FAULT TOLERANT CONTROL IN ROBOTIC MANIPULATORS VIA NEURAL NETWORKS

Mohamed Salah KHIREDDINE
Electronics Department, Batna University, Algeria
mkhireddine@yahoo.fr

Saloua OUARHLENT
Electronics Department, Batna University, Algeria
ouarhlentsaloua@hotmail.fr

Abstract: *In this paper an algorithms is developed for fault diagnosis and fault tolerant control strategy for nonlinear systems subjected to an unknown time-varying fault in the presence of modeling error. Faults in an industrial process could be timely detected and diagnosed in many cases. It is possible to subsequently reconfigure the control system so that it can safely continue its operation (possibly with degraded performance) until the time comes when it can be switched off for maintenance. In order to minimize the chances for drastic events such as a complete failure, safety-critical systems must possess the properties of increased reliability and safety. Faults in robotic systems are inevitable. They have diverse characteristics, magnitudes and origins, from the familiar viscous friction to Coulomb/Sticktion friction, and from structural vibrations. This paper presents an on-line environmental fault detection and an accommodation scheme. The performance of the proposed scheme is applied to a SCARA robot to show the effectiveness of the proposed approach.*

Keywords: *Fault Diagnosis, Robotics, SCARA Robot Arm, Fault Detection, Fault approximation, Fault tolerant control.*

1. Introduction

Recently, fault-tolerant control has gained increasing attention in the context of chemical process control; however, the available results are mostly based on the assumption of a linear process description [11,12] and do not account for complexities such as control constraints or the unavailability of state measurements. In process control, given the complex dynamics of chemical processes (example, nonlinearities, uncertainties and constraints) the success of any fault-tolerant control method requires an integrated approach that brings together several essential elements, including: (1) the design of advanced feedback control algorithms that handle complex dynamics effectively, (2) the quick detection of faults, and (3) the design of supervisory switching schemes that orchestrate the transition from the failed control configuration to available well-functioning fallback configurations to ensure fault-tolerance.

Diagnosis and supervision are important in many applications. Different approaches for fault detection using mathematical models have been developed in the last 20 years. The task consists of the detection of faults in the processes, actuators and sensors by using the dependencies between different measurable signals. This consists of comparing the behaviour of the real system and the behaviour of a model of the system. In an ideal case, the system and the model behave exactly the same and a fault is detected when the behaviours are different, but usually there are differences between the behaviours of the system and the model.

The effectiveness of the proposed approach is verified by the development of the FTC scheme for a SCARA robot. Results of this extensive numerical study are included to verify the applicability of the proposed scheme.

During the detection stage, faults are monitored and detected using a detection/approximation observer, which is robust with respect to unmodeled dynamics.

The detection/approximation observer is also used to approximate changes whose dynamics are not found to be equivalent to any a-priori known change scenarios. The dynamics of the fault can be approximated using on-line approximation techniques, which include: *multi-layer neural networks*, *polynomials*, *rational functions*, *spline functions*, *radial-basis-function (RBF) networks*, *adaptive fuzzy systems*, etc... [1][2]. From the past experience, RBF networks performed very well in robotic applications. For this reason, they are employed in this paper for approximation purposes [3][2]. Section 2 describes the dynamic model of the robotic system and of the faults. The general framework of the proposed scheme is studied in section 3. This section thoroughly investigates every stage of FTC. Simulation studies are presented in section 4. Faults can be separated into two distinct categories: those that change the nonlinear dynamics of the nominal model, and those that do not. The second category depends only on time, and not on the states or the inputs, and therefore can be modelled as *additive*.

There are very effective techniques that can accommodate such faults, which include robust control and adaptive control. Faults which belong to the first category have nonlinear dynamics and are have superior capabilities than conventional techniques. They are more difficult to handle because they depend both on the system's states and the input control signals. The purpose of our research work is to design a very effective method that specifically deals with the system state and input dependent faults, while being robust with respect to the unmodeled dynamics.

