VI. Conclusion

This paper presented a stabilizing RH controller for the regulation of nonholonomic mobile robots. The stability was guaranteed by forcing the terminal state to move into a terminal-state region and adding a stability term to the cost function. A Lyapunov-like function was developed as the stability term. A new terminal-state region and its corresponding terminal-state controller were found.

Besides the stability problem, the computation is also a main obstacle for using RH controllers in real-time systems. The measure taken in this paper was the use of "hot start" in the optimization computation.

Using the RH controller brings flexibility to controller design. As long as the stability can be guaranteed, different weight parameters in cost functions could result in different control performance. However, when the predictive control horizon is selected to be very long, the computation still causes a problem for real-time applications. How to improve the computation efficiency is currently under investigation, including using artificial neural networks [22] or binary decision trees [16].

References

[1] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robot. Autom. Mag.*, pp. 27–35, Mar. 1995.

[2] M. Alamir and H. Khennouf, "Discontinuous receding-horizon control-based stabilizing feedback for nonholonomic systems," in *Proc. 34th IEEE Conf. Decision Contr.*, New Orleans, LA, 1995, pp. 4300–4304.

[3] M. Alamir and N. Marchand, "Constrained minimum-time oriented feedback control for the stabilization of nonholonomic systems in chained forms," *J. Optim. Theory Appl.*, vol. 118, no. 2, pp. 229–244, 2003.

[4] A. Astolfi, "Discontinuous control of nonholonomic systems," *Syst. Control Lett.*, vol. 27, pp. 37–45, 1996.

[5] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, "Control and stablization of nonholonomic dynamic systems," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1746–1757, Nov. 1992.

[6] R. W. Brockett, "Asymptotic stability and feedback stabilization," *Differential Geom. Control Theory*, pp. 181–191, 1983.

[7] C. C. de Wit and O. J. Sørdalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1791–1797, Nov. 1992.

[8] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[9] G. De Nicolao, L. Magni, and R. Scattolini, "Stabilizing receding-horizon control of nonlinear time varyng systems," *IEEE Trans. Autom. Control*, vol. 43, no. 7, pp. 1030–1036, Jul. 1998.

[10] F. A. C. C. Fontes, "A general framework to design stabilizing nonlinear model predictive controllers," *Syst. Control Lett.*, vol. 42, no. 2, pp. 127–143, 2001.

[11] ——, "Discontinuous feedbacks, discontinuous optimal controls and continuous-time model predictive control," *Int. J. Robust Nonlinear Control*, vol. 13, no. 3–4, pp. 191–209, 2003.

[12] F. A. C. C. Fontes and L. Magni, "Min-max MPC of nonlinear systems using discontinuous feedbacks," *IEEE Trans. Autom. Control*, vol. 48, no. 10, pp. 1750–1755, Oct. 2003.

[13] G. Indiveri, "Kinematic time-invariant control of a 2D nonholonomic vehicle," in *Proc. 38th IEEE Conf. Decision Contr.*, 1999, pp. 2112–2117.

[14] A. Isidori, *Nonlinear Control Systems*, 3rd ed. New York: Springer-Verlag, 1995.

[15] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 46, no. 5, pp. 776–783, May 2001.

[16] T. A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 810–815, May 2003.

[17] S. S. Keerthi and E. G. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *J. Optim. Theory Appl.*, vol. 57, pp. 265–293, 1988.

[18] E. S. Meadows, M. A. Henson, J. E. Eaton, and J. B. Rawlings, "Receding horizon control and state feedback stabilization," *Int. J. Control*, vol. 62, no. 5, pp. 1217–1229, 1995.

[19] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[20] R. T. M'Closkey and R. M. Murray, "Exponential stabilization of driftless nonlinear control systems using homogeneous feedback," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 614–628, May 1997.

[21] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.

[22] T. Parisini and R. Zoppoli, "A receding-horizon regulator for nonlinear systems and a neural approximation," *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[23] J. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Syst. Control Lett.*, vol. 18, no. 2, pp. 147–158, 1992.

[24] J. Primbs, V. Nevistic, and J. Doyle, "A receding horizon generalization of pointwise min-norm controllers," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 898–909, May 2000.

[25] J. B. Rawlings and K. R. Muske, "Stability of constrained receding horizon control," *IEEE Trans. Autom. Control*, vol. 38, no. 10, pp. 1512–1516, Oct. 1993.

[26] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 55–64, 1993.

[27] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibilty implies stability)," *IEEE Trans. Autom. Control*, vol. 44, no. 3, pp. 648–654, Mar. 1999.

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

[29] H. Van Essen and H. Nijmeijer, "Nonlinear model predictive control of constrained mobile robot," in *Proc. Eur. Control Conf.*, 2001, pp. 1157–1162.

Fault Identification for Robot Manipulators

M. L. McIntyre, W. E. Dixon, D. M. Dawson, and I. D. Walker

***Abstract*—Several factors must be considered for robotic task execution in the presence of a fault, including: detection, identification, and accommodation for the fault. In this paper, a nonlinear observer is used to identify a class of actuator faults once the fault has been detected by some other method. Advantages of the proposed fault-identification method are that it is based on the nonlinear dynamic model of a robot manipulator (and hence, can be extended to a number of general Euler Lagrange systems), it does not require acceleration measurements, and it is independent from the controller. A Lyapunov-based analysis is provided to prove that the developed fault observer converges to the actual fault. Experimental results are provided to illustrate the performance of the identification method.**

***Index Terms*—Fault identification, nonlinear dynamics, robot manipulator.**

I. Introduction

Due to the sustained needs for robotic application in remote and hazardous environments, and with emerging applications in medicine

Manuscript received December 3, 2004. This paper was recommended for publication by Associate Editor R. Roberts and Editor H. Arai upon evaluation of the reviewers' comments. This work was supported in part under two DOC Grants, in part under an ARO Automotive Center Grant, in part under a DOE Contract, in part under a Honda Corporation Grant, and in part under a DARPA Contract. This paper was presented at the IEEE International Conference on Robotics and Automation, New Orleans, LA, April 2004.

