Neural Network based Fault Diagnosis and Accommodation in Robotic Manipulators

by

Arun T. Vemuri, Sotiris A. Diakourtis and Marios M. Polycarpou
Department of Electrical and Computer Engineering
University of Cincinnati
Cincinnati, Ohio 45221-0030, U.S.A.

Abstract

Fault detection, diagnosis and accommodation play a key role in the operation of autonomous and intelligent control systems. System faults, which typically result in changes in critical system parameters or even system dynamics, may lead to degradation in performance and unsafe operating conditions. This paper investigates the problem of fault diagnosis and accommodation (FDA) in rigid-link electrically driven (RLED) robotic manipulators. Neural networks are used as on-line approximators for monitoring the system for faults. A learning algorithm is described so that the neural network provides a way, not only for detecting a fault, but also for constructing a model of the fault characteristics that can be used for accommodation purposes. Simulation examples are presented to illustrate the ability of the neural network based FDA methodology described in this paper, to detect and accommodate faults in a RLED robotic system.

1 Introduction

Robotic systems are widely used in many engineering applications. Usually, robotic manipulators are used in environments which are remote, hazardous and which demand very high performance, productivity and, above-all, safe operation. Application environments include manufacturing processes [4], hazardous waste management and clean-up, and space-based operations [18]. In such operations, system faults (which are typically characterized by critical changes in the system parameters, or even, by changes in the inherent dynamics of the system) can potentially result not only in the loss of productivity but also in unsafe operation of robotic systems. Moreover, difficult and often dangerous environments limit the ability of humans to perform any supervisory and/or corrective tasks. Hence automated monitoring of the robotic manipulator for any faults and, if possible, effective accommodation of such faults plays a crucial role in the use of robotic manipulators.

The process of system fault characterization can be broken up into three steps: (i) *detection* deals with determining if a malfunction has occurred in the system; (ii) *diagnosis* considers the problem of isolating and/or identifying a fault; and (iii) *accommodation* attempts to self-correct a particular fault through reconfiguration of the control system.

A number of researchers have worked on the problem of automated fault diagnosis and accommodation (FDA) in robotic systems using *analytical redundancy* based methods. A state estimation based technique is used in [22] for detecting faults in robotic systems wherein it is concluded that the fault detection abilities of nonlinear observer based FDA approach is significantly better than the linear Luenberger observer based FDA approach. Parameter estimation based methods are used in [4], [13] to monitor and identify changes in critical parameters due to faults in robotic systems. Expert system based methodologies have also been considered for FDA purposes in [18], [19] among others. In [1] time series analysis of the data is used for detecting faults in a robotic system. The issue of faulty behavior in robotic systems due to actuator saturation is addressed in [16] and accommodation of such faults using the time regulation and the torque distribution method is proposed. In [9] a special force-torque-sensor is applied for automatic fault detection by analysis of its signal in the frequency domain.

Most of the above FDA studies in robotic systems has dealt almost exclusively with *linear* modeling techniques. Although the need for nonlinear modeling methods in FDA has long been recognized [5], difficulties in formulating, analyzing, and implementing such techniques have prevented their wider use. In this paper, we employ a neural network based methodology, developed in [12], for detecting and accommodating faults in rigid link electrically driven (RLED) robotic systems. The emergence of the neural network paradigm as a powerful tool for learning complex mappings from a set of examples has triggered interest in using neural network models, and, more generally, nonlinear modeling techniques, for fault diagnosis and accommodation. The feasibility of applying neural networks to FDA has been demonstrated via simulations in several studies, including the diagnosis of chemical processes [20], fault detection in aircraft [10], and fault accommodation in underwater vehicle systems [3].

The organization of the paper is as follows. In Section 2, the dynamic model of the RLED robotic manipulator is described. In Section 3, the FDA methodology is described briefly. Simulation results showing the application of the neural network based FDA methodology on a two-link RLED robotic system are presented in Section 4. Section 5 has some concluding remarks.

