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

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
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

Model-Based FDI Schemes For Robot Manipulators Using Soft Computing Techniques

Tolga YÜKSEL and Abdullah SEZGİN
Ondokuz Mayıs University
TURKEY

1. Introduction

While modern control methods were becoming widespread, in addition to demanded repeatability and accuracy specifications, reliability and detection and isolation of probable faults have become an obligation for automatic control systems. In the early 70's, first studies were appeared on this subject. While the first studies on fault detection and isolation (FDI) were implemented for supervisory of chemical processes, following studies were extended to systems like air and spacecrafts, automobiles, nuclear reactors, turbines and HVACs with high reliability mandatories after especially aircraft accidents with high mortality. In 1991, with extending and increasing studies, IFAC SAFEPROCESS comittee was founded and in 1993, this comittee issued some definitions about fault types, fault detection and isolation, fault diagnosis and fault tolerant control (FTC) (Isermann & Ballé, 1997).

Robots are accepted as an assistant subsystem or an individual part of a complex system in most applications. In addition to applications like serial product lines in which they can work harder, faster and with higher accuracy than humans, they are assigned to missions like waste treatment in nuclear reactors, data and sample collection, maintenance in space and underwater tasks which can be very risky for humans. As a consequence, a fault in one product line may cause a pause in all connected lines even in flexible automation systems or a developing and undetected fault may cause abortion of a whole space or underwater mission with big money costs, it may even cause harm to humans. With the increase of these events in real-life applications and with 90's, studies on robot reliability and fault detection and diagnosis in robotics have become common. In addition to these studies, NASA and US Army issued some standards on robots and on the reliability and fault possibilities of robotparts (Cavallaro & Walker, 1994).

This study is focused on model-based FDI schemes, how they can be applied to robot manipulators, how soft computing techniques can be used in these schemes and three different FDI schemes are proposed. Soft computing techniques which can overcome the difficulties of schemes using analytical methods for nonlinear systems are used as modelling, fault isolator and fault function approximator tools in the proposed schemes. In the following section, a literature overview on FDI for nonlinear systems and robot

manipulators is given. In Section 3, defined faults and their physical causes are explained. In Section 4, 5 and 6, the proposed schemes are introduced and explained. Soft computing tools used in these schemes are introduced; furthermore, how they work and what their duties are in these schemes are explained in details. In Section 7, simulation implementations and results of these schemes for a two-link robot are given. In the last section, a comparison of these schemes according to some FDI specifications and future studies on these schemes are given.

Most studies in the literature are interested in sensor faults and locked and free-swinging joint (actuator) faults. This study is interested in abrupt partial actuator faults defined in Section 3 to contribute model-based FDI studies for robot manipulators. Furthermore, most studies using soft computing techniques for FDI are interested in how the parameters in their soft computing tools can be defined in terms of faults and how they can be updated according to faults like adaptive learning. In this study, these tools are used directly and without any modification to give appropriate outputs for appropriate inputs. From this point of view, this study can be accepted as a bridge between model-based FDI and data-based FDI methods. Furthermore, in this study, in addtion to soft computing tools, a hybrid soft computing tool M-ANFIS (multiple-ANFIS) which combines and utilizes benefits of neural network (NN) and fuzzy logic (FL) is used for modelling and function approximation. The two-link robot manipulator used in the simulations can be seen simple but accepted as a test platform for most studies. Besides, it has sufficient specifications for implementation of newly proposed schemes.

2. Literature overview of model-based FDI for nonlinear systems and robot manipulators

Studies and methods on fault detection and isolation can be divided into two main groups: model-based methods and data-based methods (Chen & Patton, 1999). Model-based methods are based on modelling the system and processing the difference signals between the model and the real system named as residuals. Data-based methods are based on processing the input and output signals of the system. Proposed schemes in this study are based on model-based FDI methods and studies on model-based FDI are examined in details. Information and surveys about data-based methods can be found in (Chen & Patton, 1999; Patton et al., 2000a; Venkatasubramanian et al., 2003). Model-based fault diagnosis (detection and isolation) is defined as detection, isolation and characterization of faults in components of a system from the comparison of the system's available measurements, with a priori information presented by the system's mathematical model (Chen & Patton, 1999). According to this definition, model-based FDI methods are formed of two steps. The first step is generation of difference signals called *residuals* between real and predicted or estimated output signals of the system. Discordance of these real and predicted or estimated output signals which means nonzero residual signals indicates a potential fault in the system. The second step is isolation of faults using these obtained difference signals according to a decision set.

Model-based FDI methods can be classified according to the method used for residual generation and the decision set is defined according to the specifications of each residual generation method. The point of classification can not only be the method but also linearity (linear-nonlinear-bilinear) type of the system that will be dealt with. The methods used for

linear systems can be classified into three main titles. The first and mostly used methods are observer based methods. The main idea behind these methods is to estimate the output of the system using the measurements with Luenberger observes and deterministic adjustments or with Kalman filters and stochastic adjustments (Frank & Ding, 1997). Parity vector (relation) methods use definition of parallel or temporal redundancy which is named as defining a variable with two or more definitions and which can be obtained from measurements or analytical relations (Chow & Willsky, 1984). Parameter estimation methods are based on the principle that accepts sudden changes in parameters like friction, mass, viscosity, resistance etc. using system identification methods as a sign of faults (Isermann & Ballé, 1997; Moseler & Isermann, 2000). In addition to proposing new methods, all these methods are investigated for the robustness against disturbances and uncertainties. A detailed survey about these methods and studies can be found in (Chen & Patton, 1999; Patton et al., 2000; Frank & Ding, 1997).

Generally, two main approaches are adopted for FDI for nonlinear systems (Chen & Patton, 1999). First approach linearizes nonlinear models around one or multiple equilibrium(working) points and generates residuals insensitive to parameter changes in small equilibrium point neighbourhoods using robust techniques. This approach may give good and sufficient results only for systems with low level nonlinearities. But this approach is not suitable for nonlinear systems having highly nonlinear terms and wide working points. Second approach, as proposed for solving this problem, uses multiple nonlinear models for each working points. But this approach bring multiple FDI systems for each working points and this will be not practical for real-time implementations.

As mentioned below, to solve these problems, FDI methods which can deal with nonlinear systems directly must be proposed and developed. Therefore, most of the methods proposed for linear systems are adapted for nonlinear systems. One of the approaches is to use analytical or deterministic nonlinear observers (Chen & Patton, 1999; Patton et al. 2000; Frank & Ding, 1997; Adjallah et al.,1994; Garcia & Frank, 1997). Model of a nonlinear systems is accepted as below:

$$\dot{x}(t) = g(x(t), u(t), f(t), d(t))$$
$$y(t) = h(x(t), u(t), f(t), d(t))$$

(1)

where $x(t)$ is the state vector, $y(t)$ is the output vector, $u(t)$ is the input vector, $f(t)$ is the fault vector, $d(t)$ is the disturbance vector and $g(\,.\,,.\,,.\,,.\,)$ and $h(\,.\,,.\,,.\,,.\,)$ are nonlinear functions. FDI problem is generating residuals using the observer form defined in (2):

$$\dot{\xi}(t) = g_r(\xi(t), u(t), y(t))$$
$$r(t) = h_r(\xi(t), u(t), y(t))$$

(2)

and the residuals should prove (3):

$$\|r(t)\| \begin{cases} \approx 0 & f(t) = 0 \\ \gg 0 & f(t) \neq 0 \end{cases}$$

(3)

Here, the aim is to design ξ appropriate for $g_r(\,.\,,.\,,.\,)$ and $h_r(\,.\,,.\,,.\,)$. $\xi(t)$ defines state

estimator (Garcia & Frank, 1997). In the literature, there are some nonlinear observer desing approaches defined for certain type nonlinear systems (Frank & Ding, 1997; Adjallah et al., 1994; Garcia & Frank, 1997; Seliger & Frank, 1991; Yang & Saif, 1995; Kinnaert, 1999).

All approaches mentioned below use analytical methods but it is hard to obtain analytical models which nonlinear observers are based on. To overcome this problem, "universal approximator" soft computing modelling tools which can model nonlinear systems are preferred. Soft computing techniques involves NN, FL and genetic algorithms (GA). NN can be used to model multiple-input-multiple-output (MIMO) nonlinear systems by using nonlinear mapping capabilities in its hidden black-box structure (Haykin, 1999). Residual generation is implemented by comparing the real system outputs and estimated outputs by NN. Furthermore NNs are superior to analytical techniques on classification and a second NN can be used to isolate faults by evaluating (classifying) the residuals (Patton et al., 2000b; Marcu et al., 1998). Model-based FDI scheme with NNs is shown in Fig. 1:
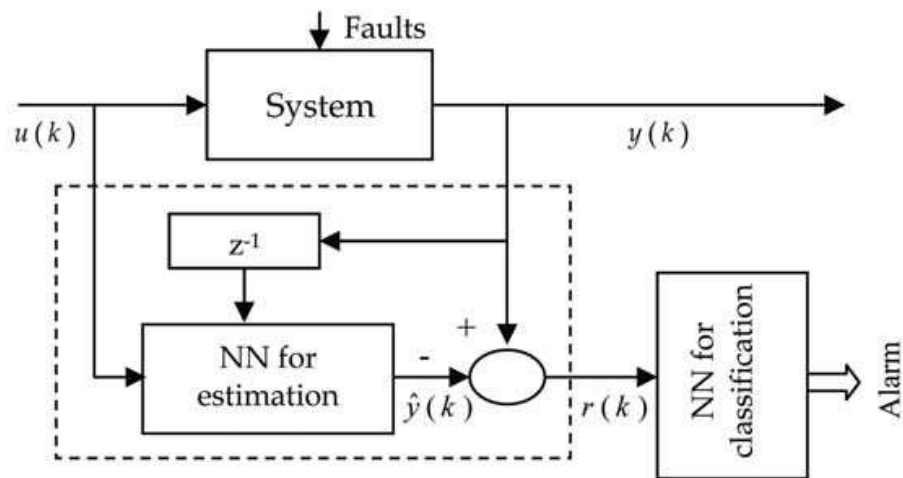

Fig. 1. Model based fault detection and isolation with two neural networks

In addition to these NN approaches, there are some approaches named *online approaches* using adaptive learning and defining NN parameters in terms of input signals anf fault functions (Zhang et al., 2002; Polycarpou & Helmicki, 1995; Polycarpou & Trunov, 2000)

NNs with self learning capabilities can be thought convenient for FDI but, as a black-box structure, keeping user experiences and interferences away from modelling is not a desired feature. To overcome this disadvantage, FL is used as modelling, observer, adaptive residual threshold selector tools for FDI for nonlinear systems (Dexter, 1995; Patton et al., 1998; Ballé, 1998). The main idea behind fuzzy observers is to define a nonlinear system using a set of locally linearized observers with Takagi-Sugeno fuzzy models. Local region definitions are generated using working points. Furthermore, FL is used for residual evaluation. The most important problem in residual evaluation is to set residual threshold. An adaptive threshold can prevent the FDI system from false and missed alarms and FL can be used as the adaptive threshold selector tool (Schneider & Frank, 1996).