M. L. McIntyre, D. M. Dawson, and I. D. Walker are with the Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: mmcinty@ces.clemson.edu; darren.dawson@ces.clemson.edu; ianw@ces.clemson.edu).

W. E. Dixon is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wdixon@ufl.edu).

Digital Object Identifier 10.1109/TRO.2005.851356

and bioengineering for the treatment of disease (often requiring patient–robot interfacing), robot reliability and fault tolerance have received significant interest. Several factors must be considered for robot operation in the presence of a fault. These factors include: detection of the fault; characterization, quantification, and identification of the fault; and then response to the fault by halting the system and/or accommodating for the fault (e.g., through a robust or adaptive controller or through system redundancy).

In [3], a model-based fault-detection approach was successfully demonstrated experimentally. This approach was based on the generation of residuals through a filtered torque estimate which does not rely upon the measurement of acceleration quantities. Adaptive and robust detection algorithms were also developed in [3] to take into account possible uncertainty in the robot parameters. A more recent adaptive fault detection and isolation scheme is presented in [2], where concurrent faults could be detected and isolated during the adaptation phase for the uncertainty in the dynamic system, based on the use of generalized momenta [1] and a suitable overparameterization.

Once the fault(s) has been detected, the next step in designing a fault-tolerant system is to identify the fault. Based on the desire to have fault-tolerant control, significant work has been focused on this topic. The different approaches can be determined by the manner in which the output residual signals are generated and if the focus is directed at linear or nonlinear systems. In general, there are two types of residual generators, structured and fixed directional [9]. For linear systems, the residuals have been derived in several ways, including: observer-based [6], parity relations [8], [10], eigenstructure assignment [17], and identification-based. Similar methods have been applied to nonlinear systems. Some of these approaches have focused on nonlinear observers [7]. Other approaches apply parity relations to the nonlinear problem [11], [12]. In [32], a fault detection and isolation architecture for nonlinear uncertain dynamics systems was presented. This approach relies on faults to be smooth functions and full state measurability is required; it uses a bank of nonlinear adaptive estimators, one for fault detection and approximation, the other for fault isolation. In [15], the authors consider fault detection and isolation in nonlinear Euler-Lagrange mechanical systems. The approach relies on faults acting as an additive effect on the system dynamics, where exact model knowledge and full state measurability is required. A nonlinear observer with linear error dynamics is used to generate the residual for the fault detection and isolation system. Some researchers have used other tools, such as fuzzy logic and artificial neural networks, to approximate the system model and/or to identify the fault [5], [18], [22], [24], [26], [30].

Various approaches have also been proposed for tolerating failures in robot manipulators. Most approaches have centered on the addition of some form of redundancy (e.g., in actuation [21], [28], joints [13], [16], [19], [23], sensors [25], [31], or software [27]), where the system degrades gracefully by using the redundant components. For example, if a manipulator is kinematically redundant, its end-effector task can often still be carried out by the surviving joints following a joint failure [4], [19].

The development in this paper leverages on the research presented in [3] to further develop a method for robot manipulator fault detection and identification. Specifically, a nonlinear fault observer (see [29]) and a filtered error signal are developed that do not rely upon the measurement of acceleration quantities. The fault observer enables the development of an estimate system which can be compared with the real system through the system states, $q(t)$ and $\dot{q}(t)$. The occurrence of a single or a concurrent fault(s) will result in a difference between the two systems, allowing instantaneous detection of the fault (see [3]). Then the fault observer asymptotically identifies the fault. A system supervisor could use this information along with knowledge of the system to determine specifically which fault has occurred.

II. System Model

The mathematical model for an n-degree-of-freedom (DOF) robotic manipulator is assumed to have the following form [3]:

$$\ddot{q}(t) + \bar{N}(q,\dot{q}) + u_{-1}(t - T_f)\bar{\zeta}(t) = \bar{\tau}(t) \tag{1}$$

where

$$\bar{N}(q,\dot{q}) \triangleq M^{-1}(q)N(q,\dot{q}) \tag{2}$$

$$\bar{\zeta}(t) \triangleq M^{-1}(q)\zeta(t) \tag{3}$$

$$\bar{\tau}(t) \triangleq M^{-1}(q)\tau(t). \tag{4}$$

In (1)–(4), $q(t)$, $\dot{q}(t)$, $\ddot{q}(t) \in \mathbb{R}^n$ denote the joint position, velocity, and acceleration, respectively, $M(q) \in \mathbb{R}^{n\times n}$ represents the positive-definite, symmetric inertia matrix, $N(q,\dot{q}) \in \mathbb{R}^n$ represents centripetal Coriolis, gravitational, and friction effects, $\zeta(t) \in \mathbb{R}^n$ represents a fault in the robot manipulator, $u_{-1}(t - T_f)$ represents a unit step function, T_f represents the time instant at which a fault occurs, and $\tau(t) \in \mathbb{R}^n$ represents the torque input vector. To further model the class of faults considered in this paper, $\zeta(t)$ can be isolated by rewriting (1) as follows [3]:

$$\zeta_i(t) = \tau_i(t) - \left[M(q)\ddot{q} + N(q,\dot{q})\right]_i \quad \forall t \geq T_f. \tag{5}$$

Hence, a *locked-joint* fault is characterized by (5) and a *free-swinging* actuator fault (i.e., $\left[M(q)\ddot{q} + N(q,\dot{q})\right]_i = 0$), a *ramp* actuator fault (i.e., $\left[M(q)\ddot{q} + N(q,\dot{q})\right]_i = (\gamma_0)_i t$), and a *saturated* actuator fault (i.e., $\left[M(q)\ddot{q} + N(q,\dot{q})\right]_i = (\tau_{\max})_i$) are characterized as [3]

$$\zeta_i(t) = \begin{cases} \tau_i(t) \\ \tau_i(t) - \gamma_{0_i} t & \forall t \geq T_f \\ \tau_i(t) - (\tau_{\max})_i \end{cases} \tag{6}$$

respectively, where $\tau_i(t)$ is the applied torque at joint i, $\gamma_{0_i} \in \mathbb{R}$ is a positive scaling term per joint i, and $\tau_{\max} \in \mathbb{R}^n$ represents a vector of maximum torques that can be applied by the actuators.