2 RLED Robotic Manipulator

In this section, a model of RLED robot manipulator and its control is described. The manipulator is modeled as a set of moving rigid bodies which are joined together with revolute joints, each joint being driven by a DC motor. The equation of motion of the ith link of an n-link robot manipulator is described by [15]

$$\sum_{j=1}^{n} D_{ij}\ddot{q}_j + J_i\ddot{q}_i + \sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\dot{q}_j\dot{q}_k + D_i = \tau_i, \quad (1)$$

where q_i, $\dot{q}_i$ and $\ddot{q}_i$ are the position, velocity and acceleration of joint i respectively, D_{ii} and D_{ij} $(j \neq i)$ are the the effective and coupling inertias respectively, J_i is the reflected actuator inertia of joint i, D_{ijj} and D_{ijk} $(j \neq k)$ are the coefficients of centripetal and Coriolis forces respectively, D_i represents torques due to gravity, and τ_i is the torque acting at joint i.

The dynamics of a permanent magnet brush DC motor used for actuating the ith link are described by [2]

$$R_i i_i + L_i\frac{di_i}{dt} + K_i^e N_i\dot{q}_i = u_i, \quad (2)$$

where R_i and L_i are the resistance and inductance of the armature circuit respectively, K_i^e is the back EMF constant, N_i is the gear ratio, i_i is the armature current, and u_i is the input voltage. Using equations (1) and (2) and the fact that $\tau_i = N_i K_i^t i_i$, where K_i^t is the torque constant of the ith link motor, we can obtain the combined dynamics of an RLED robot manipulator in the compact form as [15]

$$\bar{D}(q)\dddot{q} + \bar{P}(q,\dot{q},\ddot{q}) = u, \quad (3)$$

where $q = [q_1\ q_2\ ...\ q_n]^T$ and $u = [u_1\ u_2\ ...\ u_n]^T$. The $n \times n$-dimensional matrix, $\bar{D}$, and the n-dimensional vector, $\bar{P}$, describe the combined dynamics of the robot manipulator and the DC motor. It is known that the matrix $\bar{D}$ is invertible [15].

Based on (3), a state-space representation of the RLED robotic system is described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \bar{D}^{-1}(u-\bar{P}), \end{aligned} \quad (4)$$

where $x_1 = q$, $x_2 = \dot{q}$, $x_3 = \ddot{q}$ denote vectors of position, velocity and acceleration respectively.

Control laws for RLED robot manipulators can be obtained using nonlinear feedback linearization techniques [7], [14]. If the control objective for the RLED robot manipulator (3) is to track a desired trajectory, q_d, a control law based on the feedback linearization technique is given by

$$u = \bar{D}(\dddot{q}_d + K_3\ddot{e} + K_2\dot{e} + K_1 e) + \bar{P}, \quad (5)$$

where $e := q_d - q$ is the tracking error and K_1, K_2 and K_3 are positive definite diagonal matrices of dimension $n \times n$. If the robot dynamics are known exactly, then these matrices can be chosen so that the control law leads to an exponentially convergent tracking error [7], [14].

3 Fault Detection and Accommodation

The robotic manipulator (1), driven by a DC motor described by (2) and the control law (5), is guaranteed to track desired trajectories provided the system parameters are known. However, in the case of a fault, system parameters or even system dynamics may change leading to tracking errors and/or instabilities. In this section we describe a learning methodology for constructing automated FDA architectures. The main idea behind this approach is to monitor the physical system for any off-nominal behavior using nonlinear modeling techniques. Our principal design tool is a generic function approximator with adjustable parameters; we refer to this "structure" as *on-line approximator* [12]. Although on-line approximators are quite general models, the original inspiration for using such methods in FDA is based on recent developments in neural network technology.

A general framework for modeling various fault situations in the RLED robotic manipulator model described by (4) is provided by

$$\begin{aligned} \dot{x}_1 &= x_2 + \beta_1(t-T) f_1(x,u) \\ \dot{x}_2 &= x_3 + \beta_2(t-T) f_2(x,u) \\ \dot{x}_3 &= \bar{D}^{-1}(u-\bar{P}) + \beta_3(t-T) f_3(x,u), \end{aligned} \quad (6)$$

where $x = [x_1^T \;\; x_2^T \;\; x_3^T]^T$, $f_i : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ is a smooth map representing changes in the system parameters or dynamics due to a fault, and $\beta_i : \mathbb{R} \mapsto \mathbb{R}$, is a function representing the time profiles of faults for $i = 1, 2, 3$.

An accurate description of fault situations, most often, requires nonlinear modeling of faults, which is what is described by (6). The nonlinear modeling capability is reflected in allowing the deviations f_i due to faults to be nonlinear functions of the state x and the input u.

