ROBUST FAULT DIAGNOSIS IN ROBOTIC MANIPULATORS

N. Kesav Kumar (M.TECH)
Electronics and communication engineering
NBKRIST, Vidya nagar,
Nellore, India.
kumarkeshavn@gmail.com

Smt. R. Lavanya
Assistant professor
Electronics and communication Engineering
NBKRIST, Vidya nagar,
Nellore, India.

Abstract— In this paper the artificial neural networks are used for both residual generation and residual analysis. A Multilayer Perceptron (MLP) is employed to reproduce the dynamics of the robotic manipulator. Its outputs are compared with actual position and velocity measurements, generating the so-called residual vector. The residuals, when properly analyzed, provide an indication of the status of the robot (normal or faulty operation). The ANN architecture employed in the residual analysis is also a multilayer perceptron (MLP) or a radial basis function network (RBFN) which uses the residuals of position and velocity to perform fault identification. Simulations employing a selective compliance assembly rotate arm (SCARA) robotic manipulator are showed demonstrating that the system can detect and isolate correctly faults that can occur during the performance of its task. We opted in our study on fault diagnosis for a dual neural classification. Thus, the architecture of the proposed approach is based on two types of classifiers: Firstly a classifier consisting only of one neural network (MLP or RBF) followed by a comparison of the results of detection and localization. Secondly a classifier consisting of two neural networks (RBF and MLP) and is followed by a final decision system**.**

Keywords— MLP, RBF, RBFN, ANN, FDI, SCARA, CBR .

I. INTRODUCTION

A system can be fault tolerant if it is reconfigurable, case which FDI is essential. A number of studies have been dedicated to the assessment and analysis [I-II] of robot reliability. Other studies related to enhancing a robot's tolerance to failure include work on layered failure tolerance control, failure tolerance by trajectory planning kinematic failure recovery and manipulators specifically designed for fault tolerance. Being able to identify the extent of fault-tolerance in a system would be a useful analysis tool for the designer [III-IV]. In recent years neural networks have been applied to variety of problems in the areas of pattern recognition, signal processing, image processing, process identification, etc.

Fault diagnosis and isolation methods are usually based on the residual generation and analysis concept [V-VI]. A mathematical model is used to reproduce the dynamic behavior of the fault-free system; the deviation of the output predicted by the model from actual output measurements forms the so-called residuals, which, when properly analyzed, provides valuable information about failures. In this paper, two artificial neural networks are employed to identify and isolate the faults. A learning architecture, approximation of dynamic behavior of robot manipulator, is used to generate the residual vector, by comparing with actual measured values. The ANN outputs are compared with the measured system outputs and, thus, generate the residual vector. In this paper, two ANN are utilized: a multilayer perceptron (MLP) is employed to reproduce the manipulator dynamic behaviour and a second MLP or a radial basis function network (RBFN) is used to classify the residual vector.

The ANN used in this paper is describe in section II. The robotic manipulator system is described in section III together with some methods for FDI in this system. the multilayer perceptron (MLP), trained with the classical back-propagation algorithm, is employed to reproduce the manipulator dynamic behaviour and the radial basis function network (RBFN), initialized with the classical Kohonen self-organizing map [7], is used to classify the residual vector.

The second proposed approach with two neural networks classifier (MLP and RBF), followed by a decision system is described in section IV.

Manipulator simulation results using two ANN training procedures are given in section V and, finally, the conclusions are in section VI.

II. ARTIFICIAL NEURAL NETWORKS

In this paper, an MLP with back-propagation algorithm is used to reproduce the behavior of nonlinear dynamical systems and a second MLP is used to residual classification. For a p-dimensional input vector and a *q*-dimensional output vector, the MLP input/output relationship defines a mapping from a *p*-dimensional Euclidean space to a q-dimensional Euclidean output space. Using only one hidden layer, presenting in the n-th sample(where (

n=1,2,.....n_p), the input vector X(n)=[x_1(n) x_2(n).....x_p(n)]^T the activation of the output neuron *k* (where k = 1, 2,..., q) is:

$$O_k(n) = \varphi_k\left[\sum_{j=0}^{m} W_{jk}(n)\varphi_j\left[\sum_{i=0}^{p}\left(W_{ji}(n)X_i(n)\right)\right]\right] \quad (1)$$

where m is the number of neurons in the hidden layer, W_{jk} is the weight between the j-th neuron of the hidden layer and the k-th neuron of the output layer, W_{jk} is the weight between the i-th neuron of the input layer and the j-th neuron of the hidden layer, φ_k is the non linear activation function of the output layer and φ_j is the nonlinear activation function of the hidden layer. In this article, the weights of the MLP are trained by the well known back-propagation algorithm.