The dynamic equation in (1) exhibits the following property which is used in conjunction with the following assumptions in the subsequent development.

Assumption 1: The following upper bounds exist:

$$\left\|\bar{\zeta}(t)\right\| \leq \gamma_1, \quad \left\|\dot{\bar{\zeta}}(t)\right\| \leq \gamma_2, \quad \left\|\ddot{\bar{\zeta}}(t)\right\| \leq \gamma_3 \tag{7}$$

for some finite value of time (clearly, from (6), the ramp fault cannot be bounded as $t \to \infty$), where $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{R}$ are positive bounding constants.

Assumption 2: A continuous control is designed which ensures that $q(t), \dot{q}(t), \tau(t) \in \mathcal{L}_\infty \ \forall t$. It should be noted that based on the form of the dynamic model given in (1), if $q(t), \dot{q}(t), \tau(t) \in \mathcal{L}_\infty$, then from *Assumption 1*, it is clear that $\ddot{q}(t) \in \mathcal{L}_\infty$.

Remark 1: The bounds given in (7) and *Assumption 2* have both a practical and mathematical basis [the subsequent stability analysis requires (7)]. These assumptions require a smooth transition from the working (nonfault) condition to the fault condition, which for the class of faults in question appears to be a reasonable requirement. Examining the *free-swinging* actuator fault as described by (6), when the fault occurs, the particular joint under control is either static or in motion. After the fault, the torque on that joint is removed and the natural forces (i.e., gravity, potential energy, etc.) at work on the joint may or may not change the course of motion for the affected links, but at no time during the transition are there any discontinuous activities associated with the system states as a result of the fault. Similar statements could be made

for the three other faults discussed in this paper. Assumptions of this nature have also been made in [2] and [32] when dealing with the continuous nature of robot dynamics and faults.

III. Fault Identification

Once a fault has been detected (i.e., $\forall t \geq T_f$), additional knowledge regarding the fault may be required (e.g., to make decisions regarding the continued operation of the system, to facilitate a fault-accommodation scheme). To facilitate the development of the fault-identification scheme, a velocity observer error signal, denoted by $e(t) \in \mathbb{R}^n$, is defined as follows:

$$e(t) \triangleq \dot{\hat{q}}(t) - \dot{q}(t) \tag{8}$$

where $\dot{\hat{q}}(t) \in \mathbb{R}^n$ denotes the following velocity estimate:

$$\dot{\hat{q}}(t) \triangleq -\int_{T_f}^{t} \bar{N}(q(\sigma), \dot{q}(\sigma))d\sigma - \int_{T_f}^{t} \hat{\zeta}(\sigma)d\sigma + \int_{T_f}^{t} \bar{\tau}(\sigma)d\sigma + \dot{\hat{q}}(T_f) \quad \forall t \geq T_f \tag{9}$$

where $\hat{\zeta}(t) \in \mathbb{R}^n$ denotes a subsequently designed nonlinear fault observer. The time derivative of (8) is expressed as follows:

$$\dot{e}(t) = \ddot{\hat{q}}(t) - \ddot{q}(t) \tag{10}$$

where

$$\ddot{\hat{q}}(t) = -\bar{N}(q, \dot{q}) - \hat{\zeta}(t) + \bar{\tau}(t) \quad \forall t \geq T_f. \tag{11}$$

After substituting (1) and (11) into (10), the following simplified expression can be obtained:

$$\dot{e}(t) = \tilde{\zeta}(t) \quad \forall t \geq T_f \tag{12}$$

where $\tilde{\zeta}(t) \in \mathbb{R}^n$ is defined as

$$\tilde{\zeta}(t) \triangleq \bar{\zeta}(t) - \hat{\zeta}(t). \tag{13}$$

Based on (12) and the subsequent stability analysis (see Section IV), the following proportional-integral-like nonlinear observer is developed to identify the fault:

$$\hat{\zeta}(t) \triangleq \int_{T_f}^{t} (K_0 + I_n)e(\sigma)d\sigma + \int_{T_f}^{t} K_1 \mathrm{sgn}(e(\sigma))d\sigma + (K_0 + I_n)e(t) - (K_0 + I_n)e(T_f) \tag{14}$$

where K_0 and $K_1 \in \mathbb{R}^{n \times n}$ represent constant, diagonal, positive-definite observer gain matrices, where $\mathrm{sgn}(\cdot)$ denotes the vector signum function, and $I_n \in \mathbb{R}^{n \times n}$ is the standard identity matrix. From (8) and (14), it is clear that the fault observer does not depend on acceleration measurements.

To facilitate the subsequent analysis, an auxiliary error signal, denoted by $s(t) \in \mathbb{R}^n$, is defined as follows:

$$s(t) \triangleq \dot{e}(t) + e(t). \tag{15}$$

The time derivative of (15) can be determined as

$$\dot{s}(t) = \dot{\tilde{\zeta}}(t) - (K_0 + I_n)s(t) - K_1 \mathrm{sgn}(e(t)) + \dot{e}(t) \quad \forall t \geq T_f \tag{16}$$

where the time derivative of (12) was used along with (13), (14) and (15).

IV. Analysis

Theorem 1: The fault observer given in (14) ensures that

$$\hat{\zeta}_{fi}(t) \to \bar{\zeta}_{fi}(t) \text{ as } t \to \infty \tag{17}$$

provided the observer gain K_1 introduced in (14) is selected as follows:

$$K_{1_i} > \left|\dot{\bar{\zeta}}_i(t)\right| + \left|\ddot{\bar{\zeta}}_i(t)\right| \tag{18}$$

where the subscript $i = 1, 2, \ldots, n$ denotes the ith element of the vector or diagonal matrix.