Assuming that all the states of the system are measurable, we use the following estimation model of (6) to generate the error measure required in the learning scheme [12]:

$$\begin{aligned} \dot{\hat{x}}_1 &= G_1(\hat{x}_1 - x_1) + x_2 + \hat{f}_1(x,u;\hat{\theta}_1), \\ \dot{\hat{x}}_2 &= G_2(\hat{x}_2 - x_2) + x_3 + \hat{f}_2(x,u;\hat{\theta}_2), \\ \dot{\hat{x}}_3 &= G_3(\hat{x}_3 - x_3) + \bar{D}^{-1}(u-\bar{P}) + \hat{f}_3(x,u;\hat{\theta}_3) \end{aligned} \quad (7)$$

where for each $i = 1, 2, 3$, G_i is a constant square matrix of dimension $n \times n$, whose eigenvalues lie in the left-half complex plane, $\hat{f}_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{p_i} \mapsto \mathbb{R}^n$ represents an on-line approximator structure and $\hat{\theta}_i \in \mathbb{R}^{p_i}$ represents the weights or the parameters of the approximator.

In the above formulation, the objective to detect faults translates into the problem of adjusting $\hat{\theta}_i(t)$ at each time t so that $\hat{f}_i(x,u;\hat{\theta}_i)$ approximates the fault function, $\beta_i(t-T) f_i(x,u)$, as closely as possible (note that in the case of normal operation each $\hat{f}_i$ should be close to zero). Hence the output of the approximator could be used to indicate the occurrence of a fault. Furthermore, the function $\hat{f}_i$ provides the fault characteristics and, therefore, it can be used for identification and possibly, accommodation of system faults. The price that one has to pay for the potential to model a much larger class of faults is the need to approximate the unknown nonlinear functions f_i, on-line. However, recent advances in both hardware implementation and software simulation tools have rendered possible the use of on-line approximation methods (such as multlayer sigmoidal neural networks) for constructing and analyzing nonlinear models [21].

Based on the estimated model (7) and using the *Lyapunov synthesis approach* [11] along with the *projection algorithm* [8], we obtain the following adaptive law for the adjustment of the parameter estimates [12]:

$$\dot{\hat{\theta}} = \mathcal{P}\left(\Gamma \left[\frac{\partial \hat{f}(x,u;\hat{\theta})}{\partial \hat{\theta}}\right]^T (x - \hat{x})\right), \quad (8)$$

where $\hat{\theta} = [\hat{\theta}_1^T \;\; \hat{\theta}_2^T \;\; \hat{\theta}_3^T]^T$ and $\hat{f} = [\hat{f}_1^T \;\; \hat{f}_2^T \;\; \hat{f}_3^T]^T$. In (8), $\Gamma = \Gamma^T > 0$ is the learning rate matrix and $\mathcal{P}$ denotes the projection algorithm. The projection algorithm, which restricts the parameters to a bounded region, is used in order to avoid parameter drift. The restriction is achieved by projecting the standard adaptive law (obtained via the Lyapunov synthesis approach) onto the tangent hyperplane if the current value of $\hat{\theta}$ is at the boundary and is directed outwards. We note that in the case of a multi-layer neural network, the back-propagation algorithm along with the projection algorithm provides a convenient, structured method for computing the parameter trajectories. The stability properties of the learning scheme described by (7), (8) as applied to the FDA problem, has been analyzed in [12].

Failure accommodation (also referred to as self correction or control reconfiguration) is one of the major challenges in designing intelligent control systems. In such situations, learning methodologies are required to perform simultaneous on-line identification and control of the post-fault system. This procedure corresponds to indirect adaptive control, which is well known in the adaptive linear control literature [11]. In the nonlinear case, however, the problem becomes considerably more complex because the control is required to reject (or, at least, dampen) the effect of the fault by cancelling the nonlinear function representing the deviation in the dynamics due to a fault.

One of the nonlinear control tools available for such problems is feedback linearization [6]. The main idea behind feedback linearization is to transform the nonlinear system into a linear one through a change of coordinates and nonlinear feedback. If feedback linearization is achievable (see [6] for conditions under which a system is feedback linearizable), then it is possible to achieve, first, cancellation of the nonlinear functions and, second, desired closed-loop performance through the application of powerful linear control methodologies.

