

Fault Detection for Robot Manipulators with Parametric Uncertainty: A Prediction-Error-Based Approach

Warren E. Dixon, *Member, IEEE*, Ian D. Walker, *Member, IEEE*, Darren M. Dawson, *Senior Member, IEEE*, and John P. Hartranft, *Member, IEEE*

Abstract—In this paper, we introduce a new approach to fault detection for robot manipulators. The technique, which is based on the isolation of fault signatures via filtered torque prediction error estimates, does not require measurements or estimates of manipulator acceleration as is the case with some previously suggested methods. The method is formally demonstrated to be robust under uncertainty in the robot parameters. Furthermore, an adaptive version of the algorithm is introduced, and shown to both improve coverage and significantly reduce detection times. The effectiveness of the approach is demonstrated by experiments with a two-joint manipulator system.

Index Terms—Fault detection, nonlinear dynamics, prediction error, robot manipulator.

I. INTRODUCTION

IN THE past few years, as the application of robots has expanded, there has been significant activity in the area of robot reliability and fault tolerance [19]. Fault tolerance is especially important in remote and hazardous environments, such as found in space, underwater, and radioactive applications (where repair is often infeasible, and failure can have disastrous consequences), although reliability and safety are important issues in almost all applications.

Various approaches to tolerating failures in robot manipulators have been proposed recently. Most have centered on the addition of some form of redundancy, for example, in actuation [15], [23], joints [7], [10], [13], [16], sensors [20], [25], or software [22], to provide “backup” beyond the core requirements of the system. Given the detection of a fault, the system degrades gracefully by using the “backup” 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 [5], [13]. Similarly, redundant sensing allows a system to switch to “healthy” sensors following a sensor failure [14].

Manuscript received February 2, 2000; revised August 29, 2000. This paper was recommended for publication by Associate Editor J. Yuh and Editor V. Lumelsky upon evaluation of the reviewers’ comments. This paper was presented in part at the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000.

W. E. Dixon was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634-0915 USA. He is now with the Oak Ridge National Laboratory, Knoxville, TN 37909 USA.

I. D. Walker, D. M. Dawson, and J. P. Hartranft are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634-0915 USA (e-mail: ianw@ces.clemson.edu).

Publisher Item Identifier S 1042-296X(00)11402-8.

However, such redundancy can only be exploited if fault detection is effective, enabling the system to reconfigure and use “healthy” components. Robot fault detection has therefore become an issue of significant interest recently, with several new approaches suggested for mobile robot fault detection [11], [14] (see below).

In [25], the use of redundant sensors on each joint of the Space Shuttle Remote Manipulator System (RMS) was described. Faults were inferred if the sensors disagreed significantly with the prescribed trajectory or each other. A simple thresholding scheme was used to infer when a fault has occurred (i.e., when disagreement was “significant”). However, selection of numerical values for the thresholds proved difficult, and inappropriate choices led to false alarms during missions [25].

The problems described in [25] illustrate the key difficulty endemic to manipulator fault detection. The normal (fault-free) dynamics of the robot lead to inevitable deviations from the nominal trajectory in fault-free operation, and the magnitude of these deviations cannot be predicted (and therefore can appear to be a fault unless properly masked by the thresholds), when the dynamics are not explicitly considered in the analysis, as was the case in [25]. Clearly, fault detection will be most effective when good dynamic models for the manipulator are considered in the fault detection tests (residuals) or the threshold selection, or both. In [13], a robust tracking controller/fault detection scheme was proposed that utilized the full dynamic model of the robot manipulator; unfortunately, the fault detection residuals are based on conservative thresholds, that are obtained by taking the norm of user defined upper bounds for the position and velocity tracking errors.

A dynamics-based approach was adopted in [6], where faults were inferred for a standard industrial robot by monitoring sudden changes in a vector of on-line parameter estimates for the robot. The method was shown to be effective for certain types of faults. However the underlying dynamic model was highly simplified (constant inertias and coupling between joints was neglected), which implies again the need for either conservative thresholds, or probable false alarms.