Proof: Let $V(t) \in \mathbb{R}$ denote the following nonnegative function:

$$V = \frac{1}{2}e^T e + \frac{1}{2}s^T s \tag{19}$$

where $e(t)$ and $s(t)$ were defined in (8) and (15), respectively. After taking the time derivative of (19) and using (15) and (16), the following expression can be obtained:

$$\dot{V}(t) = e^T(s - e) + s^T\left(\dot{\tilde{\zeta}} - (K_0 + I_n)s - K_1 \mathrm{sgn}(e) + \dot{e}\right) \quad \forall t \geq T_f. \tag{20}$$

The following simplified expression can then be obtained after using (15):

$$\dot{V}(t) = -e^T e - s^T K_0 s + \dot{e}^T \dot{\tilde{\zeta}} + e^T \dot{\tilde{\zeta}} - (\dot{e}(t) + e(t))^T K_1 \mathrm{sgn}(e) \quad \forall t \geq T_f. \tag{21}$$

The integral of (21) from T_f to t can be expressed as

$$\begin{aligned} V(t) \leq V(T_f) &- \int_{T_f}^{t} \|e(\sigma)\|^2 d\sigma \\ &- \lambda_{\min}\{K_0\} \int_{T_f}^{t} \|s(\sigma)\|^2 d\sigma + \int_{T_f}^{t} \dot{e}^T(\sigma)\dot{\tilde{\zeta}}(\sigma)d\sigma \\ &- \int_{T_f}^{t} \dot{e}^T(\sigma)K_1 \mathrm{sgn}(e(\sigma))d\sigma \\ &+ \int_{T_f}^{t} e^T(\sigma)\left(\dot{\tilde{\zeta}}(\sigma) - K_1 \mathrm{sgn}(e(\sigma))\right)d\sigma. \end{aligned} \tag{22}$$

After integrating the fourth term on the right-hand side (RHS) of (22) by parts and integrating the fifth term on the RHS of (22) with respect to time, the following expression is obtained for $V(t)$:

$$\begin{aligned} V(t) \leq V(T_f) &- \int_{T_f}^{t} \|e(\sigma)\|^2 d\sigma + e^T(t)\dot{\tilde{\zeta}}(t) \\ &- \lambda_{\min}\{K_0\} \int_{T_f}^{t} \|s(\sigma)\|^2 d\sigma - e^T(T_f)\dot{\tilde{\zeta}}(T_f) \\ &+ e^T(T_f)K_1 \mathrm{sgn}(e(T_f)) - e^T(t)K_1 \mathrm{sgn}(e(t)) \\ &+ \int_{T_f}^{t} e^T(\sigma)\left[\dot{\tilde{\zeta}}(\sigma) - \ddot{\bar{\zeta}}(\sigma) - K_1 \mathrm{sgn}(e(\sigma))\right]d\sigma. \end{aligned}$$

Provided K_1 is selected according to (18), $V(t)$ can be further upper bounded as follows:

$$V(t) \leq -\lambda_{\min}\{K_0\} \int_{T_f}^{t} \|s(\sigma)\|^2 d\sigma - \int_{T_f}^{t} \|e(\sigma)\|^2 d\sigma + C \tag{23}$$

were $C \in \mathbb{R}$ represents the following positive bounding constant:

$$C \triangleq V(T_f) - e^T(T_f)\left(\dot{\tilde{\zeta}}(T_f) - K_1 \mathrm{sgn}(e(T_f))\right). \tag{24}$$

From the structure of (23) and the definition in (24), it is clear that $V(t) \in \mathcal{L}_\infty$; hence, $s(t)$, $e(t) \in \mathcal{L}_\infty$. Since $s(t)$, $e(t) \in \mathcal{L}_\infty$, (15)

can be used to prove that $\dot{e}(t) \in \mathcal{L}_\infty$. Based on the assumption that $\dot{\bar{\zeta}}(t) \in \mathcal{L}_\infty$ (*Assumption 2*) and the fact that $s(t), e(t), \dot{e}(t) \in \mathcal{L}_\infty$, (16) can be used to prove that $\dot{s}(t) \in \mathcal{L}_\infty$. The inequality in (23) can also be used to show that $s(t), e(t) \in \mathcal{L}_2$. Since $s(t), \dot{s}(t), e(t), \dot{e}(t) \in \mathcal{L}_\infty$, and $s(t), e(t) \in \mathcal{L}_2$, then Barbalat's Lemma can be used to conclude that

$$\lim_{t \to \infty} \|s(t)\|, \|e(t)\| = 0. \tag{25}$$

Based on (12), (13), and (25), the result in (17) can be obtained. □

V. Experimental Verification

The fault-identification strategy given in Section III was implemented on a Barrett Whole Arm Manipulator (WAM). This strategy requires a method of fault detection for notification that a fault has occurred, hence $t \geq T_f$. The fault-detection method used in [3] was implemented on the WAM, and integrated as part of a fault detection and identification strategy. The following sections describe the experimental testbed and results obtained for several classes of injected faults.

A. Experimental Testbed

A fault detection and identification strategy was implemented on the WAM. For simplicity, five links of the WAM were locked at a fixed, specified angle, resulting in a two-link planar manipulator (see Fig. 1). Henceforth, link 1 is used to denote the base link of the manipulator, and link 2 is used to denote the other actuated link. In this configuration, the dynamics of the robot can be expressed as follows [20]:

$$\tau = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} V_{m11} & V_{m12} \\ V_{m21} & V_{m22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} f_{d_1} & 0 \\ 0 & f_{d_2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} f_{s_1} & 0 \\ 0 & f_{s_2} \end{bmatrix} \begin{bmatrix} \operatorname{sgn}(\dot{q}_1) \\ \operatorname{sgn}(\dot{q}_2) \end{bmatrix}. \tag{26}$$