The RBFN employed in this article has three layers. There are no weights linking the first and the hidden layers. The hidden layer has neurons with radial activation functions. Each neuron *j* in the hidden layer (called radial unit *j*) is responsible for the creation of a receptive field in the *p*-dimensional input space. The receptive field of each radial unit is centered in a *p*-dimensional vector μ_j, called radial unit center. Therefore, the radial unit *j* activates according to the vector distance between the input vector and the radial unit center. There are weights between the hidden and the output layers, and the activation in the last layer is linear. Presenting in the *n*-th sample the input vector X(n)=[x_1(n) x_2(n).....x_p(n)]^T, the activation of the *k*-th output neuron (*k* = 1, 2,..., *q*) is given by μ_j

$$O_k(n) = \sum_{j=0}^{m} W_{jk}(n)h_j(n) \quad (2)$$

where *m* is the number of radial units, W_{kj} is the weight between the *j*-th radial unit and the *k*-th output neuron, and *hj* is the activation of the *j*-th radial unit. In this work, the Cauchy radial function is employed as activation function in the radial units:

$$h_j(n) = \frac{1}{1} + \left\|R^{-1}\left(X(n) - \mu_j\right)\right\|^2 \quad (3)$$

where *R* is a diagonal matrix formed by the individual parameters that define the receptive field size in each dimension of the input space.

III. MANIPULATOR FDI WITH ANN

We begin with a presentation of residual generation for a generic dynamic system, and then specialize it for the case of a robotic manipulator. In the sequence we discuss the fault isolation criterion we use and the failures that the system is able to cope.

A. *Residual Generation*

In discrete time, the state equation of a fault-free nonlinear dynamic system is given by

$$x(t + \Delta t) - g_i(x(t), u(t)) \quad (4)$$

where **x**(t) is the state vector at time *t*, **u**(t) is the applied control vector, Δt is the sample rate and **f** (.) is the vector-valued nonlinear function of the fault-free system. Considering now that a fault *i* occurs, the dynamics of the system are modified to

$$x(t + \Delta t) = f(x(t), u(t)) \quad (5)$$

where g*i*(.) is the vector-valued nonlinear function of the system affected by fault *i.* The faults may or may not be additive inputs (i.e., dependent only on the time variable). The i-th fault vector can be defined as the difference between the faulty system dynamics (Eq. 5) and the fault-free system dynamics (Eq. 4):

$$\varphi_i(t + \Delta t) = g_i\left(x(t), u(t)\right) - f(x(t), u(t)) \quad (6)$$

Obviously, for the fault-free system $\varphi_i(t + \Delta t)$.

Generally, for each possible fault i, the fault vector has a particular behavior, called the fault signature. For the identification of the fault type, the fault vector must be computed and analyzed. Therefore, the dynamic behavior of the fault-free system (Eq. 4) must be estimated (for example, by the mathematical model or by an ANN). Then, when fault i occurs, the residual vector can be computed as:

$$\bar{\varphi}_i(t + \Delta t) = x(t + \Delta t) - \hat{x}(t + \Delta t) = g_i(x(t), u(t)) - \hat{f}(x(t), u(t)) = \varphi_i(t) + e_i\left(x(t)\right), u(t)) \quad (7)$$

where is a vector that represents the input-output mapping of the estimated fault-free dynamic behavior of the system and **ei** is the error between the actual fault-free behavior and the estimated one. In real systems and fault free case, the error is due to external disturbances, unmodeled system uncertainty, mapping errors (or modeling errors in model-based systems), and measurement noise. In this work, an MLP is employed to reproduce (estimate) the fault-free dynamic behavior of a robotic manipulator. Taking as inputs the control signals **u** and the states **x** measured at *t*, the MLP reproduces the states **x** of the fault-free system measured at $t + \Delta t$ (Eq. 7).

B. *Residual Generation in Mechanical Manipulators*

As mentioned above, an MLP is used to approximate the state equation (dynamic function) of the fault-free manipulator. The dynamic of a fault-free robotic manipulator with actuators in each joint is given by:

$$x = \begin{bmatrix}\dot{\theta}(t)\\ \ddot{\theta}(t)\end{bmatrix} = M(\theta(t))^{-1}\begin{bmatrix}\dot{\theta}(t)\\ \tau(t) + \tau_d - C\left(\theta(t), \dot{\theta}(t)\right) - \\ F\left(\theta(t), \dot{\theta}(t), t\right) - G(\theta(t\theta))\end{bmatrix} \quad (8)$$

where θ is the vector of joint angular positions, τ is the vector of joint torques, M is the inertia matrix, t is the time index, C is the vector of Coriolis and centrifugal forces, G is the vector of gravitational torques, F is the vector of frictional torques and other nonlinearities, and is the vector of external uncorrelated disturbances.