A more rigorous approach to the synthesis of fault detection residuals was presented in [20], in which the theoretical maximum number of independent residuals were derived for a manipulator with redundant sensing, based on linearized dynamics for the robot. Dynamic thresholds were developed based on full (nonlinear) manipulator dynamics. The results were promising, however the thresholds required the measurement or estimation

of manipulator acceleration which is problematic at best in practice. That is, for most practical applications the manipulator acceleration is not directly measured, rather it is numerically calculated from position or velocity signals, and hence, the signal is inherently noisy.

Other approaches to manipulator fault detection have included the development of observers [1], [11], [24] for residual generation. However, the generation of appropriate thresholds for the observer-based residuals (an issue not discussed in [24]) remains an issue—for example, in [1], the estimation or measurement of joint accelerations is still required in order to mask the effects of disturbances due to parametric errors in the dynamics. In [12], a fuzzy logic approach is used to allow for such disturbances, however the approach remains somewhat heuristic. A neural network approach to manipulator fault detection was adopted in [2] and [17]; however, the fault detection algorithms are based on a user defined bound on the modeling uncertainty. We also note that other on-line estimation fault detection approaches are proposed in [18], [21], and [26].

In this paper, we build on the initial research presented in [4] to further develop a new method for robot manipulator fault detection. The approach is based on the generation of residuals and exploits the structure of the full nonlinear manipulator dynamics through a filtered torque estimate that does not rely upon the measurement of acceleration quantities. Furthermore, the structure of the algorithm lends itself to adaptive and robust versions to take into account the inevitable uncertainty in the robot parameters. New thresholds for the residuals for all the above cases are developed, and the effectiveness of the approach is demonstrated through experiments with a two joint robot system.

II. DYNAMIC MODEL

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

$$M(q)\ddot{q} + N(q, \dot{q}) + u_{-1}(t - T)\zeta(t) = \tau \quad (1)$$

where $q(t)$, $\dot{q}(t)$, $\ddot{q}(t) \in \mathbb{R}^n$ denote the link 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, $u_{-1}(t - T)$ represents a unit step function, T represents the time instant at which a fault occurs, and $\tau(t) \in \mathbb{R}^n$ represents the torque input vector. In order to further model the class of faults considered in this paper, we can isolate $\zeta(t)$ by rewriting (1) as follows:

$$\zeta_i(t) = \tau_i - [M(q)\ddot{q} + N(q, \dot{q})]_i \quad \forall t \geq T. \quad (2)$$

Hence, a *locked-joint* fault is characterized by (2) and a *free-swinging* actuator fault (i.e., $[M(q)\ddot{q} + N(q, \dot{q})]_i = 0$), a *ramp* actuator fault (i.e., $[M(q)\ddot{q} + N(q, \dot{q})]_i = (\gamma_1)_i t$), and a *saturated* actuator fault (i.e., $[M(q)\ddot{q} + N(q, \dot{q})]_i = (\tau_{\max})_i$) are characterized as

$$\zeta_i(t) = \begin{cases} \tau_i \\ \tau_i - \gamma_1 t \\ \tau_i - (\tau_{\max})_i \end{cases} \quad \forall t \geq T \quad (3)$$

respectively, where τ_i is the applied torque at joint i , $\gamma_1 \in \mathbb{R}^1$ is a positive scaling term, and $\tau_{\max} \in \mathbb{R}^n$ represents a vector of maximum torques that can be applied by the actuators.

The dynamic equation given in (1), exhibits the following property [8] which is utilized in conjunction with the following assumptions in the subsequent fault detection algorithm development.

Property 1: The robot dynamics given in (1) can be linearly parameterized as follows:

$$Y(q, \dot{q}, \ddot{q})\theta = M(q)\ddot{q} + N(q, \dot{q}) \quad (4)$$

where $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times p}$ denotes a known regression matrix that is a function of $q(t)$, $\dot{q}(t)$, $\ddot{q}(t) \in \mathbb{R}^n$, and $\theta \in \mathbb{R}^p$ contains the unknown constant system parameters.

Assumption 1: Each of the constant system parameters defined in (4) can be lower and upper bounded as indicated by the following inequalities

$$\underline{\theta}_j < \theta_j < \bar{\theta}_j \quad (5)$$

where θ_j denotes the j th component of the vector θ , and $\underline{\theta}$, $\bar{\theta} \in \mathbb{R}^p$ denote vectors of known, constant bounds for the unknown parameters.