FDI studies on robots continue on most of the robot types (mobile, flexible, kinematically redundant, parallel, mobile manipulator, humanoid, bipedal and multi-legged etc.). This study deals with the most common used serial, open chained and rigid robot manipulator and only a detailed review on this type of robots is given here. Studies on other type robots

can be found in (Goel et al., 2000; Tinós & Terra, 2002). Most studies on FDI for robot manipulators are based on nonlinear observer approaches. Caccavale and Walker tried to adapt robot dynamics to a certain kind of nonlinear systems and used nonlinear observers which are convenient for these systems (Caccavale & Walker, 1997). Similarly, Schneider and Frank used a robust observer designed for nonlinear systems for robot dynamics and fuzzy logic for residual evaluation/fault isolation (Schneider & Frank, 1996). Leuschen et al. transferred analytical redundancy from linear to nonlinear systems, they defined nonlinear analytical redundancy term and they implemented these redundancies and FDI studies on a hydraulic robot manipulator platform and on a two-link IMI robot manipulator (Leuschen et al., 2005). De Luca and Mattone resembled robot dynamics to generalized notation of adaptive controllers and proposed an adaptive FDI scheme (De Luca & Mattone, 2004). Dixon et al. proposed a filter, they passed torque signals applied to a robot manipulator through this filter and they proposed an FDI technique robust to parametric uncertainties using the difference signals between these filtered signals and their predictions (Dixon et al. 2000). Abdul and Liu proposed an analytical method for the prediction of position and velocity signals of modular type robots and designed a fault tolerant controller using these predictions (Abdul & Liu, 2008). Brambilla et al. generated residuals using inverse robot model, proposed a sliding mode observer and isolated sensor faults using this observer (Brambilla, 2008). Chen and Saif resembled robot dynamics to systems with unknown inputs in state-space and implemented fault detection using output observers (Chen & Saif, 2008).

NNs as an approved tool for FDI for nonlinear systems are also used for robot manipulators. Naughton et al. used nonlinear observer proposed by Adjallah et al. for residual generation and NN for residual evaluation (Adjallah et al., 1994; Naughton et al., 1996). Vemuri and Polycarpou considered fault as a component of robot model function and used adaptive learning strategy of NNs to approximate fault function (Vemuri & Polycarpou, 1997). Terra and Tinós used some different types of NN structures for both residual generation and evaluation (Terra & Tinós, 2001). Lee et al. tried to use parameter identification methods for fault detection and ART type NNs for fault isolation on component and sensor type faults (Lee et al., 2003). Datta et al. tried to classify coefficients obtained from discrete wavelet transform (DWT) using a NN (Datta et al., 2007).

3. Faults defined for robot manipulators

Faults can be classified according to the part of the system, according to modelling or according to time characteristics (Chen & Patton, 1999). In this section, how faults defined for robot manipulators are classified according to the part of the system (Fantuzzi et al., 2003).

Generalized dynamics of robot manipulators are defined in (4) :

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(\dot{q}) = \tau \tag{4}$$

In (4), n as the number of links, $q(t), \dot{q}(t), \ddot{q}(t) \in \Re^n$ are in order angular position, velocity and accelerations of each link, $M(q) \in \Re^{n \times n}$ is the positive defined inertia matrix, $V(q,\dot{q}) \in \Re^n$ is the Coriolis and centripedal vector, $G(q) \in \Re^n$ is the gravity vector, $F(\dot{q}) \in \Re^n$ is the friction vector and $\tau \in \Re^n$ is the vector of applied torques to joints. If the nonlinear terms in (4) except terms having angular acceleration are expressed as $N(q,\dot{q}) = V(q,\dot{q}) + G(q) + F(\dot{q})$

(4) becomes

$$M(q)\ddot{q} + N(q,\dot{q}) = \tau \tag{5}$$

Faults are defined in (5) as follows:

a) **Actuator faults**: Motors and power transmission tools are charged as actuators for manipulators and faults occured in these actuators affects the ability of movement. These type of faults change dynamics as follows:

$$M(q)\ddot{q} + N(q,\dot{q}) + u(t-T)f(t) = \tau \tag{6}$$

- Locked joint faults: These faults occur when the magnetic brake of the motor connected to a joint is locked and doesn't allow any movements (τ free - q_i fixed).
- Free-swinging joint faults: These faults occur when applied torque of the motor connected to a joint is zero because of a disconnected cable (τ zero - q_i free and under impact of other joints or gravity).
- Partial actuator faults: These faults occur when applied torque of the motor connected to a joint decreases (i.e. %20 decrease etc.) because of a fault at power electronics components.

b) **Component faults**: In robot manipulators, broken link, gear corrision or fixed or slipped chain can be component faults. Dynamics of the manipulator change as follows:

$$M(q)\ddot{q} + (N(q,\dot{q}) + f_N) = \tau \tag{7}$$

c) **Sensor faults**: Optical encoders for joint positions, tachogenerators for joint velocities and tactile sensors for contact forces are the sensors used in robot manipulators. The faults defined for all sensors can be seen in these sensors (bias etc.). Dynamics of the manipulator change as follows:

$$y(t) = [q_1(t) \cdots q_n(t) \ \dot{q}_1(t) \cdots \dot{q}_n(t)]^T + \Delta f_{q,\dot{q}} \tag{8}$$

In (6), (7) and (8), $u(t\text{-}T)$ is the delayed unit step function, T is the fault occurence instant, f_N is the component fault, $\Delta f_{q,\dot{q}}$ is the sensor fault.

4. FDI scheme with M-ANFIS and NN

The block diagram of the first scheme is shown in Fig. 2. The diagram includes not only the FDI block but also the control block to expose the whole system. To control robot, Computed Torque-PID (CT-PID) method is used and a brief introduction about this method is given in Section 6. CT-PID method needs exact robot model but it can cope with bounded uncertainties. M-ANFIS as the first block of FDI scheme is formed of multiple independent ANFIS structure and implements robot modelling. It takes instant torque values of each joint and unit time step delayed angular position and velocity values of each link. The model gives estimated values of instant angular position and velocity values of each link. Difference signals between real robot and robot model generate residuals. Fault detection alarm is accepted as overshooting of predefined thresholds by residuals obtained from

healthy robot operations or simulations. Fault isolation process is implemented by classification of the generated residuals applied to a multilayer NN. In subsections, a review on M-ANFIS and NN with resilient propagation will be given.
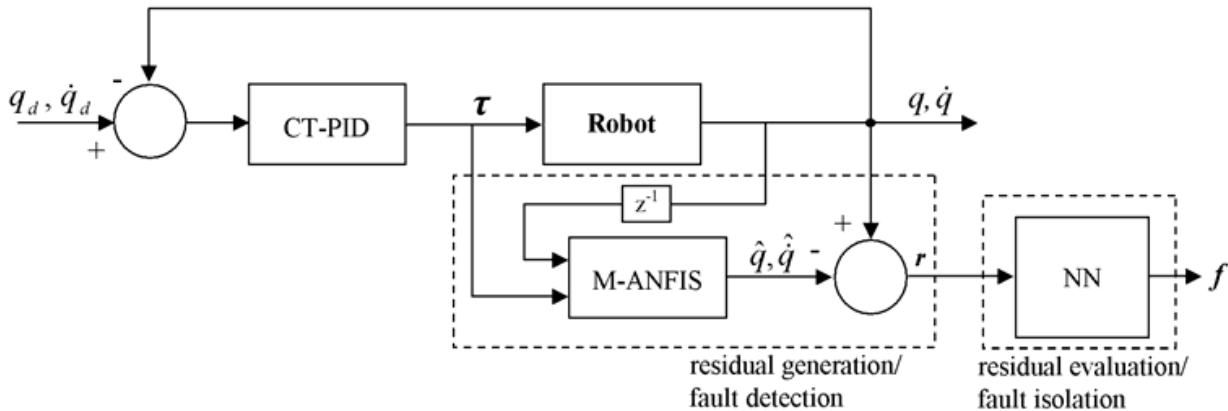

Fig. 2. The block diagram of the first FDI scheme

4.1 Residual generation with M-ANFIS

Difficulties in modelling nonlinear systems analytically, robustness obligations to uncertainty and disturbance directed researchers to soft computing techniques and structures with self, automatic learning and nonlinear mapping capabilities. Soft computing techniques involve NNs, FL, GA and hybrid structures of these tools. Despite NNs have lots of types and lots of learning algorithms, they behave like a black box due to their self-learning nature. Alike FL leaves all parametric adjustments to users, users' experiences become a parameter in modelling and performance of adjustments are dependent on the users. Jang et al. considered advantages of both structures and they decided to combine these advantages and proposed an adaptive network called ANFIS (Adaptive Neuro Fuzzy Inference System) which is functionally equivalent to fuzzy inference system (Jang et al., 1997). ANFIS has a 5-layered structure and a sample with 2 inputs is shown in Fig. 3. Functions of the layers are given below.
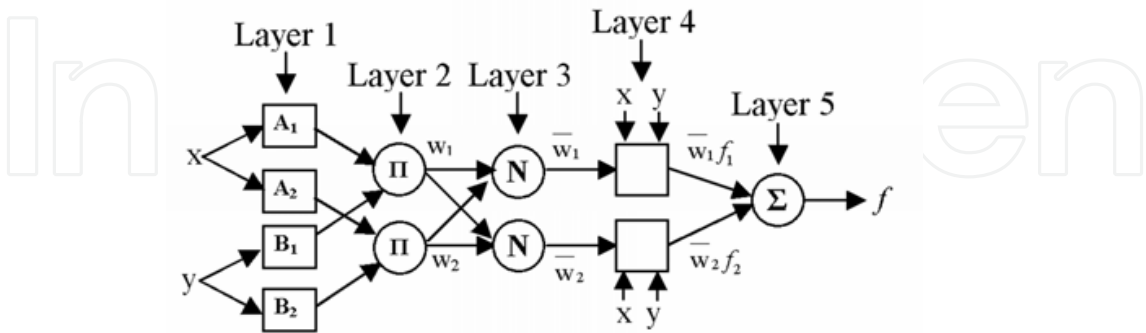

Fig. 3. General ANFIS structure

Layer 1: This layer contains membership functions of inputs as defined in (9) and all inputs are applied to these functions. Type and shape of the membership functions are defined by the user and generally, these functions are bell-shaped functions defined in (10):

$$L1x_i = \mu_{Ai}(x) \quad i = 1,2$$
$$L1y_i = \mu_{Bi}(y) \quad i = 1,2 \tag{9}$$

$$\mu_{Ai}(x) = \frac{1}{1 + \left|\dfrac{x - c_{Ai}}{a_{Ai}}\right|^{2b_{Ai}}} \quad , \quad \mu_{Bi}(y) = \frac{1}{1 + \left|\dfrac{y - c_{Bi}}{a_{Bi}}\right|^{2b_{Bi}}} \quad i = 1,2 \tag{10}$$

where a_i is center, c_i is width and b_i is crossover gradient, (a_i, b_i, c_i) are parameters of defined function and named as premise parameters.