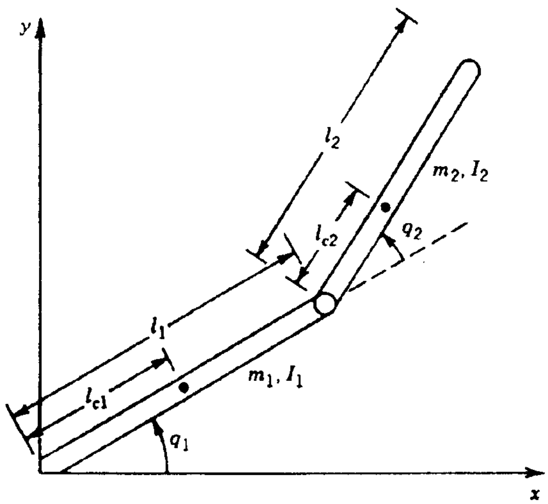

Figure 1: Two-link planar robot.

$j = 1, 2$	Link 1	Link 2
R_j (Ω)	1.5	1.5
L_j (H)	8×10^{-5}	8×10^{-5}
K_j^e $(V-s)$	25.05	21.71
K_j^t $(V-s)$	25.05	21.71
l_j (m)	0.45	0.20
l_{cj} (m)	-0.15	-0.10
m_j (Kg)	100.0	25.0
I_j $(Kg.m^2)$	6.25	0.61
J_j $(Kg.m^2)$	4.77	3.58

Table 1: Manipulator and Motor parameters

4 Simulation Results

Example simulations are carried out on a benchmark robotic system proposed in [17]. The benchmark robotic system is a two-link planar RLED robot manipulator, as shown in Figure 1, with DC motor actuators. The link and the motor parameters are given in Table 1. The controller gains in the control law (5) are chosen as follows: $K_1 = 1000I_{2\times2}$, $K_2 = 300I_{2\times2}$, $K_3 = 30I_{2\times2}$, where $I_{n\times n}$ represents an identity matrix of dimension $n \times n$.

Defining the state-vector of the two-link RLED model as $x_1 := [q_1\ q_2]^T$, $x_2 := [\dot{q}_1\ \dot{q}_2]^T$, $x_3 := [\ddot{q}_1\ \ddot{q}_2]^T$ and based on the methodology described in the previous section, we use the estimation model

$$\begin{aligned} \dot{\hat{x}}_1 &= -p(\hat{x}_1 - x_1) + x_2 \\ \dot{\hat{x}}_2 &= -p(\hat{x}_2 - x_2) + x_3 \\ \dot{\hat{x}}_3 &= -p(\hat{x} - x_3) + \bar{D}^{-1}(u - \bar{P}) + \hat{f}(x;\hat{\theta}), \end{aligned}$$

where $-p$ is the location of the eigenvalues of the stability matrices G_i; in our simulations, we set $p = 10$. Note that the above estimation model requires all state measurements. However, the state measurements coincide with the measurements required for the the control law (5). Hence for the implementation of the FDA methodology, no additional hardware, in the form of sensors, are necessary.

We use a three-layer sigmoidal neural network whose ith output has the form [21]

$$\hat{f}_i(x;\hat{\theta}) = \sum_{j=1}^{N} \hat{\theta}_{j1}(1 + exp(x^T\hat{\theta}_{j2} + \hat{\theta}_{j3}))^{-1},$$

where $\hat{\theta}_{j1}$, $\hat{\theta}_{j2}$, $\hat{\theta}_{j3}$ are adjustable parameters and N is the number of neurons in the hidden layer; in our simulations, we use 35 hidden neurons.

Three simulation experiments are performed. In the first experiment we simulate the robotic system without any fault. In the second experiment, we investigate the detection of system faults while in the third experiment we consider the problem of reconfiguring the control law in order to accommodate the system faults.

Normal operation of the Manipulator: We first simulate the robot manipulator under normal operating conditions. The weights of the neural network are initialized so that the neural network outputs are zero. The learning rate is chosen as $\Gamma = 0.75I_{2\times2}$. Figure 2 shows the plot of the desired and the actual trajectories of the joint angles. It is clear that the joint angles follow the required trajectories. The figure also shows the plot of the neural network outputs. From the figure it is clear that the neural network outputs remains close to zero, which corresponds to a no-fault situation.