In (26), the elements of the inertia and centripetal-Coriolis matrices are given as

$$\begin{aligned} M_{11} &= p_1 + 2p_2 \cos(q_2) \\ M_{12} &= p_3 + p_2 \cos(q_2) \\ M_{21} &= p_3 + p_2 \cos(q_2) \\ M_{22} &= p_3 \\ V_{M_{11}} &= -p_2 \sin(q_2)\dot{q}_2 \\ V_{M_{12}} &= -p_2 \sin(q_2)\dot{q}_1 - p_2 \sin(q_2)\dot{q}_2 \\ V_{M_{21}} &= p_2 \sin(q_2)\dot{q}_1 \\ V_{M_{22}} &= 0 \end{aligned}$$

where $p_1 = 1.35$ kg·m^2, $p_2 = 0.0529$ kg·m^2, and $p_3 = 0.2502$ kg·m^2, the constant dynamic friction coefficients are $f_{d_1} = 0.8838$ N·m·s and $f_{d_2} = 0.66285$ N·m·s, the constant static friction coefficients are $f_{s_1} = 1.50246$ N·m and $f_{s_2} = 0.7074$ N·m. The gravitational effects are not included in (26) due to the plane of motion of the manipulator.

The links of the WAM manipulator are driven by brushless motors supplied with sinusoidal electronic commutation. Each axis has encoders located at the motor shaft for link position measurements. Since no tachometers are present for velocity measurements, link velocity signals are calculated via a filtered backward difference algorithm. An AMD Athlon 1.2 GHz PC operating QNX 6.2.1 RTP (Real Time Platform), a realtime microkernel-based operating system, hosts the control, detection, and identification algorithms which were written in "C++". Qmotor 3.0 [14], was used to facilitate realtime graphing, data logging, and online gain adjustment. Data acquisition and control implementation were performed at a frequency of 1.0 kHz using the ServoToGo I/O board.

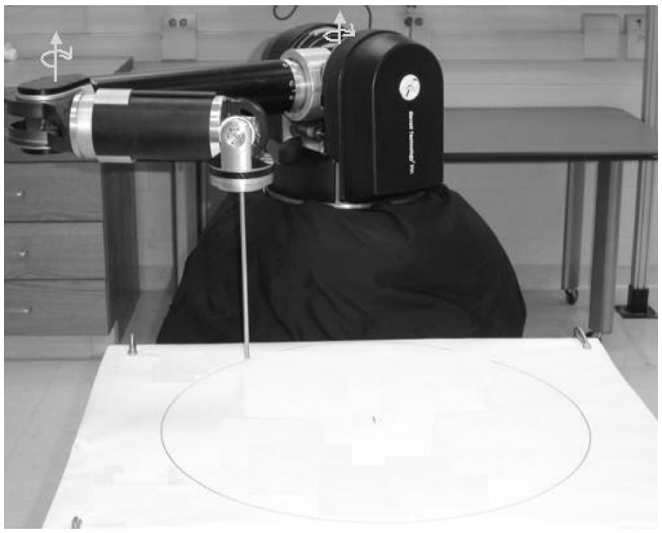

Fig. 1. Planar, two-link configuration of WAM.

B. Experimental Results

A standard proportional derivative control scheme with a feedforward desired acceleration component was used to ensure that $q(t)$ tracks $q_d(t)$ where $q_d(t)$ is defined as follows:

$$\begin{bmatrix} q_{d_1}(t) \\ q_{d_2}(t) \end{bmatrix} = \begin{bmatrix} 0.8\sin(t) \\ 0.8\sin(t) \end{bmatrix}. \tag{27}$$

Since the objective in this experiment is to identify the fault signature, the *ad hoc* threshold method as seen in [3, eq. (26), Remark 5] was implemented. Specifically, the fault indicating *dead-zone* residual function can be defined as follows:

$$D[\varepsilon_i] = \begin{cases} |\varepsilon_i|, & \text{if } |\varepsilon_i| > (\mu_o)_i \\ 0, & \text{if } |\varepsilon_i| \leq (\mu_o)_i \end{cases} \tag{28}$$

where $\mu_{o1} = 0.0375$ and $\mu_{o2} = 0.0132$ denotes positive, scalar constants that were experimentally determined to account for small uncertainties and measurement noise [3]. For fault detection, the filter parameters were selected as [3]

$$\alpha = 1, \quad \beta = 150 \tag{29}$$

and the fault-identification gains as defined in (14) were selected as shown below

$$K_0 = \operatorname{diag}\{80{,}80\}, \quad K_1 = \operatorname{diag}\{100{,}100\} \tag{30}$$

where $\operatorname{diag}\{\cdot,\cdot\}$ represent the elements along the diagonal for K_0 and K_1. All parameters were tuned by trial and error until reasonable results were observed.

To demonstrate the performance of the fault detection and identification system using the WAM, a *free-swinging* actuator fault, a *ramp* actuator fault (with $\gamma_{0_1} = 0.048$ and $\gamma_{0_2} = 0.022$), and a *saturation* actuator fault (with the maximum torque artificially limited to $\tau_{\max_1} = 0.838$ N·m and $\tau_{\max_2} = 0.66$ N·m for safety) were injected at $T_f = 30$ s for link 1 and at $T_f = 40$ s for link 2, respectively. Figs. 2–4 illustrate the performance of the fault-detection algorithm. From these plots, the fault indicating *dead-zone* residual function as defined in (28) is shown for links 1 and 2, also the threshold for μ_{o1} and μ_{o2} can be seen. In these figures, the marker (a) represents the residual

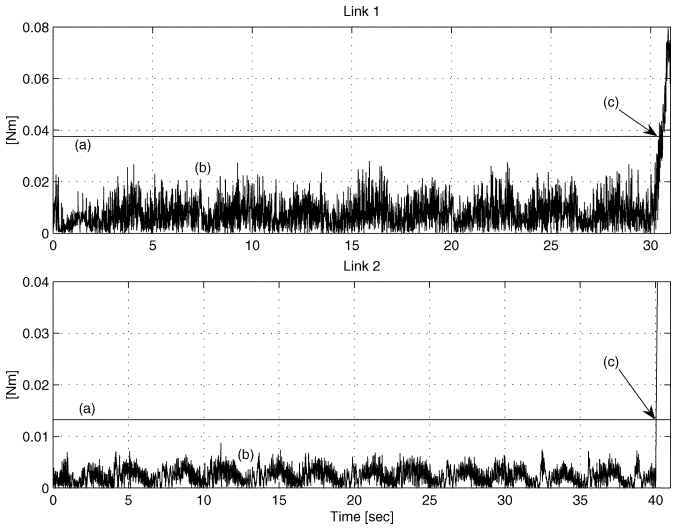

Fig. 2. Fault detection for a *free-swinging* fault. Markers: (a) represents the residual thresholds; (b) indicates the prediction errors; and (c) indicates the instants the faults are detected.