2. Dynamic models

This section presents the well-studied dynamic structure of the robotic system. The second part of this section, concentrates on the dynamical structure, configuration and nomenclature of the changes (faults) in the robotic system. Innovations like parametric change history profiling and decoupled torque-dependent and state-dependent change model are introduced and thoroughly analyzed.

2-1. Robotic system

The dynamic motion of the robot arm in a robotic system is produced by the torques generated by the actuators. This relationship between the input torques and the time rates of change of the robot arm components configurations, represent the dynamic model of the robotic system [3].

The dynamic model of the robotic system can be derived using either Lagrangian, or Newton-Euler methods [3]. Both methods lead to the identical system of differential equations, which have been extensively studied in the literature on robots [3] [5] [6]. A general healthy *n*-degree of freedom robotic system is described by the following system of differential equations:

$$M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + \mu(\theta,\dot{\theta},\tau,t) = \tau \quad (1)$$

where $\theta, \dot{\theta}, \ddot{\theta} \epsilon R^n$ denote the vectors of joint positions, velocities, and accelerations, respectively, $\tau \epsilon R^n$ is the vector of input torques, $G(\theta) \epsilon R^n$ is the vector of gravitational torque, $V(\theta,\dot{\theta}) \epsilon R^n$ is the vector representing Coriolis and centripetal forces, $M(\theta) \epsilon R^{n\times n}$ is the inertia matrix whose inverse exists, and $\mu(\theta,\dot{\theta},\tau,t) \epsilon R^n$ denotes the unmodeled dynamics. It is assumed that the unmodeled dynamics are bounded.

2-2. Faults

There are faults, which are referred to as drastic. They affect the system in such a way that it cannot function any further, and any ordinary control techniques cannot counteract their effects. An example of component catastrophic fault is a break of a joint or a link section. An example of actuator catastrophic fault is a short circuit in electric motor, permanently damaging the wiring. This type of faults is the worse case fault scenario and its effects on the system are obviously devastating. The only way they can be corrected is by direct operator (human) involvement and replacement of the system components. This paper concentrates only on the faults of smaller magnitudes, or *non-drastic*, which can be accommodated with ordinary control techniques. This type of faults includes different variations of friction, misbalances in the joint or actuator, the interaction with the external, etc… These faults can significantly affects the system's performance as well, which can be expressed in the loss of productivity, reduced life expectancy of the system, and unsafe environment for people and the external environment.

Thus, the presence of faults (non- catastrophic) is *both* state and time dependent, and their presence and magnitude is affected by a number of parameters. A general representation of the fault dynamics is taken to be:

$$F(\theta,\dot{\theta},\tau,t) = \beta(P-p)f(\theta,\dot{\theta},\tau) \quad (2)$$

Where: $f(\theta,\dot{\theta},\tau) \epsilon R^n$:Denotes the fault dynamics, and $\beta(P-p) \epsilon R^{n\times n}$: represents the state and/or time dependent *fault profile* that has the following structure:

$$F(\theta,\dot{\theta},\tau,t) = diag[\beta_1(P_1-p_1), \beta_2(P_2 - p_2), \ldots, \beta_n(P_n-p_n)]$$

$$\beta_j(P_j - p_j) = \begin{cases} 1 \; if \; p_j \in P_j \\ 0 \; otherwise \end{cases} \qquad (3)$$

$\beta_j(P_j - p_j)$ represents the state and time history of the fault in the j^{th} state, p_j is some parameter (for example time, or velocity), and P_j is a region in this parameter history where the fault is present. *The instance of the fault is declared when the value of the p_j traverses into the P_j region.*