Layer 2: Each function value is multiplied by other values coming from other inputs due to defined rule base and the result values are named as firing strength of each rule:

$$L2_i = w_i = \mu_{Ai}(x).\mu_{Bi}(y) \quad i = 1,2 \tag{11}$$

Layer 3: Firing strengths are normalized:

$$L3_i = \overline{w}_i = \frac{w_i}{\displaystyle\sum_{j=1}^{2} w_j} \quad i = 1,2 \tag{12}$$

Layer 4: Normalized firing strengths are multiplied by a first order function of inputs:

$$L4_i = \overline{w}_i.f_i = \overline{w}_i(p_i x + q_i y + r_i) \quad i = 1,2 \tag{13}$$

where (p_i, q_i, r_i) are parameters of a first order function and these parameters are named as consequent parameters.

Layer 5: Values coming from all Layer 4 outputs are summed and output value is obtained.

$$L5 = \sum_{i=1}^{2} \overline{w}_i f_i \tag{14}$$

Points that use user experience in ANFIS like in FL are the choices of membership function types at Layer 1 and multiplication operation due to rule base arrangement at Layer 2. These dependencies are exposed in the demonstration of functional equivalence of ANFIS and fuzzy inference system under some circumstances in (Jang et al., 1997).

It is purposed in all network structures to adapt or update network parameters in order to give appropriate outputs against appropriate inputs. From this point of view, ANFIS updates its own parameters using learning algorithms like NNs. As a learning algorithm, backpropagation or hybrid learning expressed in forward and backward passes can be preferred. Table 1 explains hybrid learning. In forward pass, while premise parameters are fixed, inputs go forward until Layer 4 and consequent parameters are determined with least

squares. In backward pass, consequent parameters are fixed, error is backpropagated until Layer 1 and premise parameters are determined with gradient descent.

Parameters	Forward pass	Backward pass
Premise	Fixed	Gradient descent
Consequent	LSE	Fixed

Table 1. Parameter updates for hybrid learning in two passes

Fig. 3 illustrates the main disadvantage of ANFIS, being multi-input-single-output (MISO). To model systems with multiple outputs (MIMO), multiple-ANFIS (M-ANFIS) which has independent parameters and outputs is used. A structure that considers correlations between outputs and uses some mutual parameters in order to decrease computational load coming from increasing parameter number exists and it is named as coactive-ANFIS (C-ANFIS) (Jang et al., 1997). In this study, M-ANFIS is preferred and M-ANFIS with 2 inputs and 2 outputs is illustrated in Fig. 4.
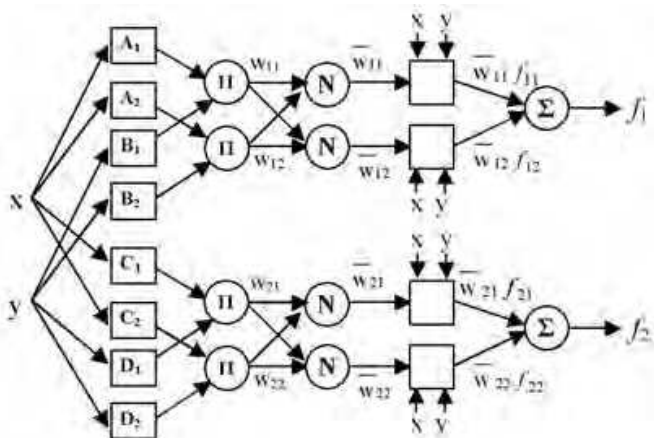

Fig. 4. M-ANFIS with 2 inputs-2 outputs

Robot model takes instant torque and unit time step delayed position and velocity signals belonging to each joint as inputs and generates instant predictions of position and velocity signals of each joint using an ANFIS for each signal. Residuals are generated from differences between real robot and robot model signals. Residual generation using M-ANFIS is shown in Fig. 5.
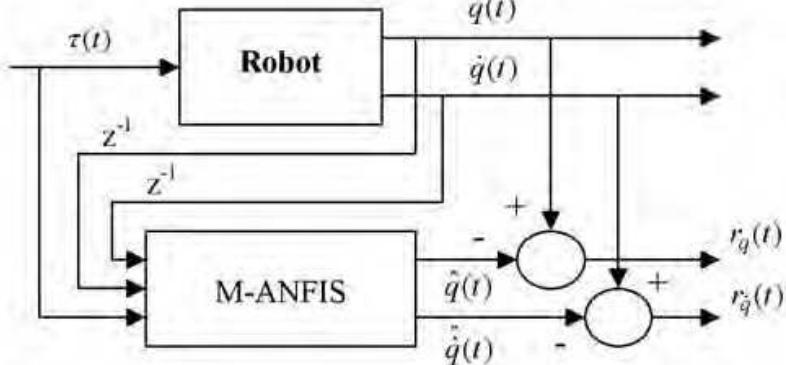

Fig. 5. Residual generation with M-ANFIS

If one of the residual signals overshoots the thresholds determined by simulations and implementations running under healthy conditions, it informs a fault alarm. In this study, in addition to generated residuals, signals called analytical redundancies are used for fault detection and isolation. These signals will be outlined in the following section.

4.2 Residual evaluation using NN with resilient propagation and analytical redundant signals

Residual generation is followed by residual evaluation to isolate faults. This operation is based on the fact that different types of faults show different residual characteristics and it is considered as a classification process. All techniques used for classification and pattern recognition can be used for this process. Multilayer feedforward NNs are very convenient in soft computing based classification tools (Haykin, 1999). (Haykin, 1999) gives all detailed information about NNs. Here, just a review on the used learning algorithm will be given.

NNs aim to update their defined parameters with learning algorithms to give appropriate outputs for appropriate inputs. The most common learning algorithm for this purpose is gradient descent algorithm and it is defined in (15):

$$w_{ij(n+1)=}w_{ij}(n) + \Delta w_{ij}(n)$$
$$\Delta w_{ij}(n) = -\eta.\frac{\partial E(n)}{\partial w_{ij}} \tag{15}$$

where $E(n)$ as the n. step error function, w_{ij} as the weight from neuron i to neuron j and η as the learning rate parameter. The updates are dependent on the learning rate parameter η and it is known that if it is chosen too small, too many operation steps will be needed and if it is chosen too large, minimum value will not be reached and error value will oscillate around it. To avoid this problem and to accelerate converge, a momentum term with μ momentum parameter is added to (15):

$$\Delta w_{ij}(n) = -\eta.\frac{\partial E(n)}{\partial w_{ij}} + \mu.\Delta w_{ij}(n-1) \tag{16}$$

Despite the momentum term in (16), observations showed that this regulation is still dependent on the selected momentum parameter. And again to avoid these parameter dependencies, adaptive learning and momentum parameters are suggested.

These regulations neglect that weight updates are dependent not only the learning rate but also partial derivatives of $E(n)$ with respect to w_{ij}. Resilient Propagation (RP) learning algorithm removes this blurred adaptation from updates and performs updates directly according to the following steps (Riedmiller & Braun, 1993). Firstly, RP assigns Δ_{ij} update values to all weights. These values are updated as defined in (17) and (18):

(17) expresses that if the partial derivative of error with respect to weight changes its sign, the update value is too big and local minima is missed, it should be decreased by η^- factor taking values between 0 and 1 and if it remains with the same sign, it should be increased by η^+ factor taking values greater than 1. After the update value is calculated, the weight update is performed in (18):

$$
\Delta_{ij}(n) = \begin{cases} \eta^+ . \Delta_{ij}^{(n-1)} & \textbf{\textit{if }} \dfrac{\partial E(n-1)}{\partial w_{ij}} . \dfrac{\partial E(n)}{\partial w_{ij}} > 0 \\[2mm] \eta^- . \Delta_{ij}^{(n-1)} & \textbf{\textit{if }} \dfrac{\partial E(n-1)}{\partial w_{ij}} . \dfrac{\partial E(n)}{\partial w_{ij}} < 0 \\[2mm] \Delta_{ij}^{(n-1)} & \textbf{\textit{else}} \\[2mm] 0 < \eta^- < 1 < \eta^+ \end{cases}
\tag{17}
$$

$$
\Delta w_{ij}^{(n)} = \begin{cases} -\Delta_{ij}^{(n)} & \textit{if } \dfrac{\partial E(n)}{\partial w_{ij}} > 0 \\[2mm] +\Delta_{ij}^{(n)} & \textit{if } \dfrac{\partial E(n)}{\partial w_{ij}} < 0 \\[2mm] 0 & \textit{else} \end{cases}
\tag{18}
$$

On the contrary of other learning algorithms, RP is more transparent, has a more powerful update process and is more efficient with respect to time, memory storage consumption and it is chosen as the learning algorithm in this study.