Assumption 2: A control is designed which ensures that in the absence of a fault (i.e., $\forall t < T$) $q(t)$, $\dot{q}(t)$, $\tau(t) \in \mathcal{L}_\infty$ and that $\lim_{t \rightarrow \infty} q(t) = q_d(t)$ where $q_d(t) \in \mathbb{R}^n$ represents the desired trajectory. Note that based on the form of the dynamic model given in (1), if $q(t)$, $\dot{q}(t)$, $\tau(t) \in \mathcal{L}_\infty$, it is clear that $\ddot{q}(t) \in \mathcal{L}_\infty$.

Remark 1: One method for detecting actuator faults could be to utilize (1) and (4) to isolate the fault as shown below

$$\zeta(t) = \tau - Y(q, \dot{q}, \ddot{q})\theta. \quad (6)$$

Unfortunately, due to the fact that (6) would require exact model knowledge of the system and acceleration measurements, it is clear that (6) is impractical for fault detection purposes; hence, we are motivated to craft a fault detection algorithm that is independent of link acceleration measurements and exact knowledge of the system parameters.

III. TORQUE FILTERING

Motivated by the desire to eliminate link acceleration measurements from the subsequent fault detection algorithm, we define a filtered torque signal denoted by $\tau_f(t) \in \mathbb{R}^n$ as follows [8]:

$$\tau_f = f * \tau \quad (7)$$

where $*$ denotes the standard convolution operation, $\tau(t)$ was defined in (1), the filter function, denoted by $f(t) \in \mathbb{R}^1$, is given by

$$f = \alpha \exp(-\beta t) \quad (8)$$

and $\alpha, \beta \in \mathbb{R}^1$ denote positive filter constants. By substituting the left-hand side of (1) into (7) for $\tau(t)$ and utilizing standard

convolution properties (see Appendix A), we can rewrite (7) in terms of the following linear parameterization:

$$\tau_f = Y_f \theta + \zeta_f \quad (9)$$

where θ denotes the same unknown, constant parameter vector defined in (4), $Y_f(q, \dot{q}) \in \mathbb{R}^{n \times p}$ denotes the measurable, filtered regression matrix which is independent of link acceleration measurements and is explicitly given by

$$\begin{aligned} Y_f \theta = & \dot{f}(t) * \{M(q(t))\dot{q}(t)\} + f(0)M(q(t))\dot{q}(t) \\ & - f(t)M(q(0))\dot{q}(0) + f(t) \\ & * \left\{ -\dot{M}(q(t))\dot{q}(t) + N(q(t), \dot{q}(t))\dot{q}(t) \right\} \quad (10) \end{aligned}$$

and $\zeta_f(t) \in \mathbb{R}^n$ denotes a filtered fault signal defined as follows:

$$\zeta_f(t) = f(t) * u_{-1}(t-T)\zeta(q, \dot{q}). \quad (11)$$

The structure of (9) will be exploited in the subsequent analysis; however, since θ is a vector of uncertain parameters, the structure of (9) cannot be implemented. An implementable form (i.e., a measurable, acceleration independent form) of (9) can be determined by utilizing (7) and (8) along with Laplace transform properties to generate the filtered torque signal via the following differential equality:

$$\dot{\tau}_f = -\beta\tau_f + \alpha\tau, \quad \tau_f(0) = 0 \quad (12)$$

where α, β were defined in (8).

Remark 2: Due to the structure of the above torque filtering technique, the filtered version of the fault is delayed from the actual fault; although, if β is made increasingly large, the delay is minimized. Based on (9), it also is clear that the fault can be isolated in terms of an expression that is independent of link acceleration measurements. Thus, we are now motivated to design an algorithm based on (9) that can detect actuator faults in the presence of parametric uncertainty.

IV. PREDICTION-ERROR-BASED FAULT DETECTION

The objective of this paper is to design an algorithm that can detect actuator faults in n -DOF robotic manipulators despite uncertainty in the mechanical parameters. To this end, we define a measurable prediction error signal, denoted by $\varepsilon(t) \in \mathbb{R}^n$, as follows:

$$\varepsilon = \tau_f - \hat{\tau}_f \quad (13)$$

where τ_f was defined in (12), and $\hat{\tau}_f \in \mathbb{R}^n$ is a subsequently designed filtered torque estimate. This is similar to one of the fault detection tests proposed in [20]; however, in [20] acceleration estimates were required for implementation.