2-2-1. Fault Dynamics

Each fault is assumed to be linearly parameterized, which can be expressed in the following form:

$$f_m(\theta,\dot{\theta},t) = \begin{bmatrix} \sum_{i=1}^{s} c^m{}_{1_i} w^m{}_{1_i}(\theta_1,\dot{\theta}_1,\tau_1) \\ \sum_{i=1}^{s} c^m{}_{2_i} w^m{}_{2_i}(\theta_2,\dot{\theta}_2,\tau_2) \\ . \\ . \\ \sum_{i=1}^{s} c^m{}_{n_i} w^m{}_{n_i}(\theta_1,\dot{\theta}_1,\tau_n) \end{bmatrix}$$

$$= \sum_{i=1}^{s} \begin{bmatrix} c_{1_i}{}^m w^m{}_{1_i}(\theta_1,\dot{\theta}_1,\tau_n) \\ c^m{}_{2_i} w^m{}_{2_i}(\theta_2,\dot{\theta}_2,\tau_2) \\ . \\ . \\ c^m{}_{n_i} w^m{}_{n_i}(\theta_1,\dot{\theta}_1,\tau_n) \end{bmatrix}$$

$$= \sum_{i=1}^{s} diag\,[C_{m_i}] W_{m_i}(\theta,\dot{\theta},\tau),$$

for m =1, 2, . . . , 2N-1 (4)

Where $C_{m_i} \epsilon R^n$ is a vector of the weights or parameters and $W_{m_i}: R^n \times R^n \times R^+ \rightarrow R^n$ is a vector of dynamic functions.

2-2-2. State and Torque-dependent Faults

The fault dynamics can be represented as

$$f_m(\theta,\dot{\theta},\tau,t) = f_{m\theta}(\theta,\dot{\theta}) + f_{m\tau}(\tau) \qquad (5)$$

Where $f_{m\tau}(\tau)$ and $f_{m\theta}(\theta,\dot{\theta})$ represent *torque-dependent* and *state-dependent* faults respectively.

Summarizing this section's analysis of the faults in the robotic system, we arrive at the following comprehensive model of the robotic system:

$$\underbrace{M(\theta)\theta + V(\theta,\ddot{\theta}) + G(\theta) + \mu(\theta,\dot{\theta},\tau,t)}_{Robotic\ System\ s\ Dynamics} + \underbrace{+\beta(P-p)[f_{m\theta}(\theta,\dot{\theta}) + f_{m\tau}(\tau)]}_{Fault\ Dynamics} = \underbrace{\tau}_{Inpu\,t\ Torque} \qquad (6)$$

3. Detection/Approximation Observers

The detection/approximation observer is a multifunction mechanism that bonds the entire FTC scheme together. While the system is healthy it is used to monitor it for faults and detect them if they do occur. During the subsequent stages, it is used to approximate and accommodate unknown fault dynamics, and to monitor the system for fault absence. Each of the detection/approximation observer application becomes evident in later sections. It is carefully designed to be robust with respect to unmodeled dynamics, and state and torque-dependent faults.

The *approximated* torque-dependent and state-dependent fault dynamics in an *n* degree of freedom system can be represented by the following equations:

$$\hat{f}_\tau(\tau,t) = \begin{bmatrix} h_1(t)\tau_1 \\ h_2(t)\tau_2 \\ . \\ . \\ h_n(t)\tau_n \end{bmatrix} = \begin{bmatrix} h_1(t) & 0 & . & . & 0 \\ 0 & h_2(t) & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & . & . & h_n(t) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ . \\ . \\ \tau_n \end{bmatrix}$$

$$= diag[H(t)]\tau$$

$$\hat{f}_\theta(\theta,\dot{\theta},t) = \sum_{i=1}^{k} \begin{bmatrix} l_{1_i}(t) q_{1_i}(\theta_1) \\ l_{2_i}(t) q_{2_i}(\theta_2) \\ . \\ . \\ l_{n_i}(t) q_{n_i}(\theta_n) \end{bmatrix} + \sum_{i=1}^{k} \begin{bmatrix} s_{1_i}(t) z_{1_i}(\dot{\theta}_1) \\ s_{2_i}(t) z_{2_i}(\dot{\theta}_2) \\ . \\ . \\ s_{n_i}(t) z_{n_i}(\dot{\theta}_n) \end{bmatrix}$$

$$= \sum_{i=1}^{k} \left(diag[L_i(t)] Q_i(\theta) + diag[S_i(t)] Z_i(\theta) \right) \qquad (7)$$