C. Residual Analysis

In this work residual analysis for fault isolation purposes is also performed with ANNs, an MLP utilizing the velocity residuals. A general schema is shown in the figure 1 below.

Outputs 1 through q-1 correspond to the q-1 possible failure modes, while output q corresponds to fault-free operation. The ANN's output i (i = 1,..., q-1) is trained to present a '1' in case fault i occurs and '0' otherwise. The output q is trained to present a '1' in case fault-free operation and '0' otherwise.

In the FDI procedure, the normalized positions and velocities and the torques applied at *t* are presented to the MLP, which estimates the positions and velocities at t+Δt; the MLP outputs are compared with the normalized positions and velocities measured at t+Δt to generate the residuals; the velocity residuals are presented to the second ANN (MLP or RBFN) which classifies the residuals and generates a vector that, when analyzed under the fault criterion, indicates the operation status of the system

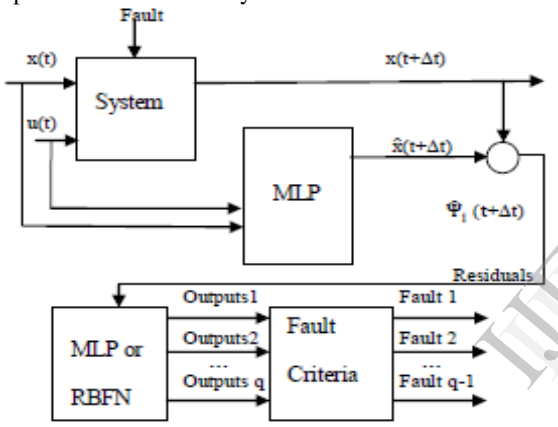

Figure 1.Residual anaylsis employing different architecture.

D. Fault Indication Criterion

Generally, the fault is signaled when a threshold is crossed. However, as the faults and the mapping errors are generally correlated with the system dynamics, a small threshold can be a source of false alarms, while a large one can hide the fault effects. The fault isolation scheme presented above will indicate the fault information in the ANNs outputs. It is known, however, that false alarms may occur, and one cannot rely on isolated information to decide on whether a failure actually occurred. In this work we adopt the following fault criterion: a fault is said to have occurred whenever one of the ANN outputs is greater than the other ones for *g* consecutive time steps; where *g* is achieved by trial and error (searching for a good compromise between false alarm rate and detection delays).

Mathematically we have:

if $\Phi_i = max_{j=1}^{q}(\Phi_i)$ during consecutive samples :

fault i=1 (9)

otherwise : *fault i*=0

where *i* is the output *i* (*i* = 1, ..., *q*-1) of the ANN (recall that output *q* refers to fault-free system operation). In this work we consider one possible joint failure type. In free-swinging joint faults, a loss of torque occurs on a joint. This fault can be caused, for example, by a loss of electric power on an electric actuator, or by a mechanical fault in a drive.

Other faults can be taken into consideration: locked joint faults occur when there is an immobilization of a joint in a fixed position. This fault may be due, for example, to a very high gear ratio on an actuator that has lost power, or due to Indeed, opting for multiple classification , one of the usual solutions is to choose the classification model giving the best result. In our work, we used both the redundancy and complementarily of classification models used.

IV. DUAL NERUAL CLASSIFICATION

The second proposed approach consists of two neural networks (MLP and RBF), followed by a decision system (Figure 2).

There are several methods for decision making (Analogy, Fuzzy Logic, statistical treatment, etc.).

We opted for the analogy approach. This method is more natural and nearest to the human reasoning.

The principle of this method is to draw the decisions taken in the past in similar situations to solve new problems. The technique of this method is the reasoning from cases (examples) or" Case-Based Reasoning (CBR)." This technique is based on the assumption that the decisions and the resolution of a problem is the access to information stored in previous experiments to further exploitation.