Simulation results in this study and in some other studies showed that it is hard to train a NN with sufficient fault isolation rates just using existing residual signals (Chen & Patton, 1999; Leuschen et al., 2005). To assist isolation process, some variant signals must be generated. This can be done by using the definition of analytical redundancy as defining one variable in two or more ways like derivation of position and integral of acceleration for velocity. It is clear that the derivative value of position residuals must be equivalent to velocity residuals mathematically and these derivatives can be applied to NN for isolation process and can be used for fault detection. It must be noted that these redundant signals are just used to help fault isolation (classification) process and these signals are not directly sensitive to a fault defined in (Leuschen et al., 2005). Definitions of analytical redundant signals and fault detection process in this study are given in (19) and (20), respectively. NN for fault isolation is shown in Fig. 6.

$$
\begin{aligned}
(r_q(t))' &= (q(t) - \hat{q}(t))' \\
&= \dot{q}(t) - \dot{\hat{q}}(t)
\end{aligned}
\tag{19}
$$

$$
\begin{aligned}
r_q(t) &> threshold\_up_{r_q} \\
r_q(t) &< threshold\_down_{r_q} \\
r_{\dot{q}}(t) &> threshold\_up_{r_{\dot{q}}} \\
r_{\dot{q}}(t) &< threshold\_down_{r_{\dot{q}}} \quad \Rightarrow \quad \textit{fault alarm} \\
(r_q(t))' &> threshold\_up_{(r_q)'} \\
(r_q(t))' &< threshold\_down_{(r_q)'}
\end{aligned}
\tag{20}
$$

$$r_q(t) = q(t) - \hat{q}(t) \rightarrow$$
$$r_{\dot{q}}(t) = \dot{q}(t) - \hat{\dot{q}}(t) \rightarrow$$
$$(r_q(t))' = (q(t) - \hat{q}(t))' \rightarrow$$
$$= \dot{q}(t) - \hat{\dot{q}}(t)$$
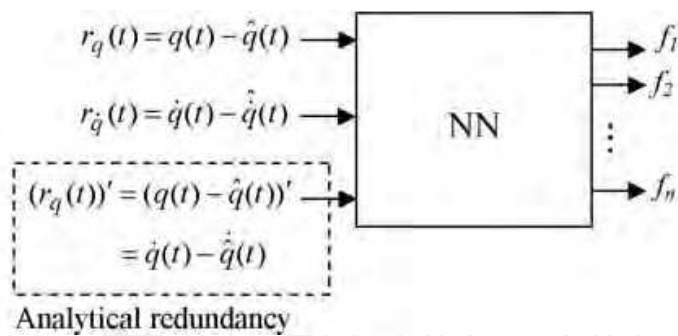
Analytical redundancy

Fig. 6. Fault isolation with NN using residuals and analytical redundancy

5. FDI scheme using generalized observers with M-ANFIS

The design procedure of fault isolation using observes is based on defining relationships between faults and generated residuals. If the residual set can isolate all faults, it can be said that the residual set has the required isolation property. Two methods can be applied to residual sets that involve all residuals to verify this fault isolability property (Chen & Patton, 1999). For dedicated observer schemes (DOS), as the first method, each of the residuals must be sensitive to one fault and insensitive to others. Although this method sounds good, in practice, it is hard to design and to obtain robustness against modelling errors for dedicated observers using analytical approaches. For generalized observer schemes (GOS), as the second method, each of the residuals is sensitive to all but one fault and it is easier to design generalized observers using generated residuals. GOS approach can easily cope with uncertainties by modelling faulty systems one by one. Nevertheless, computational load coming from each system models arises real-time implementation problem. The block diagram of the second proposed model-based FDI scheme is shown in Fig. 7.
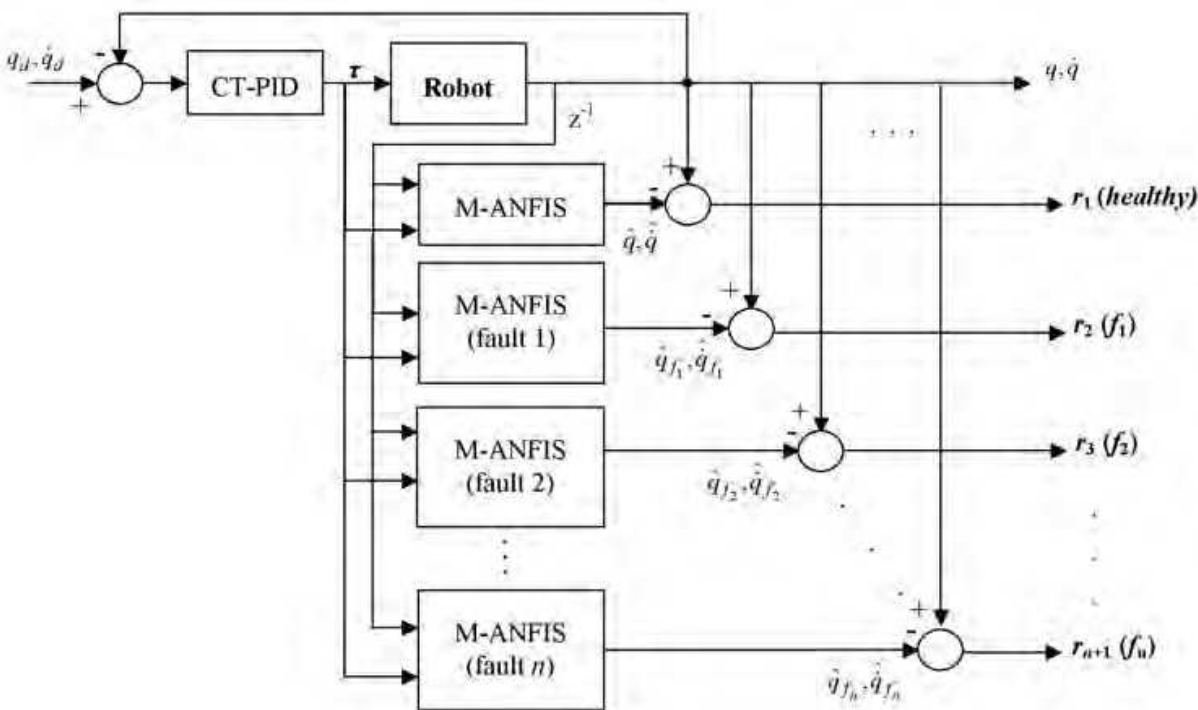

Fig. 7. The block diagram of the second FDI scheme

Again CT-PID is preferred as the robot control method. For the robot manipulator, $n+1$ systems, as one healthy and n faulty, are defined. Each system is modelled using M-ANFIS and generates residual sets (position and velocity residuals of each system) using the same definitions in Section 3.1. Using GOS approach, each of the residual sets overshoots thresholds of all other systems but itself determined by simulations or implementations. GOS combines fault detection and isolation process in one step and that makes the method more effective and attractive. But using models for each defined fault exposes heavy computational load. Table 2 illustrates residuals and residual evaluation process for the proposed scheme.

Faults	r_1	r_2	r_3	…	r_{n+1}
Healthy	0	1	1	1	1
f_1	1	0	1	1	1
f_2	1	1	0	1	1
…	1	1	1	0	1
f_n	1	1	1	1	0

Table 2. Residual evaluation for generalized observer scheme

6. Fault function approximator FDI scheme with M-ANFIS

Despite two proposed schemes are valid and effective for fault detection and isolation, due to their nature, they can be used only for predefined fault types and that makes them hard against partial actuator faults for nonlinear systems and robot manipulators. Furthermore, these schemes can not fully succeed in the design of fault tolerant controllers that use information coming from FDI.

These disadvantages makes fault function approximation more important and the third proposed scheme is focused on fault function approximation. The block diagram of the third scheme is shown in Fig. 8. Modelling part of the scheme is implemented using the same definitions in Section 3.1. In the design of this scheme, to approximate faults, firstly, a M-ANFIS that takes residuals and redundant signals as inputs is trained but the results are vain. As the second attempt, a M-ANFIS that uses instant torque, position and velocity signals as inputs is trained. Each ANFIS of M-ANFIS gives a fault function approximation of each actuator belonging to each joint. Activation of M-ANFIS and fault detection process are
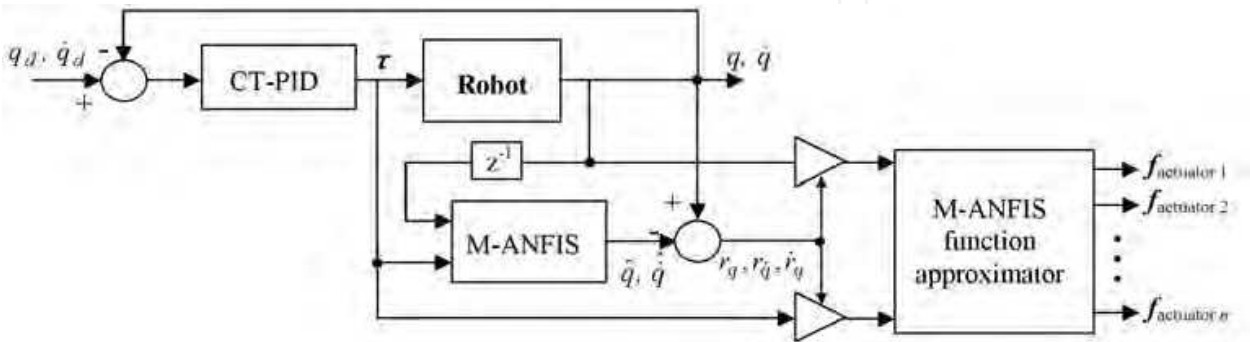

Fig. 8. The block diagram of the third FDI scheme

implemented by residuals and analytical redundant signals. M-ANFIS is designed to give one fault function, the other outputs will give zero value. That makes the fault isolation process unnecessary but simulation results showed that especially for small faults more than one output may give approximations. In this study to avoid this problem, bigger output is accepted as the real fault function approximation.

7. Simulation results

In this section, the proposed FDI schemes are simulated using MATLAB *Fuzzy Logic Toolbox* and *Neural Network Toolbox*. Robot manipulator used in the simulations is a two-link planar manipulator under gravity and masses of the links are defined at the end of the links (Lewis et al., 1993). The manipulator is shown in Fig. 9.
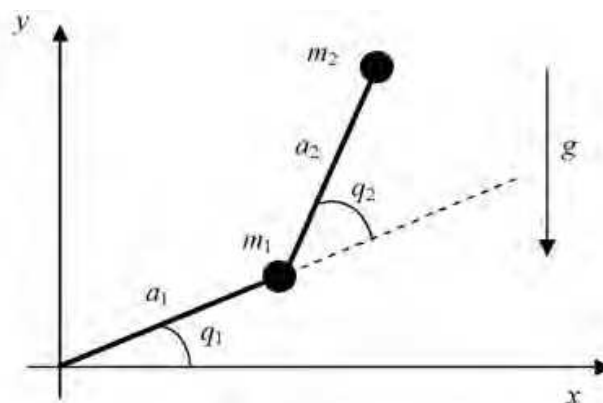
Fig. 9. Two-link manipulator under gravity

The dynamics of the manipulator and the generalized form of dynamics are given in (21) and (22), respectively:

$$
\begin{aligned}
&M_{11} = (m_1 + m_2)a_1{}^2 + m_2 a_2{}^2 + 2m_2 a_1 a_2 \cos(q_2) \\
&M_{12} = M_{21} = m_2 a_2{}^2 + m_2 a_1 a_2 \cos(q_2), \ M_{22} = m_2 a_2{}^2 \\
&V_1 = -m_2 a_1 a_2 (2\dot{q}_1 \dot{q}_2 + \dot{q}_2{}^2)\sin(q_2) \\
&V_2 = m_2 a_1 a_2 \dot{q}_1{}^2 \sin(q_2) \\
&G_1 = (m_1 + m_2)g a_1 \cos(q_1) + m_2 g a_2 \cos(q_1 + q_2) \\
&G_2 = m_2 g a_2 \cos(q_1 + q_2) \\
&M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \ V(q,\dot{q}) = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \ G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}
\end{aligned}
\tag{21}
$$

$$
\tau = M(q)\ddot{q} + V(q,\dot{q}) + G(q) = M(q)\ddot{q} + N(q,\dot{q}) \tag{22}
$$

where $\tau \in \Re^2$ is the applied torques to joints, $M(q) \in \Re^{2 \times 2}$ is the inertia matrix, $V(q,\dot{q}) \in \Re^2$ is the Coriolis/centripedal vector and $G(q) \in \Re^2$ is the gravity vector. Friction and disturbance terms are neglected. The link masses are $m_1 = m_2 = 1$ kg., the link lengths are $a_1 = a_2 = 1$ m. and the sampling frequency is 100 Hz.