Where: $H(t) \in R^n, L_t(t) \in R^n$ and $S_i(t) \in R^n$ are the vectors of the weights or parameters. In equation (7) the velocity and the position dynamics are decoupled for analytical purposes. It does not affect the approximation effort, although it allows detecting the position-dependent faults and the velocity-dependent faults individually. Both velocity - dependent and position-dependent dynamics of the fault are approximated using RBF neural network structures composing the $Q_i(\theta) \in R^n$ and $Z_i(\dot{\theta}) \in R^n$ vectors, and are structured as follows:

$$q_{ij}(\theta_j) = exp\left(-\frac{(\theta_j - a_{ij})^2}{\sigma_{ij}{}^2}\right),\ z_{ij}(\theta_j) = exp\left(-\frac{(\theta_j - b_{ij})^2}{\omega_{ij}{}^2}\right)$$

$$for = \begin{cases} i = 1,2,\ldots,k \\ j = 1,2,\ldots,n \end{cases} \quad (8)$$

Where a_{ij}, b_{ij}, σ_{ij} and ω_{ij} are the parameters of these networks [2] [7].

The detection/approximation observer is proposed:

$$\ddot{\hat{\theta}} = -M^{-1}(V+G) + M^{-1}(I - diag[H])\tau - M^{-1}\sum_{i=1}^{k}(diag[L_i]Q_i + diag[S_i]Z_i) - \gamma\left(\dot{\hat{\theta}} - \dot{\theta}\right) \quad (9)$$

Where: $\gamma = diag[\gamma_1, \gamma_2, \ldots, \gamma_n]$ is a positive definite stability matrix [4] [5].

3-1. Detection

Let $e_o = \dot{\hat{\theta}} - \dot{\theta}$ denote the state estimation error, which will serve also as the *residual vector* [5] [2]. During the detection stage, the FTC monitors the system for the presence of the faults. While the system is healthy or no fault is present, the true system dynamics is represented as follows:

$$\dot{\theta} = -M^{-1}(V+G) + M^{-1}\tau \quad (10)$$

While the system is healthy, the approximation model has the following form:

$$\ddot{\hat{\theta}} = -M^{-1}(V+G) + M^{-1}\tau - \gamma\left(\dot{\hat{\theta}} - \dot{\theta}\right) \quad (11)$$

The dynamics of the estimation error in this case will be equal to:

$$\dot{e} = -\gamma e \quad (12)$$

3-2. Approximation

Let:

$\grave{H} = H - H^*$, $\grave{L_t} = L_t - L^*$, and $\grave{S_t} = S_t - S^*$

Consequently we obtain:

$$\tilde{e}_0 = -\gamma e_0 M^{-1}\left[diag[\widetilde{H}]\tau + \right]\sum_{i=1}^{k}\left(diag\left[\breve{L_i}\right]Q_i + diag\left[\breve{S_i}\right]Z_i\right) - \eta] \quad (13)$$

A Lyapunov function of the following form is employed:

$$U = \frac{1}{2}e_0^T e_0 + \frac{1}{2}\widetilde{H^T}T^{-1}\widetilde{H} + \frac{1}{2}\sum_{i=1}^{k}\widetilde{L_i^{\Gamma}}\Psi^{-1}\widetilde{L_i} + \frac{1}{2}\sum_{i=1}^{k}\widetilde{S_i^{\Gamma}}\Upsilon^{-1}\widetilde{S_i} \geq 0$$