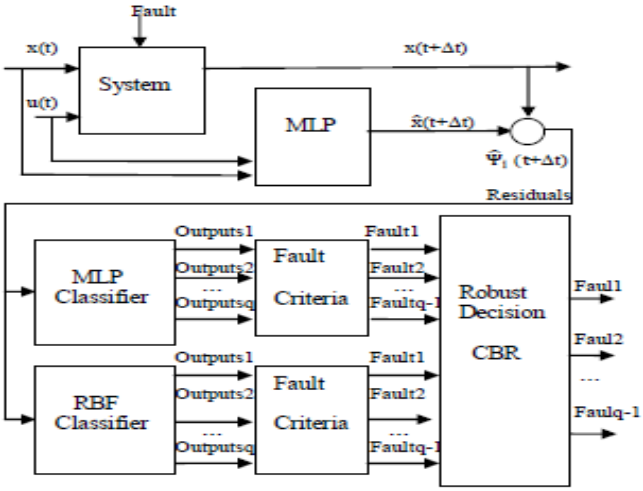

Figure 2. Dual classification with robust decision

V. SIMULATIONS AND RESULTS

To evaluate our ANN-based FDI method we performed an extensive simulation study with a three-link SCARA robot. The dynamic equations, which can be derived via the Euler-Lagrangian method, are represented as follows:

$$\begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} l_1 l_2 \sin(q_2)$$

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3 g \end{bmatrix} = \tau(t)$$

where:

$$D_{11} = I_1^2 \left(\frac{m_1}{3} + m_2 + m_3\right) + l_1 l_2 (m_2 + 2m_3)\cos(q_2) + l_2^2 \left(\frac{m_2}{3} + m_3\right)$$

$$D_{12} = D_{23} = D_{31} = D_{32} = 0$$

$$D_{12} = -l_1 l_2 \left(\frac{m_2}{2} + m_3\right)\cos(q_2) - \left(\frac{m_2}{2} + m_3\right) = D_{21}$$

$$D_{22} = l_2^2 \left(\frac{m_2}{3} + m_3\right), D_{33} = m_3$$

$$C_{11} = -\dot{q}_2 (m_2 + 2m_3), C_{12} = -\dot{q}_2 \left(\frac{m_2}{3} + m_3\right)$$

$$C_{13} = C_{22} = C_{23} = C_{31} = C_{32} = C_{33} = 0$$

In which q1,q2 and q3 are the angles of joints 1, 2 and 3; and are the mass of links m1, m2 and m3; and are the length of links l1, l2 and l3; *g* is the gravity acceleration. This simulation study demonstrates that the presented scheme is effective when applied to a real life robotic system. The simulation was conducted using Matlab & Simulink. In this study we considered free-swinging joint faults, as described in Section 3.

A. *Residual generation results*

The MLP used to reproduce the manipulator dynamic behavior has one hidden layer with 49 neurons. The input layer has 9 neurons (3 joint positions, 3 joint velocities, and 3 joint torques measured at (t+ Δt) and the output layer has 6 neurons (joint positions and joint velocities at (t+Δt)). The training set is formed from 2500 patterns obtained by simulating more than 90 different trajectories with 25 samples each (at a sample rate of 0.073 s).

Two test sets are used to validate the residual generation. The first set has 5000 patterns obtained in the simulation of 200 random trajectories. The second set has 5000 patterns with measurement noise (with variance = 0.01) added to the positions and velocities.

B. *Residual analysis results: free-swinging joint faults*

Here, we considered a free-swinging joint fault in the robot dynamics as described in Section 3. The MLP and the RBFN are trained with 900 patterns, obtained with the simulation of 90 trajectories with 10 samples each as shown in Table 1.

Table 1. Trajectories of the RBFN Training set

Trajectories	Operation
1-20	Fault in joint 1 (fault 1)
21-40	Fault in joint 2 (fault 2)
41-75	Fault in joint 3 (fault 3)
76-90	No fault

The faults are set to occur in the beginning of the trajectories, before the manipulator reaches its set-point. Two test sets were used to evaluate the architecture. The first set has 5000 patterns obtained by simulating of 250 random trajectories with 10 samples each. The second set has 5000 patterns with measurement noise (with variance = 0.01) added to the positions and velocities. The Table 2 presents the mean squared errors for the architecture.

Table 2. Mean Squared Errors (MSE) of the MLP and RBFN trained in Classifying the Residual

	MLP	RBFN
Training (900 patterns)	2.1×10^{-3}	2.1×10^{-2}
Test 1 (5000 patterns – without noise)	1.9×10^{-2}	3.9×10^{-2}
Test 2 (5000 patterns – with noise)	2.8×10^{-2}	1.5×10^{-1}

To simulate the occurrence of free-swinging joint faults, we set the torque applied at a joint to zero.
In first time, the residual analysis for fault isolation purposes is performed with an MLP utilizing the velocity residuals.