Detection of Faults: In this experiment, the manipulator is simulated in the presence of faults and with no reconfiguration of the control law. The weights of the neural network are initialized such that the outputs of the neural network are zero and the learning rate is chosen as $\Gamma = 0.75I_{2\times2}$. We first simulate the system with one fault which occurs occurs at $t = 6$s and is due to a 5% change in the time constant of the DC motor armature circuit of joint 1. Figure 3 shows the plot of the joint angles and the norm of the neural network output. The figure shows that the fault causes a large tracking error in Link 1. It is also clear from Figure 3 that when the fault occurs, the neural network output 1 jumps to a non-zero value very quickly, indicating a fault in Link 1.

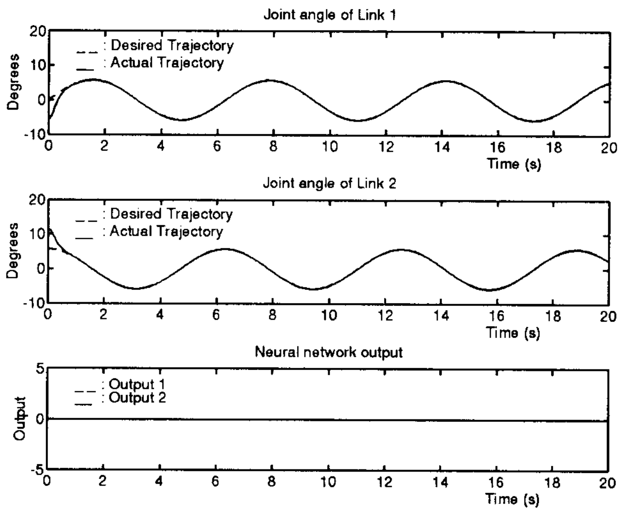

Figure 2: Normal operation: Joint angles and Neural Network outputs.

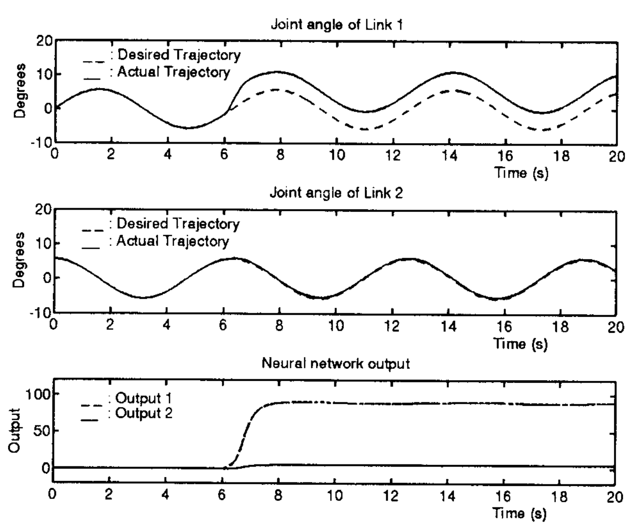

Figure 3: Operation with one fault: Joint angles and the Neural Network outputs.

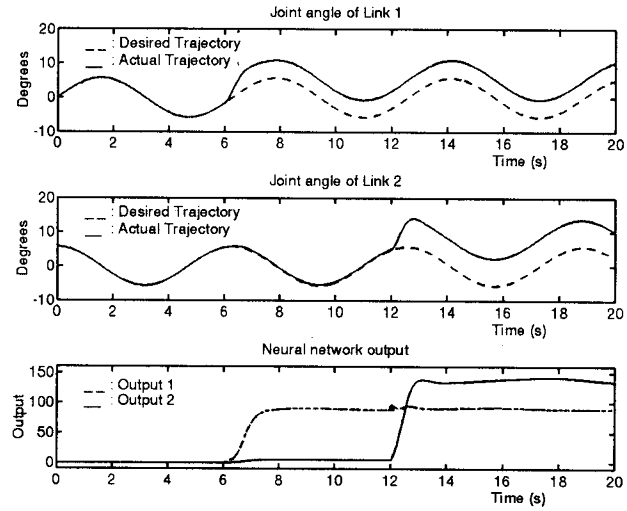

Figure 4: Operation with two faults: Joint angles and the Neural Network outputs.