Where $\Gamma, \Psi, Y \in R^{R\times a}$ are adaptive gain matrices gains. Therefore:

$$\dot{U} = e_0^T\dot{e}_0 + \dot{H^T}\Gamma^{-1}\widetilde{H} + \sum_{i=1}^{k}\dot{L_i^{\Gamma}}\Psi^{-1}\widetilde{L_i} + \sum_{i=1}^{k}\dot{S_i^{\Gamma}}\Upsilon^{-1}\widetilde{S_i}$$

By setting :

$$\dot{H^T}\Gamma^{-1} = e_0^T M^{-1}diag[\tau] \ or\ \dot{H} = \Gamma diag[\tau]M_{e_0}^{-1}$$

$\dot{L_i^T}\Psi^{-1} = e_0^T M^{-1}diag[Q_i]\ or\ \dot{L_i} = \Psi diag[Q_i]M_{e_0}^{-1}$, for i =1, 2,…, k

$$\dot{S_i^T}\Psi^{-1} = e_0^T M^{-1}diag[Z_i]\ or\ \dot{S_i} = \Upsilon diag[Z_i]M_{e_0}^{-1}$$

We obtain:

$$\dot{U} = -e^T\gamma e - e^T M^{-1}\eta \quad (14)$$

When $\eta = 0$, we obtain: $\dot{U} = -e^T\gamma e \leq 0$
Which is a negative semi-definite matrix, and therefore the approximation error will converge to zero. When $\eta \neq 0$, we obtain:

$$\dot{U} = -e_0^T\gamma e_0 - e_0^T M^{-1}\eta$$

Following the previous analysis, the approximation observer's equation will be:

$$\ddot{\hat{\theta}} = -M^{-1}(V+G) + M^{-1}(I + diag[H])\tau + M^{-1}\sum_{i=1}^{k}(diag[L_i]Q_i + diag[S]) - \gamma e \quad (15)$$

3-3. Accommodation

Under healthy conditions, the nominal input torque $\tau = \tau_0$ is given by:

$$\tau_0 = M(\theta)\left\lfloor K_P\left(\theta - \dot{\theta}_d\right) + K_d\left(\dot{\theta} - \dot{\theta}_d\right) + \ddot{\theta}_d\right\rfloor + V\left(\theta, \dot{\theta}\right) + G(\theta) \quad (16)$$

Where $\theta_d, \dot{\theta}_d, \ddot{\theta}_d \in R^n$ are the vectors of desired joint positions, velocities, and accelerations, respectively, and $K_p \in R^{a\times n}$ and $K_y \in R^{n\times R}$ are negative definite matrices, which are designed, so that exponential convergence of the tracking errors is achieved.
Applying the proposed torque and stage-dependent fault models, the input should have the following structure:

$$\tau = \begin{cases} (I - diag[H(t)])^{-1} \left(\tau_0 + \hat{f}_\theta(\theta, \dot{\theta}, t)\right) \\ \tau_0 \end{cases} \quad (17)$$

4. Simulations results

For simplicity, a three-link SCARA robot is utilized in this study. The dynamic equations, which can be derived via the Euler-Lagrangian method, are represented as follows:

$$\begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + l_1 l_2 sin(q_2) \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3 g \end{bmatrix}$$

Where:

$$D_{11} = l^2{}_1 \left(\frac{m_1}{3} + m_2 + m_3\right) + l_1 l_2 (m_2 + 2m_3) cos(q_2) + l^2{}_2 \left(\frac{m_2}{3} + m_3\right)$$

$$D_{13} = D_{23} = D_{31} = D_{33} = 0$$

$$D_{12} = -l_1 l_2 \left(\frac{m_2}{2} + m_3\right) cos(q_2) - \left(\frac{m_2}{3} + m_3\right) = D_{21}$$

$$D_{22} = l^2{}_2 \left(\frac{m_2}{3} + m_3\right), \ D_{33} = m_3$$

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

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

In which q_1, q_2 and q_3 are the angles of joints 1, 2 and 3; m_1, m_2 and m_3 are the mass of links 1, 2 and 3; l_1, l_2 and l_3 are the length of links 1, 2 and 3; g is the gravity acceleration. Moreover, the system parameters of the SCARA robot are selected as:

$l_1 = 1.0\,m \quad l_2 = 0.8m \quad l_3 = 0.6m$

$m_1 = 1.0kg \ \ m_2 = 0.8kg \ \ m_3 = 0.5kg \ \ g = 9.8$

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 [12].