Computed Torque control is a special application of feedback linearization of nonlinear systems to robotics. It has commonly PD or PID types and CT-PID is used in this study (Lewis et al., 1993). CT-PID is not very effective against uncertainties but in this study, the controller is not important because suggested FDI schemes use soft computing techniques and they use only the datas coming from the system. If another controller is used, only datas will change and soft computing tools will be trained again using these new datas. Equations of CT-PID are given in (23):

$$\dot{\varepsilon} = e \quad , \quad e = q_d - q$$
$$\tau = M(q)(\ddot{q}_d + K_d \dot{e} + K_p e + K_i \varepsilon) + N(q, \dot{q}) \tag{23}$$

where e is the error, ε is the derivative of error. The gain matrices of the controller are $K_d = 100 \ I_{2\times2}$, $K_p = 20 \ I_{2\times2}$, $K_i = 500 \ I_{2\times2}$.

In this study, 4 different partial actuator faults are defined, examined and they are given in Table 3 with loss percentages. Expression of partial actuator faults defined in (6) with respect to time in (22) is given in (24).

$$M(q(t))\ddot{q}(t) + N(q(t), \dot{q}(t)) = \tau(t) - \alpha.u(t - T).\tau(t) \tag{24}$$

where $\alpha \in \Re^2$ is the partial actuator loss, $u(t) \in \Re^2$ is the unit step and T is the fault occurence instant.

Fault	Name
Actuator 1 %50 loss	f_1
Actuator 1 %30 loss	f_2
Actuator 2 %50 loss	f_3
Actuator 2 %30 loss	f_4

Table 3. Simulated faults

7.1 Case 1: FDI with the first scheme

As the first case, FDI scheme in Fig. 2 is implemented. Firstly, M-ANFIS is constructed as the model. M-ANFIS is formed of 4 independent ANFIS having instant torque and unit time step delayed position and velocity signals as common 6 inputs coming from 2 joints and each giving assigned instant position or velocity signal of each joint. Manipulator is simulated for 146 different joint trajectories defined in sin-, cos-shape with amplitudes varying between ±1 and 76 sampled datas from these simulations are used to train M-ANFIS. Each ANFIS has two bell-shaped membership functions for each input and hybrid learning is selected as the learning algorithm. To show the robustness of the model and residuals generated, time varying %5 dynamics uncertainty is added to (22) as accepted in (25) and the schemes are tested against different uncertainties added to robot dynamics.

$$\|M(q)\| \le 1.05\|M(q)\|, \|N(q,\dot{q})\| \le 1.05\|N(q,\dot{q})\| \tag{25}$$

Difference signals between this robot model and real robot generate residuals. Although residuals must be zero in healthy conditions, it may have very small values due to uncertainties. 12 trajectories of 146 trajectories are selected, up and down boundary thresholds of residuals generated from healthy 12 different trajectories are determined and any overshooting is defined as the fault alarm and fault detection part of the scheme.

A four-layered NN with 10-20-20-4 neurons using RP learning algorithm is constructed for residual evaluation/fault isolation. Four faults defined in Table 4 for the same 12 different trajectories are simulated, 4 residuals and 2 analytical redundant (position residual derivatives of each link) signals with 101 samples (6×4848) as 6 inputs and 4 outputs giving 0.9 value, each representing one fault in Table 4, are used to train NN. 7000 epochs and 0 error target are selected as training parameters. Training process is accomplished under 5 minutes and error value is under 10^{-3}.

Simulation tests showed that NN is confused by residuals during transition from healthy to faulty condition and this increases false alarm rate. To prevent this, 3.85 seconds delay is added before NN starts to evaluate residuals after fault detection alarm.

If one of the NN outputs overshoots 0.5, it is defined as fault isolation signal. Simulation studies showed that some faults belonging to some trajectories may cause two outputs (especially fault outputs of the same actuator) to overshoot 0.5 for short time durations, and these results increase false alarm rate. To regulate this, continuity is accepted as criteria and network output signals with time durations shorter than 0.35 s. are neglected.

With these specifications, simulation studies showed %89.58 fault isolation rate for defined faults and trajectories.

As an illustration of the proposed scheme, a trajectory is defined in (26), a %50 fault at actuator 1 (f_1) at t = 24 s. is simulated. Followed trajectories by joints, errors and applied torques to joints are given in Fig. 10, residuals and analytical redundant signals are given in Fig. 11 and NN outputs are given in Fig. 12 for 4-50 s. time interval:

$$q_{d1}(t) = 0.8 * \cos(t/2)$$
$$q_{d2}(t) = 0.8 * \sin(t/2) \tag{26}$$

It can be seen in Fig. 12 that the outputs of f_1 and f_2 overshoots 0.5 but the time continuity rule neglects this time interval. And fault isolation instant t = 28.21 s. can be seen in Fig. 12.

7.2 Case 2: FDI with the second scheme

As the second case, scheme in Fig. 7 is implemented. Four actuator faults, as in the first case defined in Table 1, are considered and from this consideration, as can be seen in Fig. 7, 1 M-ANFIS for healthy and 4 M-ANFIS for faulty robot models are defined and each is formed of 4 independent ANFIS having instant torque and unit time step delayed position and velocity signals as common 6 inputs and each giving assigned instant position or velocity signal. As a result, 5 M-ANFIS with 20 independent ANFIS is constructed. To train, same 146 different joint trajectories defined in sin- and cos-shape with amplitudes varying between ±1 are simulated for healthy and faulty robot models and 76 sampled datas from these simulations are used.
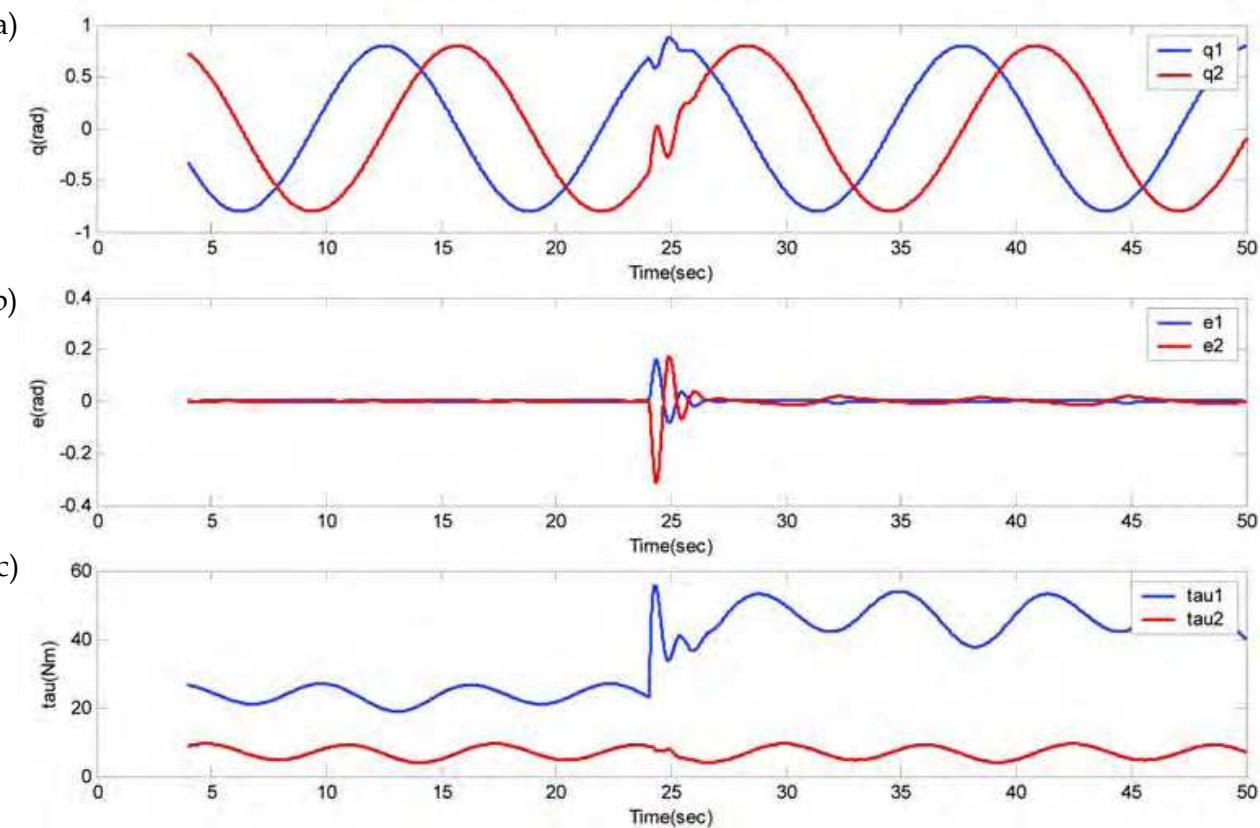
Fig. 10. Time histories of joint 1,2: a) Followed trajectories b) Errors c) Applied torques
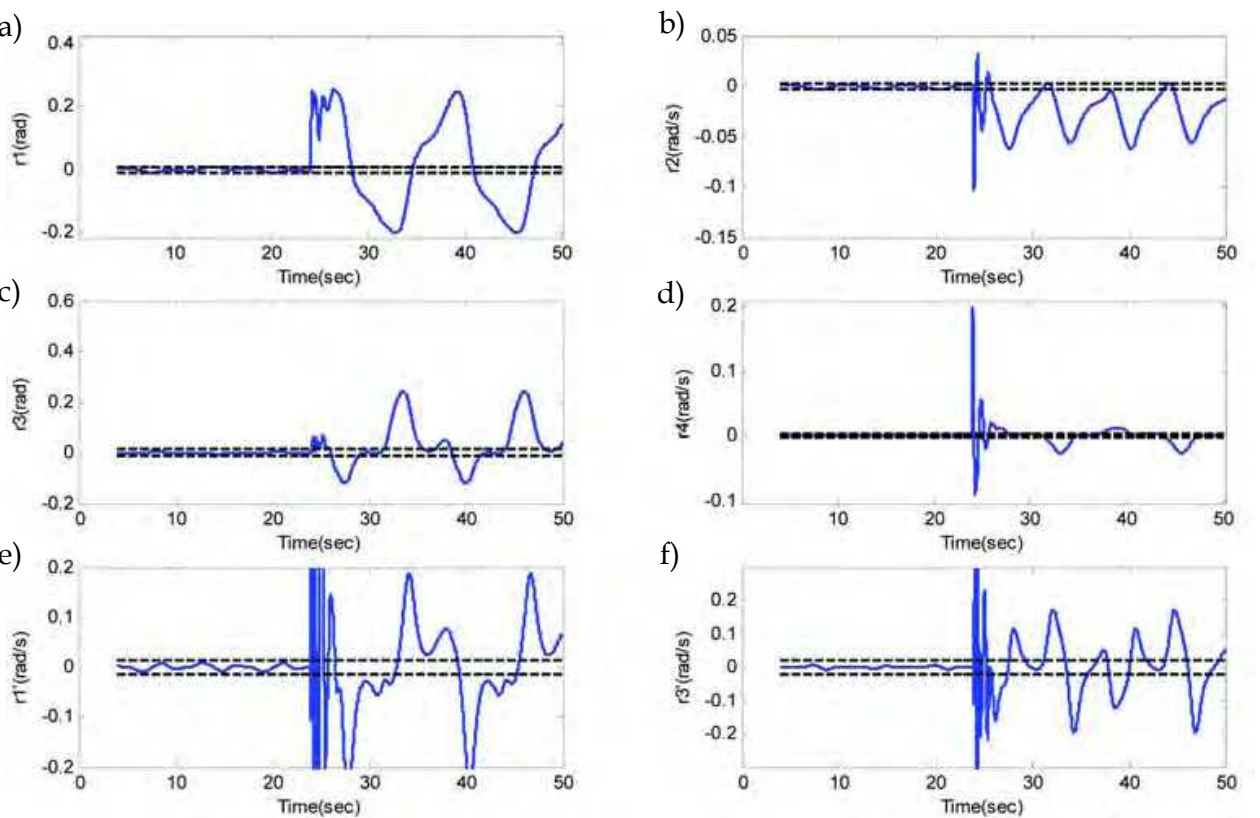