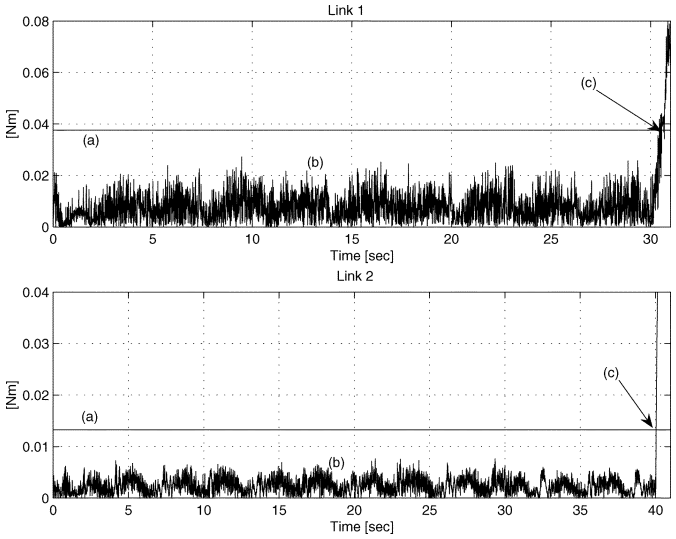

Fig. 3. Fault detection for a *ramp* fault. Markers: (a) represents the residual thresholds; (b) indicates the prediction errors; and (c) indicates the instants the faults are detected.

threshold, the marker (b) indicates the prediction error, and the marker (c) indicates the instant the fault is detected, which is achieved when the fault indicating *dead-zone* residual function breaks the respective threshold. This *ad hoc* threshold method for fault detection worked perfectly without any false alarms or missed detections. The robustness of this method comes from the ability to set the detection thresholds, μ_{o1} and μ_{o2}, for each experimental testbed. See [3] for additional experimental results using alternate fault-detection methods.

Figs. 5–7 illustrate the performance of the nonlinear observer. From these plots, the fault-identification error $\dot{e}(t) = \tilde{\zeta}(t)$ is shown to decrease toward zero, indicating identification of the fault. Figs. 8–10 illustrate the identified fault signature $\hat{\zeta}(t)$ for each joint. In the event of concurrent faults, the resultant fault signature, $\hat{\zeta}(t)$, is the summation of both faults. Future efforts will target exploring the fault signatures to further characterize the fault, leading to potential fault accommodation. Future work may include the use of estimation theory to properly account for sensor noise that can be seen in Figs. 5–10.

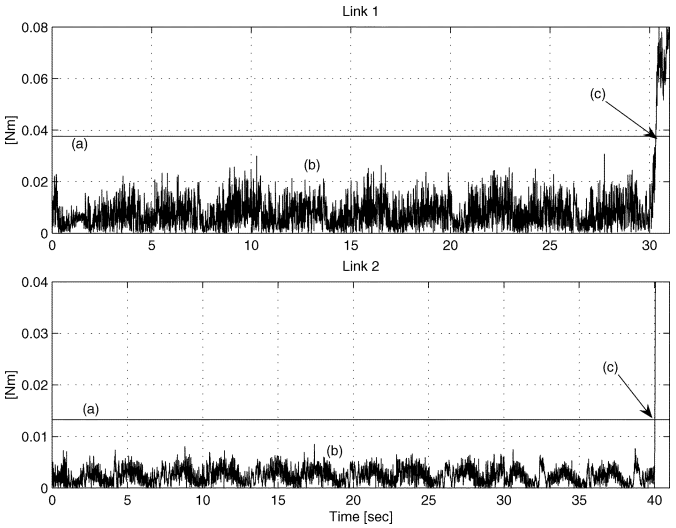

Fig. 4. Fault detection for a *saturated* fault. Markers: (a) represents the residual thresholds; (b) indicates the prediction errors; and (c) indicates the instants the faults are detected.

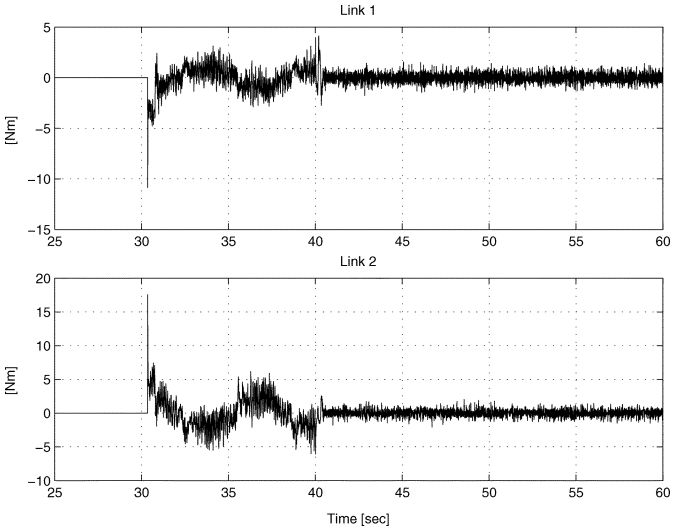

Fig. 5. Fault-identification error $\tilde{\zeta}(t)$ for a *free-swinging* fault.