In the joints (components), the most common and ever present type of faults is friction. Friction has been extensively analyzed and varieties of models are available. Friction models in the works by C. Canudas de Wit [8] [9] [10] provide an excellent reflection of friction in the real joint:

- Coulomb / Sticktion : $f(\dot{\theta}) = \alpha sign(\dot{\theta})$
- Viscous : $f(\dot{\theta}) = \alpha\, \dot{\theta}$

In SCARA manipulators, actuators are generally electric motors. Faults in rotating electric motors may be classified as electric faults, rotational faults and vibration faults:

$f(\tau) = \alpha\tau, \ -1 < \alpha \leq K \leq \infty$

Where K is some maximum value that α can reach.

The first stage of the numerical study analyzes performance of the detection/approximation observer. Figure 1 - Figure 9 demonstrate results of such study with an example of actuator and component fault detection and accommodation in a SCARA robot. The previously described fault dynamics are applied in this simulation.

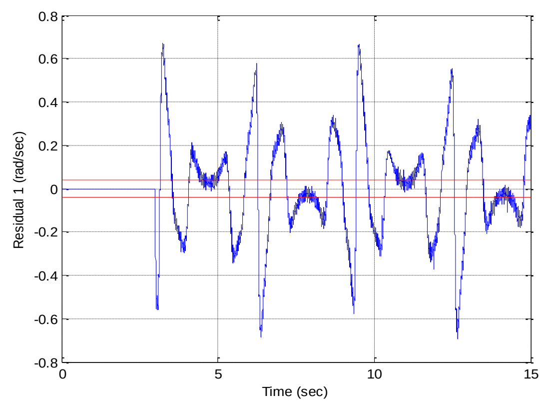

Fig. 1. Residual 1

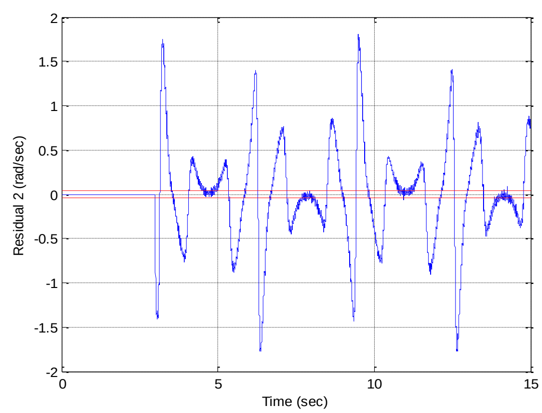

Fig. 2. Residual 2

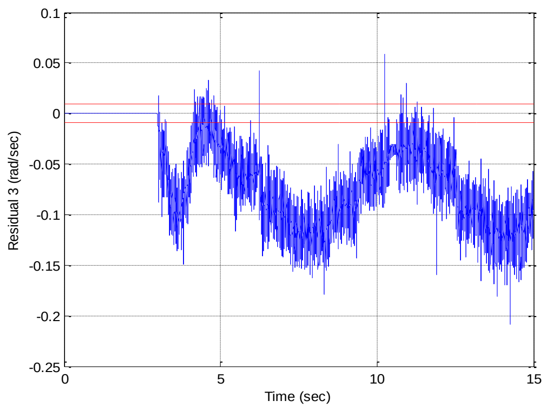

Fig. 3. Residual 3

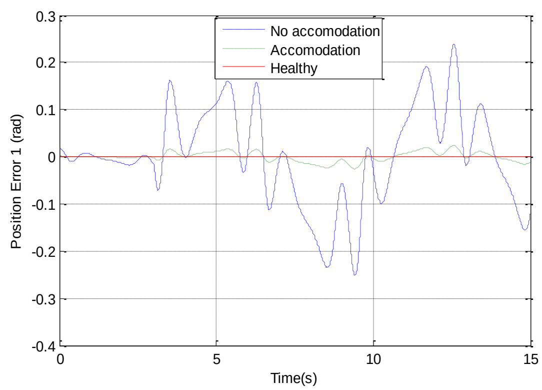

Fig. 4. Position error 1

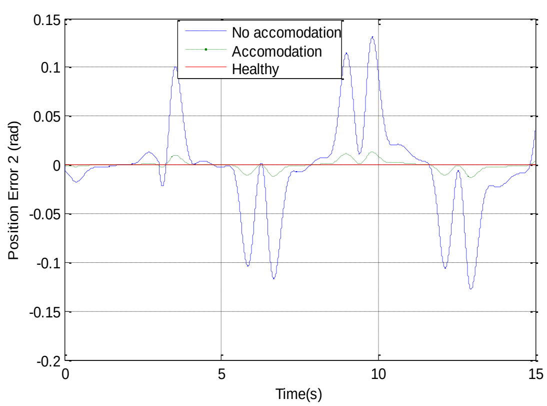

Fig. 5. Position error 2

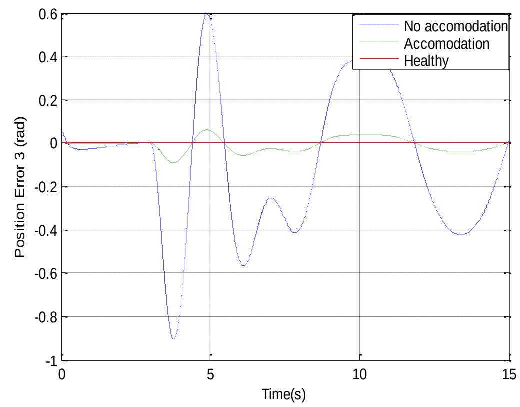

Fig. 6. Position error 3

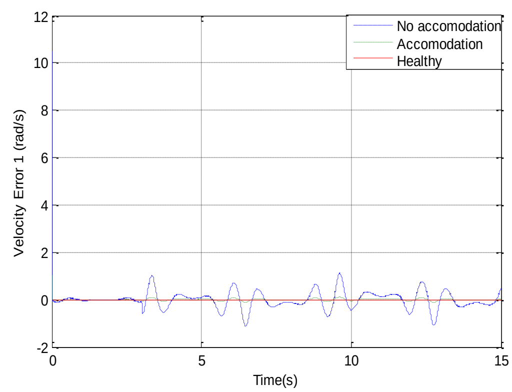

Fig. 7. Velocity error 1

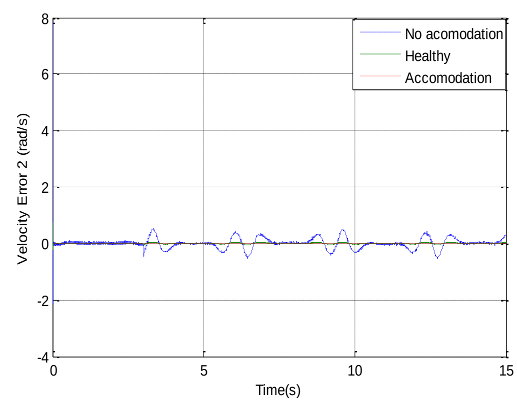

Fig. 8. Velocity error 2

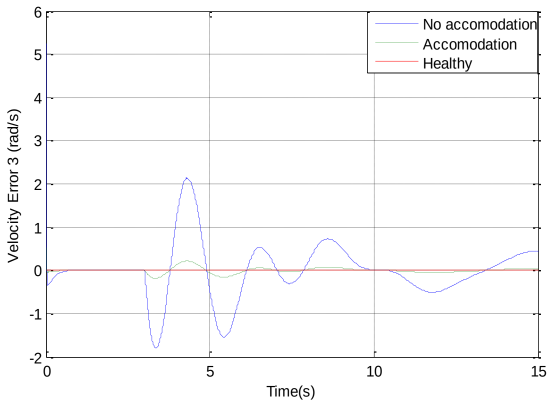

