine engine. *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 3, (May 2001) 494–509, ISSN: 1063-6536.

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.

Eterno, J.; Looze, D; Weiss, J. & Willsky, A. (1985). Design Issues for Fault-Tolerant Restructurable Aircraft Control, *Proceedings of 24th Conference on Decision and Control*, pp. 900-905, December 1985, IEEE, Fort Lauderdale, Florida, United States.

Farrell, J.; Berger, T. & Appleby, B. (1993). Using learning techniques to accommodate unanticipated faults. *IEEE Control Systems Magazine,* Vol. 13, No. 3, (June 1993) 40–49, ISSN: 0272-1708.

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.

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.

Gurney, K. (1997). *An Introduction to Neural Networks*, CRC Press Company, ISBN: 1857285034, London, United Kingdom.

Holmes, M. & Ray, A. (2001). Fuzzy damage-mitigating control of a fossil power plant. *IEEE Transactions on Control Systems Technology*, Vol. 9, No. 1, (January 2001) 140– 147, ISSN: 1558-0865.

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.

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.

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

Karsai, G.; Biswas, G.;Narasimhan, S.; Szemethy, T.; Peceli, G.; Simon, G. & Kovacshazy, T. (2002). Towards Fault-Adaptive Control of Complex Dynamic Systems, In: *Software- Enabled Control*, Tariq Samad and Gary Balas, Wiley-IEEE press, 347-368, ISBN: 9780471234364, United States.

Kwong,W.; Passino, K.; Laukonen, E. & Yurkovich, S. (1995). Expert supervision of fuzzy learning systems for fault tolerant aircraft control. *Proceedings of the IEEE*, Vol. 83, No. 3, (March 1995) 466–483, ISSN: 0018-9219.

Liang, B. & Duan, G. (2004). Robust H-infinity fault-tolerant control for uncertain descriptor systems by dynamical compensators. *Journal of Control Theory and Applications*, Vol. 2, No. 3, (August 2004) 288-292, ISSN: 1672-6340.

Lunze, J. & J. H. Richter. (2006). *Control reconfiguration: Survey of methods and open problems.* , ATP, Bochum, Germany.

Mahmoud, M.; Jiang, J. & Zhang, Y. (2003). Active fault tolerant control systems: Stochastic analysis and synthesis, Springer, ISBN: 2540003185, Berlin, Germany.

Mitchell, M. (1996). *An introduction to genetic algorithms*, MIT Press, ISBN: 0262631857, Cambridge, Massachusetts, United States.

Nagrath, .J (2006). *Control Systems Engineering*, Anshan Ltd, ISBN: 1848290039, Indian Institute of Technology, Delhi, India.

Neimann, H. & Stoustrup, J. (2005), Passive fault tolerant control of a double inverted pendulum - a case study. *Control Engineering Practice*, Vol. 13, No 8, 1047-1059, ISNN: 0967-0661.

Nguyen, H.; Nadipuren, P.; Walker, C. & Walker, E. (2002). *A First Course in Fuzzy and Neural Control*, CRC Press Company, ISBN: 158488241, United States.

Oudghiri, M.; Chadli, M. & El Hajjaji, A. (2008). Sensors Active Fault Tolerant Control For Vehicle Via Bank of Robust H∞ Observers. *17th International Federation of Automatic Control (IFAC) World Congress*, July 2008, IFAC, Seoul, Korea.

Passino, K. and Yurkovich, S. (1997). *Fuzzy Control*, Addison-Wesley Longman, ISBN: 020118074, United States.

Pashilkar,A.; Sundararajan, N.; Saratchandran, P. (2006). A Fault-tolerant Neural Aided Controller for Aircraft Auto-landing. *Aerospace Science and Technology*, Vol. 10, pp. 49-61.

Patton, R. J. (1997). Fault-tolerant control: The 1997 situation. *Proceedings of the 3rd IFAC symposium on fault detection, supervision and safety for technical processes,* pp. 1033–1055, Hull, United Kingdom.

Patton, R.; Lopez-Toribio, C. & Uppal, F. (1999). Artificial intelligence approaches to fault diagnosis. *IEEE Condition Monitoring: Machinery, External Structures and Health, I*, pp. 5/1 – 518, April 1999, IEEE, Birmingham, United Kingdom.

Perhinschi, M.; Napolitano, M.; Campa, G., Fravolini, M.; & Seanor, B. (2007). Integration of Sensor and Actuator Failure Detection, Identification, and Accommodation

Schemes within Fault Tolerant Control Laws. *Control and Intelligent Systems*, Vol. 35, No. 4, 309-318, ISSN: 1480-1752.

Polycarpou, M. & Helmicki, A. (1995). Automated fault detection and accommodation: A learning systems approach. *IEEE Transactions on Systems*, Vol. 25, No. 11, (November 1995) 1447–1458.

Polycarpou, M. & Vemuri, A. (1995). Learning methodology for failure detection and accommodation. *IEEE Control Systems Magazine*, Vol. 15, No. 3, (June 1995) 16–24, ISSN: 0272-1708.

Polycarpou, M. (2001). Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach. *IEEE Transactions on Automatic Control*, Vol. 46, No.5, (May 2001) 736–742, ISSN: 0018-9286.