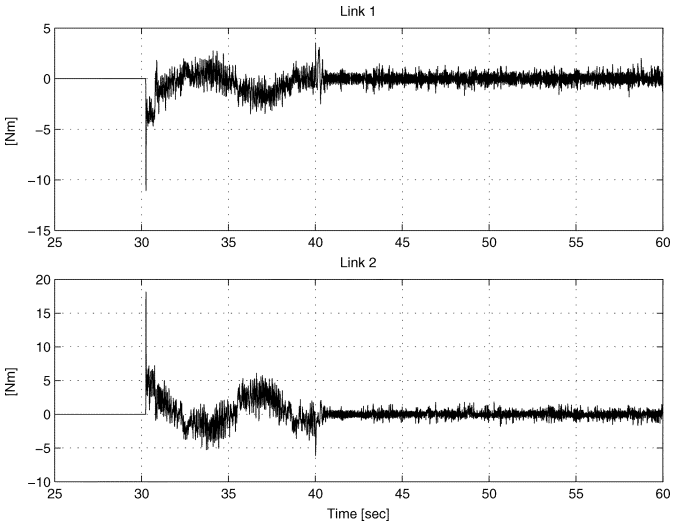

Fig. 6. Fault-identification error $\tilde{\zeta}(t)$ for a *ramp* fault.

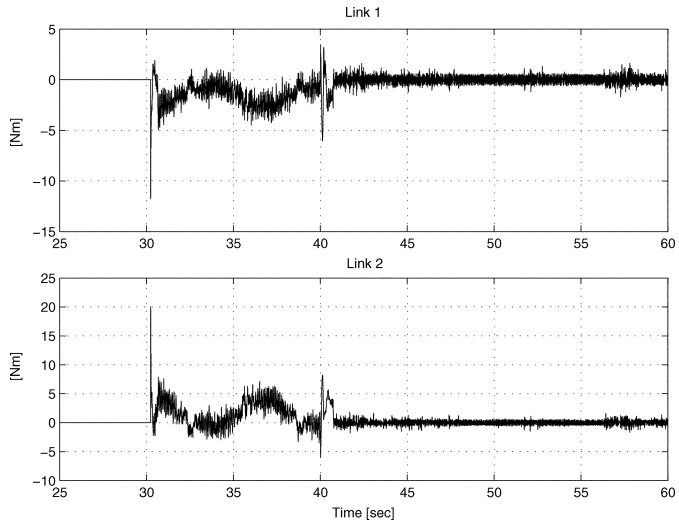

Fig. 7. Fault-identification error $\bar{\zeta}(t)$ for a *saturated* fault.

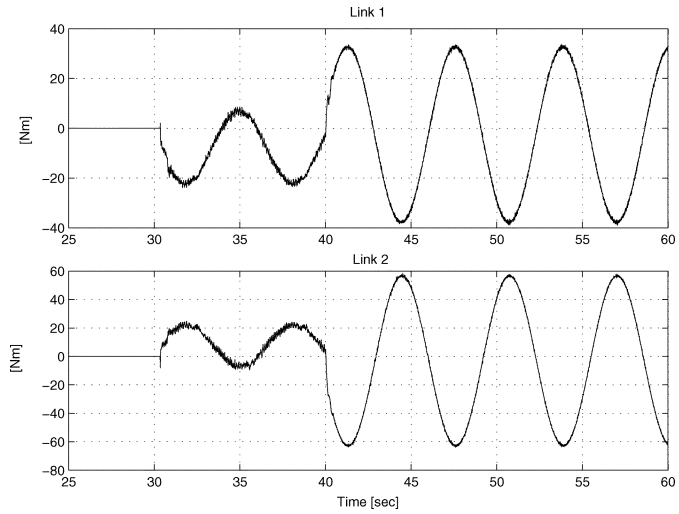

Fig. 8. Fault signature $\hat{\zeta}(t)$ for a *free-swinging* fault.

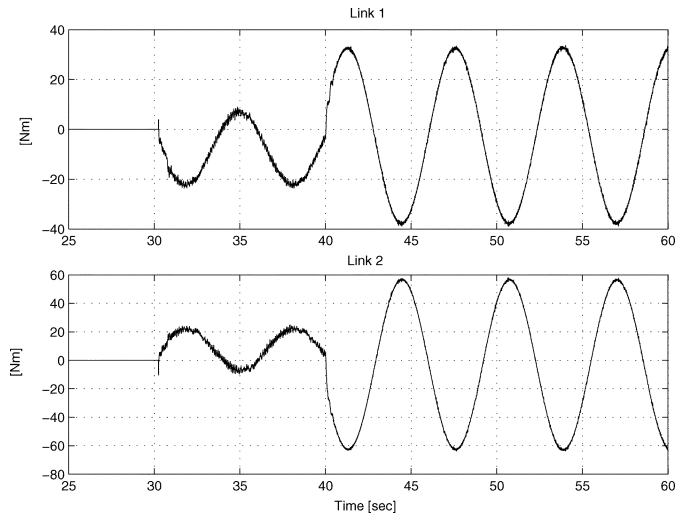

Fig. 9. Fault signature $\hat{\zeta}(t)$ for a *ramp* fault.

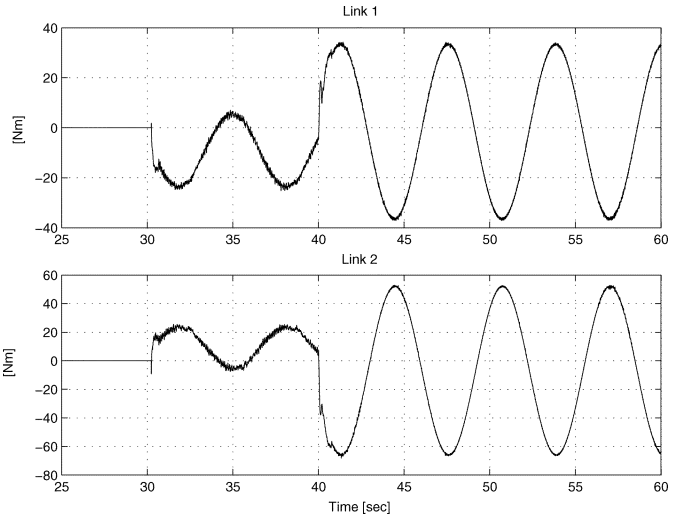

Fig. 10. Fault signature $\hat{\zeta}(t)$ for a *saturated* fault.