Fig. 11. Residuals and analytical redundant signals with thresholds

a) $r_1 = q_1 - \hat{q}_1$ b) $r_2 = \dot{q}_1 - \dot{\hat{q}}_1$ c) $r_3 = q_2 - \hat{q}_2$ d) $r_4 = \dot{q}_2 - \dot{\hat{q}}_2$ e) $\dot{r}_1 = \dot{q}_1 - \dot{\hat{q}}_1$ f) $\dot{r}_3 = \dot{q}_2 - \dot{\hat{q}}_2$
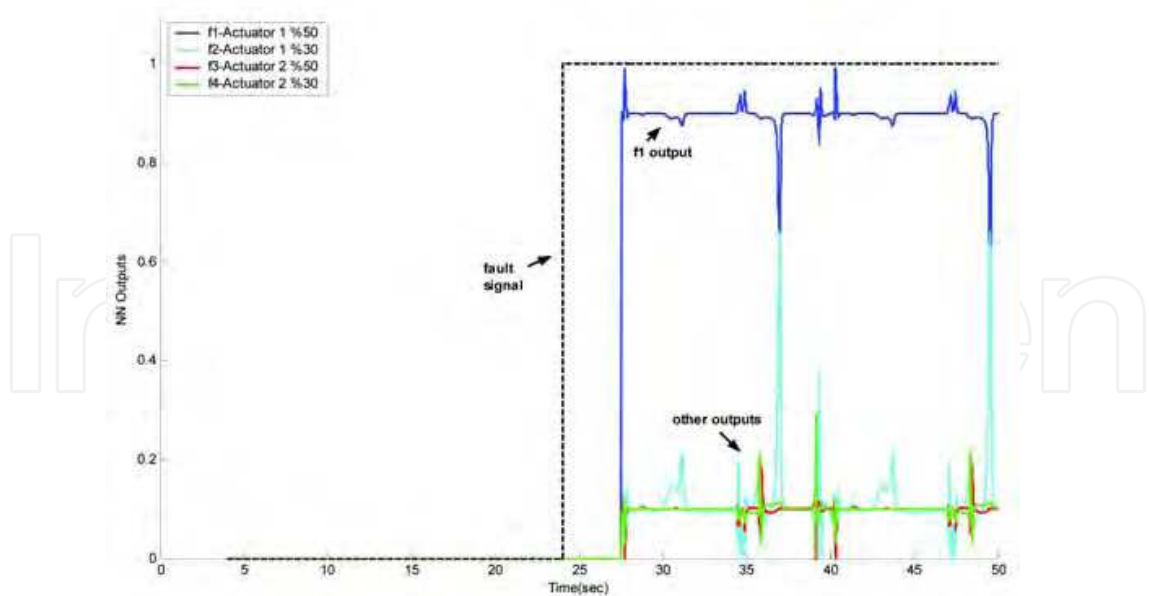
Fig. 12. Neural network outputs

The operation of GOS shows that it can detect and isolate faults at the same time instant. Although Table 2 implies that the residual of the defined fault or healthy condition must be zero, it may have very small values due to modelling errors. To define thresholds for each residual of defined conditions (1 healthy + 4 faulty), 80 trajectories are simulated, up and down thresholds are determined, overshooting of these thresholds for each residual is defined as another condition except belonging to itself. Analytical redundant signals are not defined for this scheme because the residual signals are sufficient and capable of isolating faults using GOS scheme.

Simulation studies for determining fault isolation rate showed that for some faults belonging to some trajectories, two residual sets coming from two models may fall between threshold intervals causing decrease of isolation rate. To increase this rate, residuals are evaluated in the sequence of *healthy-f_1-f_2-f_3-f_4*. Furthermore, again continuity is accepted as criteria and residuals falling between threshold intervals with time durations shorter than 0.8 s. are neglected.

With these specifications, simulation results showed %87.81 fault isolation rate for defined faults and trajectories. As an example of this scheme, the same trajectory in (26) and the same fault f_1 is accepted. Residuals of f_1 are given in Fig. 13 and isolation signals are given in Fig. 14. Fault isolation instant t = 24.93 s. can be seen in Fig. 14.

7.3 Case 3: FDI with the third scheme

As the third case, the scheme in Fig. 8 is implemented. Robot modelling, residual generation and fault detection processes are implemented in the same way with the first scheme. The scheme is simulated for the 80 trajectories used in the second scheme and residual thresholds are determined. To approximate the fault function, M-ANFIS is formed of two independent ANFIS (for each actuator) taking instant torque, position and velocity signals as inputs. Activation of M-ANFIS is realized by the residuals and analytical redundant
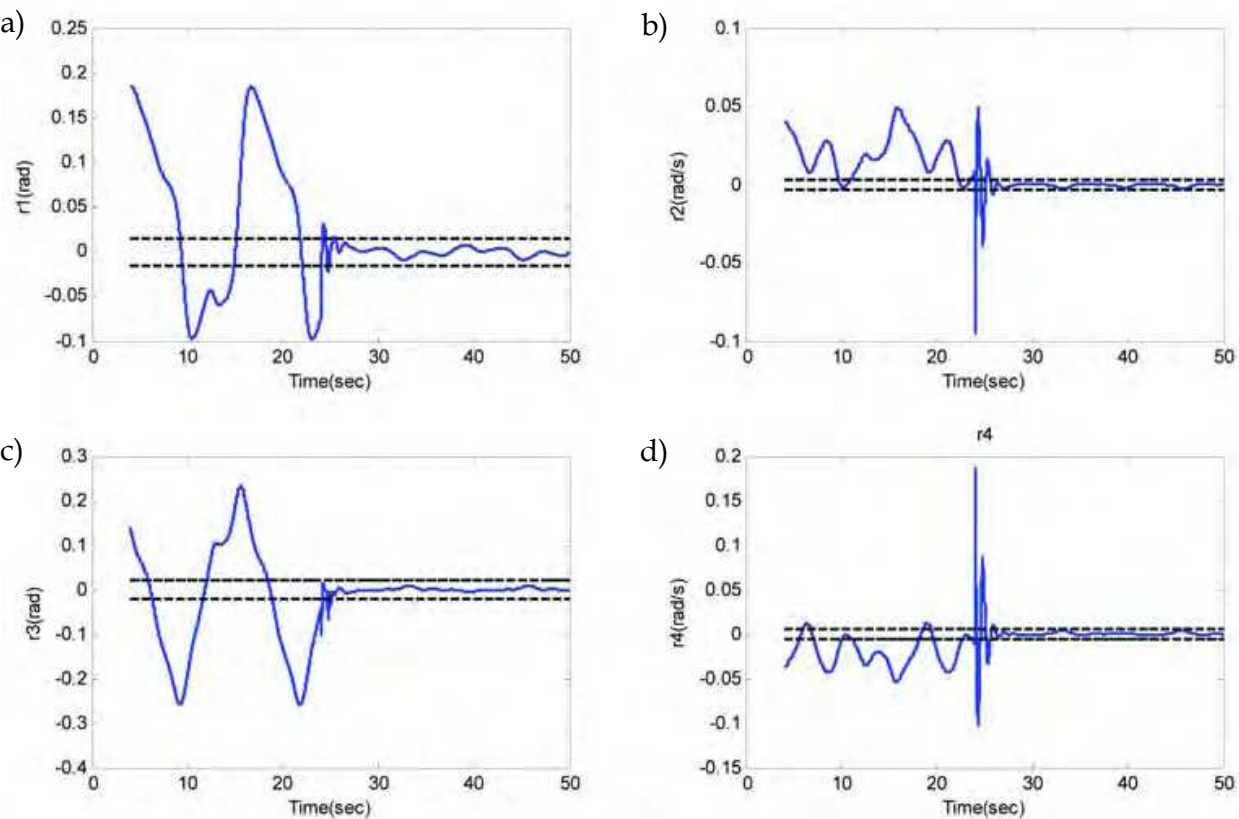
a)

b)

c)

d)