Next, we simulate the system with two faults; one fault occurs at $t = 6$s and is due to a 5% change in the time constant of the DC motor armature circuit of joint 1 while the second fault occurs at $t = 12$s and is due to a similar change in the time constant of the armature circuit of the DC motor of joint 2. Figure 4 shows the joint angles and the neural network outputs. The figure shows that the error between the desired and the actual trajectories of the joint angles increases considerably after the occurrence of each fault. It is also clear from Figure 4 that when the second fault occurs at $t = 12$s, the neural network output 2 jumps to a non-zero value very quickly. Thus the neural network outputs can be used not only to indicate the occurrence of a fault in the robotic manipulator but can also be used to identify the location of the fault.

Accommodation of faults: In the final experiment, we consider the accommodation of faults by reconfiguring the control law. Based on the state-space representation of the RLED robotic system and the feedback linearization technique, we obtain the reconfigured control law

$$u_r = u - \bar{D}\hat{f},$$

where u_r represents the reconfigured control used to accommodate the fault in the manipulator. The weights of the neural network were again initialized so that the outputs of the neural network are zero and the learning rate is chosen as $\Gamma = 0.75 I_{2\times 2}$. We first simulate the manipulator with one fault which occurs at $t = 6$s and is due to a 5% change in the time constant

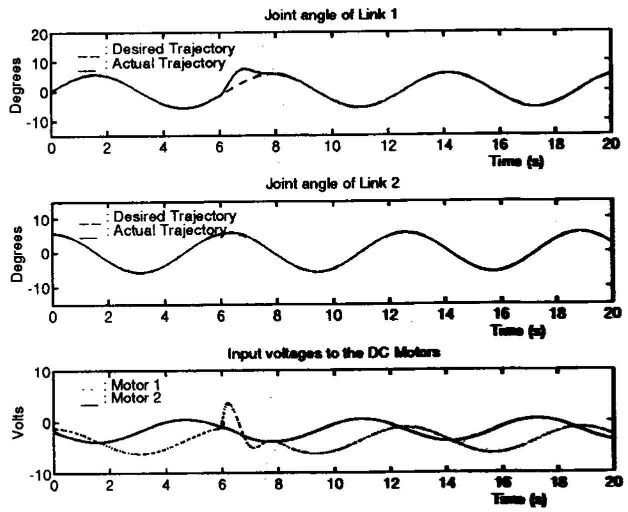

Figure 5: Accommodation under one fault: Joint angles and the Voltage inputs to the DC motors.

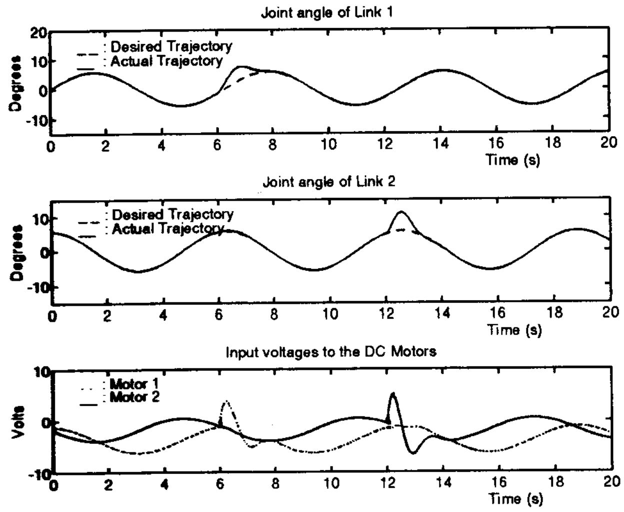

Figure 6: Accommodation under two faults: Joint angles and the Voltage inputs to the DC motors.

of the DC motor armature circuit of joint 1. Figure 5 shows that the reconfiguration of the control law results in the tracking of the desired trajectories by the joint angles. The figure also shows the voltage input to the motors as a result of the reconfiguration of the control law remains within reasonable bounds.

We next simulate the manipulator with two faults; one fault occurs at $t = 6$s and is due to a 5% change in the time constant of the DC motor armature circuit of joint 1 while the second fault occurs at $t = 12$s and is due to a change in the time constant of the armature circuit of the DC motor of joint 2. Figure 6 shows that the reconfiguration of the control law in the presence of two faults results in the tracking of the desired trajectories. The figure also shows the voltage input to the motors as a result of the reconfiguration of the control law.