The figures 3, 4 and 5 shows the residual, position and velocity of the joint 2 with a free-swinging joint faults occurring at *t* = 4s.

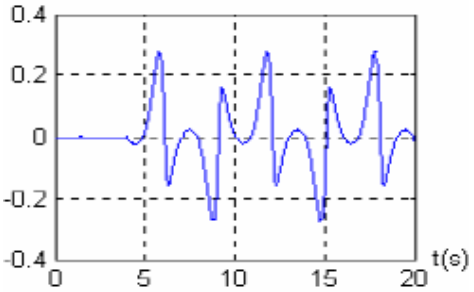

Figure 3. *Residual with a free-swinging joint faults in joint 2 at t=4 s.*

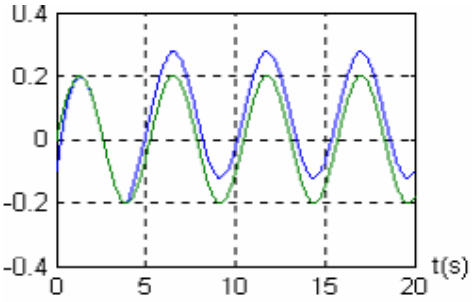

Figure 4. *Normalized position and respective MLP output in a trajectory with fault in joint 2 at t=4 s.*

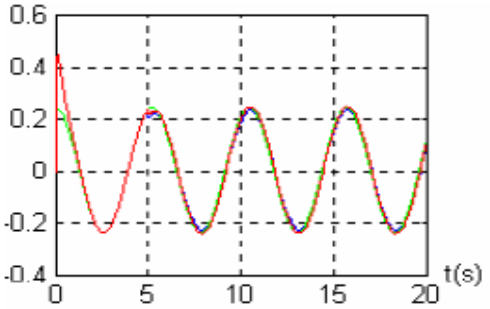

Figure 5. *Normalized velocity and respective MLP output in a trajectory with fault in joint 2 at t=4 s.*

In second time, the residual analysis for fault isolation purposes is performed with an RBFN utilizing the velocity residuals.

The figures 6, 7 and 8 shows the residual, position and velocity of the joint 2 with a free-swinging joint faults occurring at t = 5s.

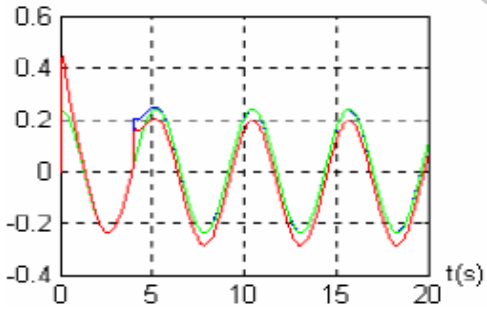

Figure 6. *Residual with afree-swwinging joint faults in foint 2 at t=5s.*

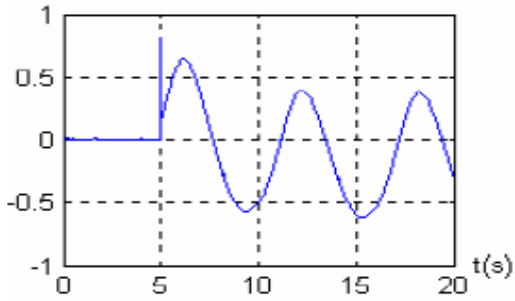

Figure 7. *Normalized velocity and respective RBFN output in a trajectory with fault in joint 2 at t=5 s.*

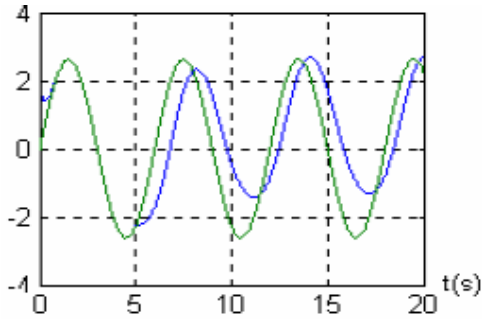

Figure 8. *Normalized velocity and respective RBFN output in a trajectory with fault in joint 2 at t=5 s.*