Fig. 13. Residuals of f_1 : a) $r_1 = q_1 - \hat{q}_{1f_1}$ b) $r_2 = \dot{q}_1 - \dot{\hat{q}}_{1f_1}$ c) $r_3 = q_2 - \hat{q}_{2f_1}$ d) $r_4 = \dot{q}_2 - \dot{\hat{q}}_{2f_1}$
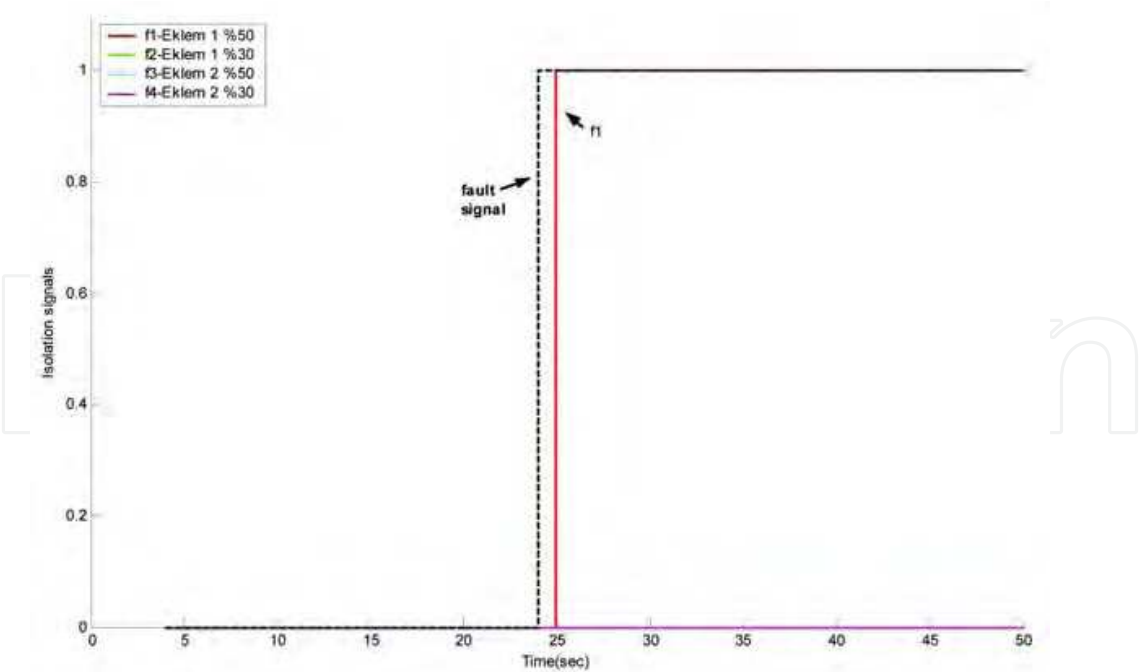
Fig. 14. Fault isolation signals of the second scheme

signals defined for the first scheme. For the training of M-ANFIS, to realize fault function approximation more correctly, 2 extra faults f_5-Actuator 1 %70 torque loss and f_6-Actuator 2 %70 torque loss are defined. Simulations are implemented for defined 6 faults and 101 sampled datas from these simulations are used for training.

Function approximator M-ANFIS is formed of two ANFIS and they are trained to give fault approximation of the faulty actuator. It is hard for the other ANFIS to give zero output because of uncertainties and that causes false alarms. This problem is solved by the rule "The bigger signal is the right signal". This rule is based on that ANFIS output torque of the faulty actuator is mostly bigger than other ANFIS output torques. This rule constitutes false alarms in the cases of low percentage actuator faults resulting close ANFIS outputs.

The second problem is the torque signals containing high frequency components coming from the change of the system region from stable to unstable in the cases of some very high percentage actuator faults. Sudden changes in short time intervals as the high frequency signals lead M-ANFIS to false alarms. These high frequency signals are removed by using a filter like sliding mode control with saturation but the problem still continues and these high frequency components are accepted as a part of the study.

In the simulation studies, like the first scheme, transition from healthy to faulty condition increased false alarms and to decrease these alarms, function approximator M-ANFIS is activated after 3.2 s. delay from fault detection instant.

Again for this scheme, 0.8 s. continuity test is added to the scheme to increase fault isolation rates.

With these specifications, simulation results showed %87.5 fault isolation rate for defined faults and trajectories. Fault function and aproximation signals are given in Fig. 15 and fault isolation signals are given in Fig. 16 for the same example defined for the other schemes. Fault isolation instant $t = 28$ s. can be seen in Fig. 16. In addition to approximating fault function, this approximation can give information if the fault function causes actuator saturation problem.
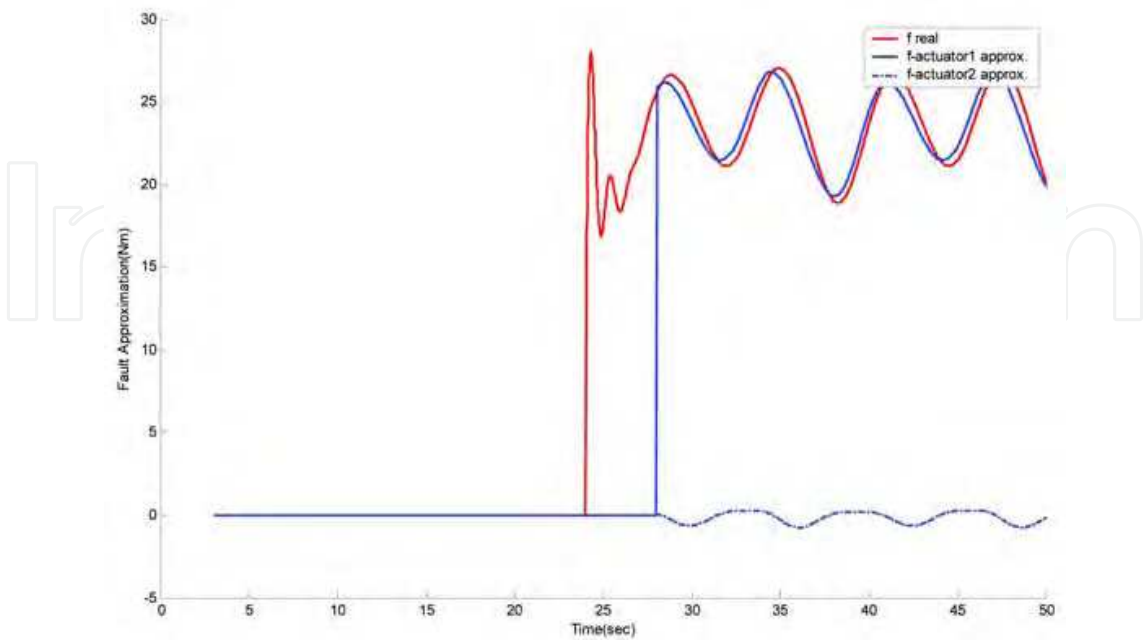

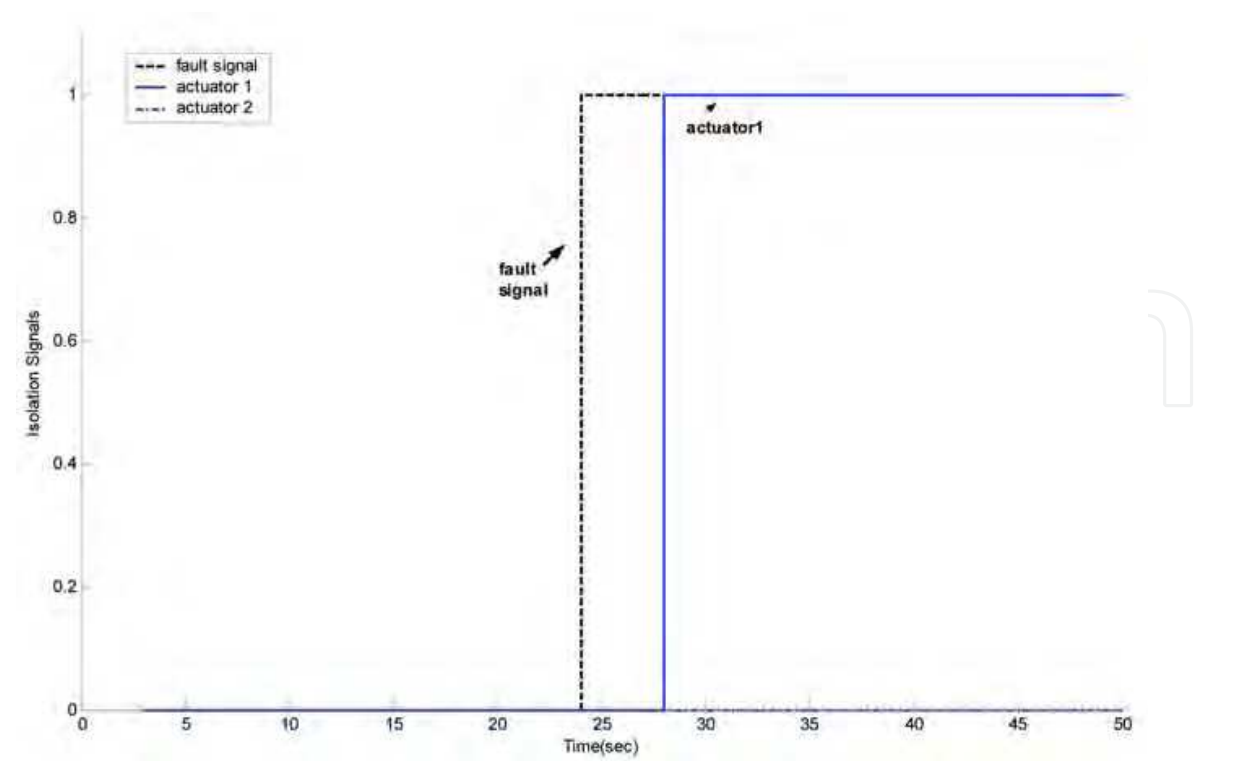Fig. 15. Fault function and approximation signals

Fig. 16. Fault isolation signals of the third scheme

8. Comparison of the schemes and conclusions

In this study, three model-based fault detection and isolation schemes for robot manipulators using soft computing techniques are proposed. All schemes use multi-input-multi-output type of ANFIS, M-ANFIS structure for modelling. The first scheme uses overshooting of the residuals for fault detection and a NN for fault isolation and the second scheme uses generalized observer approach for fault detection and isolation. The third scheme differs from these schemes by approximating fault function and it uses M-ANFIS for approximation. A comparison of the schemes according to some important FDI specifications is given in Table 4.