5 Conclusions

A neural network based methodology is used for fault detection and accommodation. Simulations show that abrupt jumps in the armature circuit time constant of the DC motors can be effectively detected, diagnosed and accommodated. Thereby, the performance of the robotic control system is maintained. These preliminary results of using nonlinear modeling techniques for FDA purposes are very encouraging.

In this paper, neural network outputs are used for FDA purposes in the robotic manipulator. Using this method to detect faults could lead to false alarms when there are modeling uncertainties. Hence a more robust mechanism for fault detection needs to be investigated. The effects of modeling uncertainties on the detection and accommodation of the fault needs a rigorous mathematical treatment vis-a-vis the stability and the overall performance of the methodology and is left for future work.

References

[1] W. Bucken and B. Freyermuth. Fault detection in industrial processes by statistical methods in the example of an industrial robot. *ITG-Fachberichte*, 116:123–136, 1991.

[2] D. M. Dawson, J. J. Caroll, and M. Schneider. Integrated backstepping control of a brush dc motor turning a robotic load. *IEEE Trans. Control Systems Tech.*, 2(3):233–244, 1994.

[3] J. A. Farrell, T. Berger, and B. D. Appleby. Using learning techniques to accommodate unanticipated faults. *IEEE Control Systems*, 13(3):40–49, 1993.

[4] B. Freyermuth. An approach to model based fault diagnosis of industrial robots. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1350–1356, Sacramento, CA, April 1991.

[5] J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems*, 8(6):3–11, 1988.

[6] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, Heidelberg, Germany, 2 edition, 1989.

[7] F. L. Lewis, C. T. Abdallah, and D. M. Dawson. *Control of robot manipulators.* McMillan Publishing Co., New York, USA, 1993.

[8] D. G. Luenberger. *Optimization by Vector Space Methods.* Wiley, 1969.

[9] M. Mitsuishi. Diagnostic system for robot using a force-torque-sensor. *Robotersysteme*, 5:40–46, 1989.

[10] M. R. Napolitano, C.I. Chen, and S. Naylor. Aircraft failure detection and identification using neural networks. *AIAA J. Guidance Control Dynam.*, 16(6):999–1009, 1993.

[11] K. S. Narendra and A. M. Annaswamy. *Stable Adaptive Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[12] M. M. Polycarpou and A. J. Helmicki. Automated fault detection and accommoation: a learning control approach. *To appear in the IEEE Trans. on Systems, Man and Cybernetics*, 1995.

[13] A. Pouliezos, S. Tzafestas, and G. Stavrakakis. *System Fault Diagnostics,Reliability and Related Knowledge-Based Approaches*, volume 1, chapter : Fault detection and location in MRAC robotic systems., pages 203–217. D. Reidel Publishing Co., Dordrecht, Holland, 1986.

[14] J. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, Englewood Cliffs, NJ, 1991.

[15] T. J. Tarn, A. K. Bejczy, X. Yun, and Z. Li. Effect of motor dynamics on nonlinear feedback robot arm control. *IEEE Transactions on Robotics and Automation*, 7(1):114–121, 1991.

[16] Y. Ting, S. Tosunoglu, and R. Freeman. Actuator saturation avoidance for fault-tolerant robots. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2125–2130, San Antonio, TX, December 1993.

[17] I. Troch and K. Desoyer. IFAC theory committee report. Technical report, 1990.

[18] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker. Expert system framework for fault detection and fault tolerance in robotics. *Computers and Electrical Engineering*, 20(5):421–435, 1994.

[19] H.-J. Warnecke, W. Engeln, and K. Lehmann. Diagnosis for industrial robots. *Robotersysteme*, 6(2):65–75, 1990.

[20] K. Watanabe, I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau. Incipient fault diagnosis of chemical processes via artificial neural networks. *AIChe Journal*, 35:1803–1812, 1989.

[21] D. A. White and D. A. Sofge, editors. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* Van Nostrand and Reinhold, 1993.

[22] J. Wunnenberg and P. M. Frank. Dynamic model based incipient fault detection concept for robots. In *Proceedings of the IFAC 11th Triennial World Congress*, pages 61–66, Tallinn, Estonia, 1990.