VI. Conclusion

In conclusion, a fault-identification method is proposed for robot manipulators. This method is based on the nonlinear dynamics of the robot model, it does not require acceleration measurements, and is independent from the controller. The fault-identification scheme can be applied to a generic class of actuator faults that are second-order differentiable. The effectiveness of the proposed fault-identification method is illustrated through experimental results.

References

[1] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, 2003, pp. 634–639.

[2] ——, "An adapt-and-detect actuator FDI scheme for robot manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, 2004, pp. 4975–4980.

[3] W. E. Dixon, I. D. Walker, D. M. Dawson, and J. P. Hartranft, "Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 689–699, Dec. 2000.

[4] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Minneapolis, MN, 1996, pp. 460–467.

[5] J. A. Farrell, T. Berger, and B. D. Appleby, "Using learning techniques to accommodate unanticipated faults," *IEEE Control Syst. Mag.*, vol. 13, pp. 40–49, 1993.

[6] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy, a survey and some new results," *Automatica*, vol. 26, pp. 459–474, 1990.

[7] P. M. Frank and R. Seliger, "Fault detection and isolation in automatic processes," in *Control Dynamic Systems*, C. Leondes, Ed. New York: Academic, 1991, pp. 241–287.

[8] J. J. Gertler and M. M. Kunwer, "Optimal residual decoupling for robust fault diagnosis," *Int. J. Control*, vol. 61, no. 2, pp. 395–421, 1995.

[9] J. J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.

[10] ——, "Analytical redundancy methods in fault detection and isolation," in *Proc. IFAC Symp. Fault Detection, Supervision Safety Tech. Process.*, Baden, Germany, 1991, p. 921.

[11] V. Krishnaswami and G. Rizzoni, "Nonlinear parity equation residual generation for fault detection and isolation," in *Proc. IFAC Symp. Fault Detection, Supervision Safety Tech. Process.*, Copenhagen, Denmark, 1994, pp. 317–322.

[12] ——, "Robust residual generation for nonlinear system fault detection and isolation," in *Proc. IFAC Symp. Fault Detection, Supervision Safety Tech. Process.*, Copenhagen, Denmark, 1994, pp. 163–168.

[13] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of joint failures," *Comput. Elect. Eng.*, vol. 20, no. 3, pp. 273–288, 1994.

[14] M. Loffler, N. Costescu, and D. Dawson, "Qmotor 3.0 and the Qmotor robotic toolkit—An advanced PC-based real-time control platform," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 12–26, Jun. 2002.

[15] R. Mattone and A. De Luca, "Fault detection and isolation in robot manipulators," IFATIS, IRAR002R01, 2003.

[16] C. J. Paredis, W. K. F. Au, and P. K. Khosla, "Kinematic design of fault tolerant manipulators," *Comput. Elect. Eng.*, vol. 20, no. 3, pp. 211–220, 1994.

[17] R. J. Patton, "Robust fault detection using eigenstructure assignment," in *Proc. 12th IMACS World Congr. Math. Modeling Sci. Computat.*, Paris, France, 1988, pp. 431–433.

[18] M. M. Polycarpou and A. J. Helmicki, "Automated fault detection and accommodation: A learning systems approach," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 11, pp. 1447–1458, Nov. 1995.

[19] J-H. Shin and J-J. Lee, "Fault detection and robust fault recovery control for robot manipulators with actuator failures," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 861–866.

[20] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

[21] D. Sreevijayan, S. Tosunoglu, and D. Tesar, "Architectures for fault tolerant mechanical systems," in *Proc. IEEE Mediterranean Electrotech. Conf.*, Antalya, Turkey, 1994, pp. 1029–1033.

[22] M. H. Terra and R. Tinos, "Fault detection and isolation in robotic systems via artificial neural networks," in *Proc. 37th IEEE Conf. Decision Control*, vol. 2, Tampa, FL, 1998, pp. 1605–1610.

[23] Y. Ting, S. Tosunoglu, and D. Tesar, "A control structure for fault tolerant operation of robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Atlanta, GA, 1993, pp. 684–690.

[24] A. B. Trunov and M. M. Polycarpou, "Automated fault diagnosis in nonlinear multivariable systems using a learning methodology," *IEEE Trans. Neural Netw.*, vol. 11, no. 1e, pp. 91–101, Jan. 2000.

[25] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 4, pp. 477–490, Aug. 1995.

[26] H. Wang, Z. J. Huang, and S. Daley, "On the use of adaptive updating rules for actuator and sensor fault diagnosis," *Automatica*, vol. 33, pp. 217–225, 1997.

[27] T. S. Wikman, M. S. Branicky, and W. S. Newman, "Reflex control for robot system preservation, reliability, and autonomy," *Comput. Elect. Eng.*, vol. 20, no. 5, pp. 391–407, 1994.

[28] E. C. Wu, J. C. Hwang, and J. T. Chladek, "Fault tolerant joint development for the space shuttle remote manipulator system: Analysis and development," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 675–684, Oct. 1993.

[29] B. Xian, M. S. de Queiroz, and D. Dawson, "A continuous control mechanism for uncertain nonlinear systems," in *Optimal Control, Stabilization, and Nonsmooth Analysis*. Heidelberg, Germany: Springer-Verlag, 2004, Lecture Notes in Control and Information Sciences.

[30] G. Yen and L. Ho, "Fault tolerant control: An intelligent sliding model control strategy," in *Proc. Amer. Control Conf.*, Chicago, IL, 2000, pp. 4204–4208.

[31] F. Zanaty, "Consistency checking techniques for the space shuttle remote manipulator system," *SPAR J. Eng. Technol.*, vol. 2, no. 1, pp. 40–49, 1993.

[32] X. Zhang, M. Polycarpou, and T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems," *IEEE Trans. Autom. Control*, vol. 47, no. 4, pp. 576–593, Apr. 2002.