All schemes have low false alarm rates, common and positive responses against robustness, and adding new trajectories and training. It is hard to define robustness of these schemes analytically but all schemes are tested against different uncertainties added to robot dynamics and all give similar isolation results. It is observed that new added position trajectories with lower values than ±1 peak values show same results. When new trajectories are added, residual thresholds should be calibrated, isolation structure should be trained and isolation delay and continuity time intervals should be revised if the same isolation results are desired.

One of the most demanded specification for FDI schemes is low or absent fault detection and isolation delays. None of the schemes need fault detection delay. The first and the third schemes have fault isolation delays arising from transition from healthy to faulty condition and these delays should be paid attention especially in real-time implementations. The second scheme realizes detection and isolation in the same method and doesn t need any delays.

Resembling of faulty condition residuals to healthy and other type residuals in short time intervals causes low isolation rates. To prevent this, continuity tests are added to all of the schemes.

Scheme	False Alarm Rate	Robustness	Fault Detection Delay	Fault Isolation Delay	Adding New Trajectories& Training
1. Scheme	low	reliable	no	yes	easy
2. Scheme	low	reliable	no	no	easy
3. Scheme	low	reliable	no	yes	easy

	Continuity Test	Computational Load	Fault Function Approximation	Usable For Fault Tolerance	
1. Scheme	yes	low	just defined faults	just defined faults	
2. Scheme	yes	very high	just defined faults	just defined faults	
3. Scheme	yes	low	yes	yes	

Table 4. Comparison of the schemes

The first and the third schemes can be considered to have low level, the second scheme to have very high level computational load (simulation durations as the proof). The main sources of the computational load in these schemes are soft computing structures. Especially, ANFIS and its big brother M-ANFIS bring high computational loads. The second scheme includes M-ANFIS for all models and that makes it hard for real-time implementations.

The main disadvantage of the first and second scheme is that both are functional for just defined faults. This is a limiting characteristic for fault function approximation and for fault tolerant controller (FTC) schemes as an extension of FDI schemes. Fault function approximation is important to give information about how fault function changes and to give the peak value of control signal to observe the saturation occurence. The third scheme can approximate all faults and that makes it the first choice for real-time implementation.

These three schemes, especially the third one, are suitable for real-time implementation. In the future studies, schemes involving fault tolerant controllers will be proposed. These FTC schemes will use gain scheduling nonlinear PID approach and all proposed schemes will be implemented on a real manipulator.

9. References

Abdul, S. & Liu, G. (2008). Decentralised Fault Tolerance and Fault Detection of Modular and Reconfigurable Robots with Joint Torque Sensing, *IEEE International Conference on Robotics and Automation*, pp. 3520-3526

Adjallah, K.; Maquin, D. & Ragot, J. (1994). Non-linear observer-based fault detection, *Third IEEE Conf. on Control Applications*, pp. 1115-1120

Ballé, P. (1998). Fuzzy Model-Based Symptom Generation and Fault Diagnosis for Nonlinear Processes, *IEEE International Conference On Fuzzy Systems*, Vol. 2, pp. 945-950

Brambilla, D.; Capisani, L. M.; Ferrara, A. & Pisu P. (2008). Fault Detection for Robot Manipulators via Second-Order Sliding Modes , *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 11, pp. 3954-3963

Caccavale, F. & Walker, I. D. (1997). Observer-based fault detection for robot manipulators, *IEEE International Conference on Robotics and Automation*, pp. 2881-2887

Cavallaro, J. R. & Walker, I. D. (1994). A survey of NASA and military standards on fault tolerance and reliability applied to robotics, *AIAA*

Chen, J. & Patton, R. J. (1999). *Robust Model-Based Fault Diagnosis For Dynamic Systems*, Kluwer Academic Publishers

Chen, W. & Saif, M. (2008). Output Estimator Based Fault Detection for a Class of Nonlinear Systems with Unknown Inputs, *American Control Conference*, pp. 3307-3312

Chow, E. Y. & Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection system, *IEEE Transactions on Automatic Control*, Vol. 29 No. 7, pp. 603-614

Datta, A.; Mavroidis, C.; Krishnasamy, J. & Hosek, M. (2007). Neural network based fault diagnostics of industrial robots using wavelet multi-resolution analysis, *American Control Conference*, pp: 1858-1863

De Luca, A. & Mattone, R. (2004). An adapt-and-detect actuator FDI for robot manipulators, *IEEE International Conference on Robotics and Automation*, pp. 879-884

Dexter, A. L. (1995). Fuzzy Model-Based Fault Diagnosis, *IEE Proceedings On Cont. Theory and Applications*, Vol. 142, No. 6, pp. 545-550

Dixon, W. E.; Walker, I. D.; Dawson, D. M. & Hartranft, J. P. (2000). Fault detection for robot manipulators with parametric uncertainty: a prediction-error-based approach, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 6, pp. 689-699

Fantuzzi, C.; Secchi, C.; Visioli A. (2003). On The Fault Detection And Isolation Of Industrial Robot Manipulators, *7th International IFAC Symposium on Robot Control*, pp. 399-404

Frank, P. M. & Ding, X. (1997). Survey of robust residual generation and evaluation methods in observer-based fault detection systems, *J. Proc. Cont.*, Vol.7, No.6, pp. 403-424.

Garcia, E. A. & Frank, P. M. (1997). Deterministic nonlinear observer-based Approaches To Fault Diagnosis: A survey, *Control Eng. Practice,* Vol. 5, No. 5, pp. 663-670

Goel, P.; Dedeoglu, G.; Roumeliotis, S. I. & Sukhatme, G. S. (2000). Fault Detection and Identification In A Mobile Robot Using Multiple Model Estimation and Neural Network, *IEEE International Conference on Robotics and Automation*, pp. 2302-2309

Haykin, S. (1999). *Neural Networks: A Compherensive Foundation*, Prentice-Hall, 2. edition

Isermann, R.& Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes, *Control Eng. Practice*, Vol. 5, No. 5, pp. 709-719

Jang, J.R.; Sun, C.T. & Mizutani, E. (1997). *Neuro-fuzzy And Soft Computing*, Prentice-Hall Inc.

Kinnaert, M. (1999) Robust fault detection based on observers for bilinear systems, *Automatica*, Vol. 35, pp. 1829-1842

Lee, I. S.; Kim, J. T.; Lee, J. W.; Lee, D. Y. & Kim K. Y. (2003). Model-based fault detection and isolation method using ART2 neural network, *International Journal Of Intelligent Systems,* Vol. 18, pp: 1087–1100

Leuschen, M. L.; Walker, I. D. & Cavallaro, J. R. (2005). Fault residual generation via nonlinear analytical redundancy, *IEEE Transactions on Control System Technology*, Vol. 13, No. 3, pp. 452-458

Lewis, F. L.; Abdallah, C. T. & Dawson D. M. (1993). *Control Of Robot Manipulators*, MacMillan Publishing

Marcu, T.; Mirea, L. & Frank, P. M. (1998). Neural Observer Schemes For Robust Detection And Isolation Of Process Faults, *UKACC International Conference on CONTROL '98*, pp. 958-963

Moseler, O. & Isermann, R. (2000). Application of model based fault detection to a brushless dc motor, *IEEE Transactions on Industrial Electronics*, Vol. 47, No. 5, pp. 1015-1020

Naughton, J. M.; Chen, Y.C. & Jiang J. (1996). A neural network application to fault diagnosis for robotic manipulator, *IEEE International Conference on Control Applications*, pp. 988-993

Patton, R. J.; Chen, J. & Lopez-Toribio, C. J. (1998). Fuzzy Observers for Nonlinear Dynamic Systems Fault Diagnosis , *37th IEEE Conference on Decision & Control*, pp. 84-89

Patton, R. J.; Frank, P. M. & Clark, R. N. (2000a). *Issues Of Fault Diagnosis For Dynamic Systems*, Springer-Verlag

Patton, R.J.; Uppal, F. J.; Lopez-Toribio, C. J. (2000b). Soft Computing Approaches To Fault Diagnosis For Dynamic Systems: A Survey, *IFAC-SAFEPROCESS*, pp. 298-311

Polycarpou, M. M. & Helmicki, A. J. (1995) Automated Fault Detection and Accommodation: A Learning Systems Approach, *IEEE Transactions On Systems, Man and Cybernetics*, Vol. 25, No. 11, pp. 1447-1458

Polycarpou, M. M. & Trunov, A.B. (2000). Learning Approach to Nonlinear Fault Diagnosis: Detectability Analysis , *IEEE Transactions On Automatic Control*, Vol. 45, No. 4, pp. 806-812

Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm, *IEEE International Conference on Neural Networks*, Vol. 1, pp. 586-591

Schneider, H. & Frank, P. M. (1996). Observer based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation, *IEEE Transactions on Control System Technology*, Vol. 4, No. 3, pp. 274-282

Seliger, R. & Frank, P. M. (1991). Fault Diagnosis By Disturbance Decoupled Nonlinear Observers, *30th Conf. on Decision and Control*, pp. 2248-2253

Terra, M. H. & Tinós, R. (2001). Fault detection and isolation in robotic manipulators via neural networks: a comparison among three architectures for residual analysis, *Journal of Robotic Systems*, Vol. 18, No. 7, pp. 357-374

Tinós R. & Terra M. H. (2002). Free-Swinging and Locked Joint Fault Detection and Isolation In Cooperative Manipulators , *E. Symp. on Artificial Neural Networks*, pp. 513-518

Vemuri, A. T. & Polycarpou, M. M. (1997). Neural-network-based robust fault diagnosis in robotic systems, *IEEE Transactions on Neural Networks*, Vol. 8, No. 6, pp. 1410-1420

Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S.N. & Yin K. (2003). A review of process fault detection and diagnosis part III: process history based methods, *Computers and Chemical Engineering*, Vol.27, pp. 327-346

Yang, H. & Saif, M. (1995). State Observation, Failure Detection And Isolation(FDI) In Bilinear Systems, *34th Conference on Decision & Control*, pp. 2391-2396

Zhang, X.; Polycarpou, M. M. & Parisini, T. (2002). A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems, *IEEE Transactions On Automatic Control*, Vol. 47, No. 4, pp. 576-592

Fault Detection

Edited by Wei Zhang

In this book, a number of innovative fault diagnosis algorithms in recently years are introduced. These methods can detect failures of various types of system effectively, and with a relatively high significance.

How to reference

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

INTECH

open science | open minds