Fig. 9. Velocity error 3

5. Conclusion

The new modeling technique was used to develop a very effective approach that both monitors the robotic system's health and its environment, and provides significant improvements to its performance. It is robust with respect to unmodeled dynamics. Detection, Isolation, and Accommodation (FTC) can be easily reshaped to work with a wide variety of systems and faults. One of the great advantages of the approach is that it can be applied to hydraulic, electrical or other types of robotic systems with minor modifications. This approach gives robotic system the tools to be aware of its constantly changing internal and external environment, identify or learn any faults, and accommodate them.

The results of simulations in the case of a healthy, but subject to uncertainties modeling show that residues vary and are moving away from zero. They are therefore sensitive to uncertainties in modeling. This creates false alarms and false detections. Future works is to introduce a threshold of detection and show how it can improve defect detection and reduce false alarms.

References

1. Kim, Y. H., Lewis, F. L.: *High-Level Feedback Control with Neural Network.* In: World Scientific Publishing Co. Pte. Ltd., 1998.

2. Polycarpou, M. M., Helmicki, A. J.: *Automated Fault Detection and Accommodation: A Learning Systems Approach.* In: IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no. 11, pp. 1447-1458, 1995.

3. Patton, R.J.: *Fault tolerant control: The 1997 situation (survey)*. In: IFAC SAFEPROCESS'97, Hull, UK, Aout 26-28, Vol. 2, pp.1033-1055, 1997.