C. ANALYSIS OF THE RESULTS

The ANN-based FDI system presents very satisfactory results when applied in a simulation of the Scara manipulator for consider free-swinging joint faults. It can detect and isolate faults that occur in trajectories that do not belong to the training set. The MLP reproduces the dynamic behavior of the fault-free robot with a small residual signal. It is important to remember that the quality of the MLP training (residual generation) is very important for the performance of the FDI system. The comparison of different ANN architectures for residual analysis led us to conclude that the RBFN outperforms the MLP for the task of residual analysis. This occurs because the classification made by the RBFN is more robust than that by the MLP. Recall that the RBFN classifies the patterns according to the distance between them and the radial unit centers, while the MLP classifies the patterns according to decision surfaces that are placed according to the training patterns. Another advantage of the RBFN is that the time spent for training it is smaller than the time spent for training the MLP. One disadvantage is that the time for running the RBFN is greater than the time for running the MLP because the number of radial units (and the time of processing) in the former are generally greater than the number of hidden units in the latter. It is important to emphasize that the results presented are somewhat dependent on the choice of the parameters in each training procedure. Also, fault isolation is difficult to perform when different faults occupy the same region in the residual space.

Indeed, the comparison between the results obtained with the MLP network may be different from those of RBF network, which makes diagnosis difficult to achieve.

While with the second approach involving two neural networks (MLP and RBF) followed by a decision system, the diagnosis becomes more robust.

VII. CONCLUSIONS

The Fault Diagnosis System proposed here presents good results when it is applied to a three-link SCARA robot.

With an extensive simulation study results, we were able to conclude that ANNs are a powerful means for Fault Diagnosis System tasks, robustly performing both residual generation and residual analysis.

Research work developed in this paper deals with decision support systems for fault diagnosis and decision-making based on Artificial Intelligence using hybrid techniques, and soft computing implying neural networks and Case-Based Reasoning (CBR).

Further work on this is to extend the Fault Diagnosis System scheme to robotic manipulators with a larger number of degrees of freedom.

Other different types of faults can be detected and isolated using other methods: Fault detection and isolation of robotic manipulator via fuzzy logic, neuro-fuzzy control, hybrid system and expert system [8-12]. The work presented here can be expanded to include post failure control of the robotic manipulator in a hybrid system framework.

REFERENCES

[1] C. Fantuzzi, C. Secchi, and A. Visioli, "On the fault detection and isolation of industrial robot manipulator", In 7th International IFAC Symposium on Robot Control, 2003.

[2] A. T. Vemuri, M. M. Polycarpou, and S. A. Diakourtis, "Neural network based fault detection in robotic manipulators", IEEE Transactions on Robotics and Automation, 14(2), 1998, pp. 342-348.

[3] Y. Yi, J. E. Mcinroy, and Y. Chen, "Fault tolerance of parallel manipulators using task space and kinematic redundancy", IEEE Transactions and Robotics, vol. 22, no. 5, 1017-1021, 2006.

[4] W. Chen, and M. Saif, "Unknown input observer design for a class of nonlinear systems: An LMI approach", Proc. of the American Control Conference ACC'06, USA , 2006.

[5] M. H. Terra, and R. Tinos, ""Fault detection and isolation in robotic manipulators via neural networks: A comparison among three architectures for residual analysis", Journal of Robotic Systems, 18(7), 57-374, 2001.

[6] M. H. Terra, and R. Tinos, "Fault detection and isolation for robotic systems using a multilayer perceptron and a radial basis function network", IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2, pp. 1880-1885, 1998.

[7] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin, 1995.

[8] D. Fragkoulis, G. Roux, and B. Dahhou, "Application of a model based fault isolation method for nonlinear dynamic systems", International Conference on Prognostic and Health Management, USA, 2007.

[9] R. Merzouki, K. Fawaz, and B. Ould Bouamama, "Hybrid Fault Diagnosis for Telerobotics System", Mechatronics, vol. 20, 729-738, 2010.

[10] A. F. T. Winfield, J. Nembrini, and Y. Chen, "Fault tolerance in robot swarms", International Journal of Modeling, Identification and Control, vol. 1, No. 1, 30-37, 2006.

[11] J. Gertler, Proc. Of the IFAC Simp. "On Fault Detection, Supervision and Safety for Technical Processes", Kingston Upon Hull, U. K., vol. 1, pp. 133, 1997.

[12] I. Eski, S. Erkaya, S. Savas, and S. Yildrim, "Fault detection on robot manipulators using artificial neural networks", Robotics and Computer-Integrated Manufacturing, vol. 27, Issue 1, pp. 115-123, 2011.