A. Constant Best-Guess Estimate¹

Due to the presence of parametric uncertainty in (1), the filtered torque estimate given in (13) is designed as follows:

$$\hat{\tau}_f = Y_f \hat{\theta} \quad (14)$$

where $\hat{\theta} \in \mathbb{R}^p$ is a constant, best-guess parameter estimate for θ defined in (4) and $Y_f(q, \dot{q})$ was defined in (9). From the design of $\hat{\tau}_f(t)$, we can use (9), (13), and (14) to obtain the following new expression for $\varepsilon(t)$:

$$\varepsilon = Y_f \tilde{\theta} + \zeta_f \quad (15)$$

where $\tilde{\theta} \in \mathbb{R}^p$ quantifies the mismatch between the actual uncertain parameters and the constant, best guess parameter estimate as shown below

$$\tilde{\theta} = \theta - \hat{\theta}. \quad (16)$$

Based on *Assumption 1*, we can upper bound the prediction error signal given in (15) as follows:

$$|\varepsilon_i| \leq \rho_i(t) + |\zeta_{fi}| \quad (17)$$

where $\rho(t) \in \mathbb{R}^n$ is a positive bounding signal selected to satisfy the following inequality:

$$\left| (Y_f \tilde{\theta})_i \right| \leq \rho_i(t) \quad (18)$$

and $(\cdot)_i$ represents the i th element of a vector. Based on the structure of (17), we define a fault indicating, *dead-zone* residual function, denoted by $D_1[\cdot] \in \mathbb{R}^1$, as follows:

$$D_1[\varepsilon_i] = \begin{cases} |\varepsilon_i|, & \text{if } |\varepsilon_i| > \rho_i(t) \\ 0, & \text{if } |\varepsilon_i| \leq \rho_i(t) \end{cases} \quad (19)$$

to determine if a fault occurs. That is, if

$$D_1[\varepsilon_i] > 0 \quad (20)$$

then an actuator fault is present in the system; however, if the parameter uncertainty in the system is relatively large, then some faults may not be detected due to the inability of the fault detection scheme given in (19) to distinguish the actuator faults from the parameter uncertainty.

Remark 3: Since the ability of the fault detection algorithm to detect faults is directly linked to the degree of parametric uncertainty in the system [see (18) and (19)], we are motivated to examine the bounding signal $\rho(t)$ given in (18). One method for selecting $\rho(t)$ is given below

$$\rho_i(t) = \sum_{j=1}^p |Y_{fij}| \max \left\{ \left| \theta_j - \hat{\theta}_j \right|, \left| \bar{\theta}_j - \hat{\theta}_j \right| \right\} \quad (21)$$

where (5) and (16) were utilized. It is clear that selecting $\rho(t)$ according to (21) may yield a bound that is too conservative, and hence, the sensitivity of the fault detection algorithm is reduced.

¹The term best-guess-estimate is utilized to signify a constant parameter estimate that is defined by the user as a best-guess of the actual value of the unknown parameter. Specifically, the user may obtain a value for the best-guess estimate utilizing any of the appropriate parameter identification techniques that are found in literature.

Another method for selecting $\rho(t)$ is to utilize interval methods [9]. That is, since we assume that θ_j is contained in the interval $[\underline{\theta}_j, \bar{\theta}_j]$ [see (5)], a less conservative method for selecting $\rho(t)$ is given below

$$\rho_i(t) = \max(|\lambda_{1i}(t)|, |\lambda_{2i}(t)|) \quad (22)$$

where the bounding functions $\lambda_1(t), \lambda_2(t) \in \mathfrak{R}^n$ represent time-varying interval parameters that are generated on-line according to the following expression:

$$[\lambda_{1i}(t), \lambda_{2i}(t)] = \sum_{j=1}^p Y_{fij} \left([\underline{\theta}_j - \hat{\theta}_j, \bar{\theta}_j - \hat{\theta}_j] \right). \quad (23)$$

The advantage of selecting $\rho(t)$ in a less conservative manner [e.g., (22)] is demonstrated in the subsequent experimental verification section.