Rumerhart, D.; McClelland, J.; & the PDP Research Group. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*, MIT Press, ISBN: 0262631105, Cambridge, Massachusetts, United States.

Ruan, D. (1997). *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*, Kluwer Academic Publishers, ISBN: 0792399994, United States.

Schroder, P.; Chipperfield, A.; Fleming, P. & Grum, N. (1998). Fault tolerant control of active magnetic bearings. IEEE International Symposium on Industrial Electronics, pp. 573-578, ISBN: 0-7803-4756-0, July 1998, IEEE, Pretoria, South Africa.

Skogestad, S., & Postlethwaite I. (2005). *Multivariable Feedback Control-Analysis and Design*, John Wiley & Sons, ISBN: 9780470011676, United States.

Staroswiecki, M. (2005). Fault tolerant control: The pseudo-inverse method revisited. *Proceedings 16th IFAC World Congress*, pp. Th-E05-TO/2, IFAC, Prague, Czech Republic.

Steffen, T. (2005). *Control reconfiguration of dynamic systems: Linear approaches and structural tests*, Springer, ISBN: 3540257306, Berlin, Germany.

Stengel, R. (1991). Intelligent Failure-Tolerant Control. *IEEE Control Systems Magazine*, Vol. 11, No. 4, (June 1991) 14-23, ISSN: 0272-1708.

Sugawara, E.; Fukushi, M. & Horiguchi, S. (2003). Fault Tolerant Multi-layer Neural Networks with GA Training. *The 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI systems*, pp. 328-335, ISBN: 0-7695-2042-1, IEEE, November 2003 Boston, Massachusetts, United States.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K. & Kavuri, S. (2003a). A review of process fault detection and diagnosis. Part I. Quantitative modelbased methods. *Computers and Chemical Engineering*, Vol. 27, No. 3, 293–311, ISSN-0098-1354.

Venkatasubramanian, V.; Rengaswamy, R. & Kavuri, S. (2003b). A review of process fault detection and diagnosis. Part II. Qualitative models and search strategies. *Computers and Chemical Engineering*, Vol. 27, No. 3, 313–326, ISSN: 0098-1354.

Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. & Yin, K. (2003c). A review of process fault detection and diagnosis. Part III. Process history based methods. *Computers and Chemical Engineering*, Vol. 27, No. 3, 327–346, ISSN: 0098-1354.

Wang, H. & Wang, Y. (1999). Neural-network-based fault-tolerant control of unknown nonlinear systems. *IEE Proceedings – Control Theory and Applications*, Vol. 46, No. 5, (September 1999) 389–398, ISSN; 1350-2379.

Yang, G. & Ye, D. (2006). Adaptive fault-tolerant Hinf control via state feedback for linear systems against actuator faults, *Conference on Decision and Control*, pp. 3530-3535, December 2006, San Diego, California, United States.

Yen, G. & DeLima, P. (2005). An Integrated Fault Tolerant Control Framework Using Adaptive Critic Design. *International Joint Conference on Neural Networks*, Vol. 5, pp. 2983-2988, ISBN: 0-7803-9048-2.

Zhang, D.; Wang Z. & Hu, S. (2007). Robust satisfactory fault-tolerant control of uncertain linear discrete-time systems: an LMI approach. *International Journal of Systems Science*, Vol. 38, No. 2, (February 2007) 151-165, ISSN: 0020-7721.

Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Elsevier Annual Reviews in Control*, Vol. 32, (March 2008) 229-252.

www.intechopen.com

Automation Control - Theory and Practice
Edited by A D Rodi

ISBN 978-953-307-039-1
Hard cover, 350 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

The present edited book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of automation and control. The volume is organized in four parts according to the main subjects, regarding the recent advances in this field of engineering. The first thematic part of the book is devoted to automation. This includes solving of assembly line balancing problem and design of software architecture for cognitive assembling in production systems. The second part of the book concerns different aspects of modelling and control. This includes a study on modelling pollutant emission of diesel engine, development of a PLC program obtained from DEVS model, control networks for digital home, automatic control of temperature and flow in heat exchanger, and non-linear analysis and design of phase locked loops. The third part addresses issues of parameter estimation and filter design, including methods for parameters estimation, control and design of the wave digital filters. The fourth part presents new results in the intelligent control. This includes building a neural PDF strategy for hydroelectric satation simulator, intelligent network system for process control, neural generalized predictive control for industrial processes, intelligent system for forecasting, diagnosis and decision making based on neural networks and self-organizing maps, development of a smart semantic middleware for the Internet , development of appropriate AI methods in fault-tollerant control, building expert system in rotary railcar dumpers, expert system for plant asset management, and building of a image retrieval system in heterogeneous database. The content of this thematic book admirably reflects the complementary aspects of theory and practice which have taken place in the last years. Certainly, the content of this book will serve as a valuable overview of theoretical and practical methods in control and automation to those who deal with engineering and research in this field of activities.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Luis E. Garza Castanon and Adriana Vargas Martinez (2009). Artificial Intelligence Methods in Fault Tolerant Control, Automation Control - Theory and Practice, A D Rodi (Ed.), ISBN: 978-953-307-039-1, InTech, Available from: http://www.intechopen.com/books/automation-control-theory-and-practice/artificial-intelligence-methods-in-fault-tolerant-control

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai

www.intechopen.com

Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen

IntechOpen

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.