4. Assada, H., Slotime, J. J. E.: *Robot Analysis and Control*. In: A Wiley Interscience, 1986.

5. Slotine, J. J. E., Li, W.: *Applied Nonlinear Control*. In: Prentice Hall, 1991.

6. Robert J. Schilling, *Fundamentals of Robotics: Analysis and Control*. In: Prentice Hall, 1990.

7. Terra, M. H., Tinos, R.: *Fault Detection and Isolation in Robotic Manipulators via Neural Networks: A Comparison Among Three Architectures for Residual Analysis .*
In: Journal of Robotic Systems, pp. 357-374, 2001.

8. Canudas de Wit, C., Noel, Aubin, P. A., Brogliato, B., Drevet, P.: *Adaptive Friction Compensation in Robot Manipulators*. In: Low-Velocities", *IEEE*, pp. 1352-1357, 1989.

9. Canudas de Wit, C., Olsson, H. ,Astrom, K. J., Lischinsky, P.: *A New Model for Control of Systems with Friction.* In: IEEE Transactions on Automatic Control, vol. 40, no. 3, 1995.

10. Canudas de Wit, C., Ge, S. S.: *Adaptive Friction Compensation for Systems with Generalized Velocity/Position Friction Dependency*. In: IEEE Proceedings of the 36th Conference on Decision & Control, pp. 2465-2470, 1997.

11. P. Mhaskar, A. Gani, N. H. El-Farra, C. McFall, P. D. Christodes, and J. F.Davis.: *Integrated fault-detection and fault-tolerant control for process systems*. In:AIChE Journal, pp 2129-2148, 2006.

12. P. Mhaskar, C. McFall, A. Gani, P. D. Christodes, and J. F. Davis.: *Fault-tolerant control of nonlinear systems: Fault detection and isolation and controller reconfiguration*. In: Proceedings of the American Control Conference,pp 5115-5122, Minneapolis, Minnesota, 2006.