Remark 4: The motivation for selecting (19) as shown below

$$D_1[\varepsilon_i] = |\varepsilon_i|, \quad \text{if } |\varepsilon_i| > \rho_i(t)$$

versus some positive constant (i.e., $D_1[\varepsilon_i] = 1$ if $|\varepsilon_i| > \rho_i(t)$), arises from the additional flexibility gained with regard to observing the extent that the residual given in (19) was violated. That is, by utilizing (19), possible false alarm conditions that may occur (e.g., due to signal noise, numeric roundoff, etc.) may be avoided.

Remark 5: If exact model knowledge of the system is available, then we can simply redesign $\hat{\tau}_f(t)$ as follows:

$$\hat{\tau}_f = Y_f \theta \quad (24)$$

where θ defined in (4) is now assumed to be known. After substituting (9) and (24) into (13), we obtain the following expression for $\varepsilon(t)$:

$$\varepsilon = \zeta_f \quad (25)$$

hence, at least, in the theory, $\varepsilon(t) = 0$ for $t < T$. It should be noted that in practice small uncertainties and measurement noise will no doubt ensure that $\varepsilon(t) \neq 0$ for $t < T$ (i.e., $\|\varepsilon_i(t)\|$ will equal some unknown time-varying function); hence, we define a fault indicating, dead-zone residual function, denoted by $D_2[\cdot] \in \mathfrak{R}^1$, as follows:

$$D_2[\varepsilon_i] = \begin{cases} |\varepsilon_i|, & \text{if } |\varepsilon_i| > (\mu_o)_i \\ 0, & \text{if } |\varepsilon_i| \leq (\mu_o)_i \end{cases} \quad (26)$$

such that, if an actuator fault is present in the system

$$D_2[\varepsilon_i] > 0 \quad (27)$$

where $\mu_o \in \mathfrak{R}^n$ is a vector of positive, scalar design constants that are experimentally determined to account for small uncertainties and measurement noise.

B. Parameter Uncertainty with Adaptive Update Law

In order to craft a more sensitive fault detection algorithm, and hence, decrease the effects of the parameter uncertainty [i.e., reducing $\rho(t)$ in (19)], we now construct a dynamic on-line estimate for the uncertain parameters versus the constant, best-

guess estimate used in (14). Specifically, $\hat{\theta}(t)$ is generated via a prediction error driven gradient update law as follows:

$$\dot{\hat{\theta}}(t) = \Gamma Y_f^T \varepsilon \quad (28)$$

where $\Gamma \in \mathfrak{R}^{p \times p}$ is a constant, diagonal, positive-definite, adaptation gain matrix. In order to facilitate the subsequent analysis, we define a nonnegative function denoted by $V(\tilde{\theta}(t)) \in \mathfrak{R}^1$ as follows:

$$V = \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}. \quad (29)$$

After taking the time derivative of (29), utilizing the fact that $\dot{\tilde{\theta}}(t) = -\dot{\hat{\theta}}(t)$, and then substituting for (28), we obtain the following expression:

$$\dot{V} = -\varepsilon^T \varepsilon + \varepsilon^T \zeta_f \quad (30)$$

where (15) was utilized. After integrating both sides of (30), we have

$$V(\tilde{\theta}(t)) = - \int_0^t \|\varepsilon(\sigma)\|^2 d\sigma + \int_0^t \varepsilon^T(\sigma) \zeta_f(\sigma) d\sigma + V(\tilde{\theta}(0)). \quad (31)$$

Since $\zeta_f(t) = 0$ for $t < T$, we can utilize (31) to upper bound the integral of the norm of the prediction error as follows:

$$\int_0^t \|\varepsilon(\sigma)\|^2 d\sigma \leq \Lambda \quad \forall t < T \quad (32)$$

where the facts that $V(\tilde{\theta}(t))$ is nonnegative and

$$V(\tilde{\theta}(0)) \leq \Lambda \quad \forall t < T \quad (33)$$

have been utilized, where $\Lambda \in \mathfrak{R}^1$ is a positive constant defined as

$$\Lambda = \frac{1}{2} \|\Gamma^{-1}\|_{i\infty} \max \left\{ \left\| \underline{\theta} - \hat{\theta}(0) \right\|^2, \left\| \bar{\theta} - \hat{\theta}(0) \right\|^2 \right\} \quad (34)$$

and $\|\cdot\|_{i\infty}$ represents the induced infinity norm. Based on the structure of (32), we define a fault indicating, *dead-zone* residual function, denoted by $D_3[\cdot] \in \mathfrak{R}^1$, as follows:

$$D_3[\varepsilon] = \begin{cases} \int_0^t \|\varepsilon\|^2 d\sigma, & \text{if } \int_0^t \|\varepsilon\|^2 d\sigma > \Lambda \\ 0, & \text{if } \int_0^t \|\varepsilon\|^2 d\sigma \leq \Lambda. \end{cases} \quad (35)$$

That is, for the adaptive-based fault detection scheme, we can use the following residual:

$$D_3[\varepsilon] > 0 \quad (36)$$

or the residual given in (19) to determine if an actuator fault is present in the system.

Due to the fact that the residuals given by (19) and (35) are based on fixed, conservative thresholds, we are motivated to exploit the properties of the adaptive-based prediction error signal and design a *dead-zone* update rule to modify the *dead-zone* thresholds as $\|\varepsilon(t)\|$ decreases. To illustrate that $\lim_{t \rightarrow \infty} \|\varepsilon(t)\| = 0$ in the absence of a fault, we first

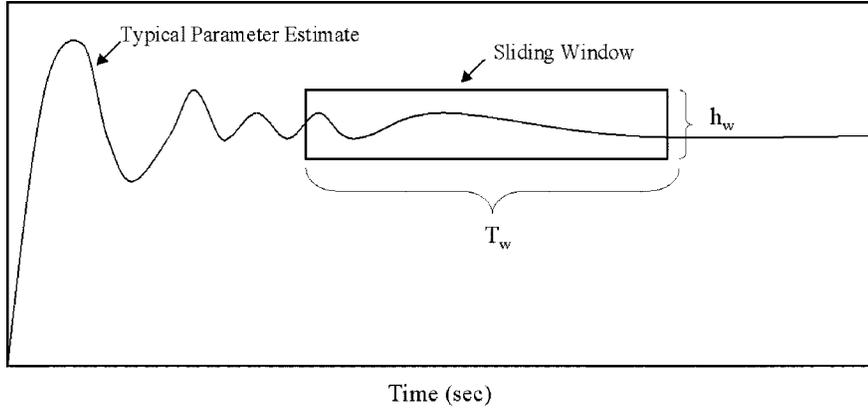


Fig. 1. Sliding window diagram for a typical parameter estimate.

use (16), (29), and (30) to conclude that $\tilde{\theta}(t), \hat{\theta}(t) \in \mathcal{L}_\infty$. Furthermore, from (12) and *Assumption 2*, we have that $Y_f(q, \dot{q}), \dot{Y}_f(q, \dot{q}, \ddot{q}), \tau_f(t), \dot{\tau}_f(t) \in \mathcal{L}_\infty$. Based on the facts that $Y_f(q, \dot{q}), \hat{\theta}(t) \in \mathcal{L}_\infty$, we can utilize (15) to conclude that $\varepsilon(t) \in \mathcal{L}_\infty$. Since $Y_f(q, \dot{q}), \varepsilon(t) \in \mathcal{L}_\infty$, we can conclude from (28) that $\dot{\hat{\theta}}(t) \in \mathcal{L}_\infty$. Based on the facts that $Y_f(q, \dot{q}), \dot{Y}_f(q, \dot{q}, \ddot{q}), \varepsilon(t), \hat{\theta}(t) \in \mathcal{L}_\infty$, we can prove that $\dot{\varepsilon}(t) \in \mathcal{L}_\infty$. Furthermore, utilizing (30), we can conclude that $\varepsilon(t) \in \mathcal{L}_2$; therefore, since $\varepsilon(t), \dot{\varepsilon}(t) \in \mathcal{L}_\infty$ and $\varepsilon(t) \in \mathcal{L}_2$, we can utilize Barbalat's Lemma to conclude that $\lim_{t \rightarrow \infty} \|\varepsilon(t)\| = 0$ in the absence of a fault. From (29) and (30), it is clear that

$$\lim_{t \rightarrow \infty} V(t) = \varrho_1 \quad (37)$$

in the absence of a fault where $\varrho_1 \in \mathbb{R}^1$ is a positive constant. Based on (37), it is straightforward to prove that $\lim_{t \rightarrow \infty} \hat{\theta}(t) = \varrho_o$ in the absence of a fault, where $\varrho_o \in \mathbb{R}^p$ is a vector of positive constants.

Based on the facts that $\lim_{t \rightarrow \infty} \|\varepsilon(t)\| = 0$ and $\lim_{t \rightarrow \infty} \hat{\theta}(t) = \varrho_o$ in the absence of a fault, we now design a fault indicating, *dead-zone* residual function with the improved capability of detecting smaller magnitude faults. That is, since the parameter estimates converge to constants, we construct an adjustable threshold window around each parameter estimate that allows for residual threshold reduction as the estimate values become constant. To detect if the condition $\lim_{t \rightarrow \infty} \hat{\theta}(t) = \varrho_o$ is satisfied before a potential fault, we define a set of sliding windows (see Fig. 1) with a user-defined window height, denoted by $h_{wj} \in \mathbb{R}^1$, and a user-defined window length, denoted by $T_{wj} \in \mathbb{R}^1$, as follows:

$$\left| \hat{\theta}_j(t) \right| - \frac{1}{2}h_{wj} \leq \left| \hat{\theta}_j(t_{sw}) \right| \leq \left| \hat{\theta}_j(t) \right| + \frac{1}{2}h_{wj} \quad (38)$$

$\forall t_{sw} \in [t - T_{wj}, t]$ and $j = 1, 2, \dots, p$. If $\hat{\theta}_j(t)$ violates (38) at any time instant during the interval $[0, T_{wj}]$ (i.e., if the parameter estimate has not settled within the tolerances of the window), then the window will slide along the parameter estimate with time. If $\hat{\theta}_j(t)$ satisfies (38) for all values of time during the interval $[t - T_{wj}, t]$, then it is clear that the parameter estimates have approximately settled within the tolerances of the sliding window, and hence, $\hat{\theta}_j(t)$ has approximately converged to a constant. When (38) is satisfied for all of the param-

eter estimates (i.e., $\forall j = 1, 2, \dots, p$), the maximum value of the prediction error is measured and utilized as the threshold for the residual. That is, the time instant when the final parameter estimate is contained within the user-defined sliding window, denoted by $T_m \in \mathbb{R}^1$, the maximum value of the prediction error (i.e., $\max |\varepsilon_i(T_m)|$) is recorded and utilized as the residual threshold. Specifically, we define a fault indicating, *dead-zone* residual function, denoted by $D_4[\cdot] \in \mathbb{R}^1$, as follows:

$$D_4[\varepsilon_i] = \begin{cases} |\varepsilon_i|, & \text{if } |\varepsilon_i| > \max |\varepsilon_i(T_m)| \\ 0, & \text{if } |\varepsilon_i| \leq \max |\varepsilon_i(T_m)| \end{cases} \quad (39)$$

such that if

$$D_4[\varepsilon_i] > 0 \quad (40)$$

then an actuator fault is present in the system.

Remark 6: Note that strictly speaking, we cannot guarantee that $\max |\varepsilon_i(T_m)|$ is a true upper bound for $|\varepsilon_i(t)|$ in the absence of a fault; hence, some *false alarms* may result. However, since $\varepsilon(t) \in \mathcal{L}_2$, if the lengths of the sliding windows are selected to be appropriately long enough then the likelihood of false alarms can be practically eliminated.

V. EXPERIMENTAL VERIFICATION

The proposed prediction-error-based fault detection algorithm was implemented on an Integrated Motion Inc. two-link, revolute, direct-drive robot manipulator (see Fig. 2) with the following dynamics [3]:

$$\begin{aligned} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} p_1 + 2p_3 \cos(q_2) & p_2 + p_3 \cos(q_2) \\ p_2 + p_3 \cos(q_2) & p_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\ &+ \begin{bmatrix} -p_3 \sin(q_2) \dot{q}_2 & -p_3 \sin(q_2) (\dot{q}_1 + \dot{q}_2) \\ p_3 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \\ &\cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} f_{d1} & 0 \\ 0 & f_{d2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\ &+ \begin{bmatrix} f_{s1} & 0 \\ 0 & f_{s2} \end{bmatrix} \begin{bmatrix} \text{sgn}(\dot{q}_1) \\ \text{sgn}(\dot{q}_2) \end{bmatrix} \end{aligned} \quad (41)$$

where the mechanical parameters have been experimentally determined as follows: $p_1 = 3.473 \text{ kg}\cdot\text{m}^2$, $p_2 = 0.193 \text{ kg}\cdot\text{m}^2$, $p_3 = 0.242 \text{ kg}\cdot\text{m}^2$, $f_{d1} = 1.3 \text{ Nm}\cdot\text{s}$, $f_{d2} = 0.88 \text{ Nm}\cdot\text{s}$, $f_{s1} =$

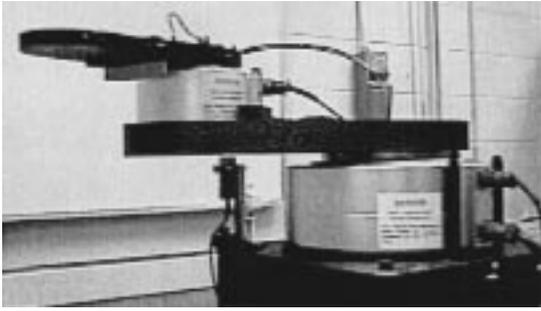


Fig. 2. IMI direct-drive manipulator.

1.519 Nm, and $f_{s2} = 0.932$ Nm. For experimental verification of the proposed prediction-error-based fault detection algorithms, the above mechanical parameters are assumed to be unknown. In addition, the parameter vector corresponding to θ given in (4) is shown below

$$\begin{aligned} & [p_1 \ p_2 \ p_3 \ f_{d1} \ f_{d2} \ f_{s1} \ f_{s2}]^T \\ & = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6 \ \theta_7]^T. \end{aligned} \quad (42)$$

The links of the robotic manipulator are directly actuated by switched-reluctance motors which are controlled through NSK torque controlled amplifiers. A Pentium 266-MHz PC operating under Qmotor 3.0² hosts the control and fault detection algorithms. Data acquisition, control implementation, and fault detection were performed using the Quanser MultiQ I/O board at a frequency of 1 kHz.

To demonstrate the performance of the prediction-error-based, best-guess estimate and adaptive estimate fault detection algorithms, a *free-swinging* actuator fault, a *ramp* actuator fault [with $(\gamma_1)_i = 1.0$], and a *saturation* actuator fault were injected at $t = 34.0$ s on Link 1 and Link 2, respectively, where a standard proportional derivative control scheme was utilized to ensure that $q(t)$ tracks $q_d(t)$ where

$$q_{di}(t) = 1.5 \sin(1.75t). \quad (43)$$

Furthermore, in order to examine the sensitivity of the fault detection algorithm with regard to $\rho_i(t)$ (see Remark 3), the best-guess estimate experiments were performed with $\rho_i(t)$ selected according to (21)–(23), with (21) and (23) modified as shown below

$$\begin{aligned} [\lambda_{1i}(t), \lambda_{2i}(t)] &= \sum_{j=1}^p Y_{fij} \left(|\underline{\theta}_j - \hat{\theta}_j, \bar{\theta}_j - \hat{\theta}_j| \right) + (\gamma_2)_i \quad (44) \\ \rho_i(t) &= \sum_{j=1}^p |Y_{fij}| \max \left\{ \left| \bar{\theta}_j - \hat{\theta}_j \right|, \left| \underline{\theta}_j - \hat{\theta}_j \right| \right\} + (\gamma_3)_i \end{aligned} \quad (45)$$

where $\gamma_2, \gamma_3 \in \mathfrak{R}^n$ are vectors of positive constants experimentally determined as shown below

$$\gamma_2 = [0.4 \ 0.04] \quad \gamma_3 = [0.35 \ 0.05]$$

to provide robustness to signal noise, numerical roundoff error, etc., $\bar{\theta}_j, \underline{\theta}_j$ were selected to be $\pm 40\%$ of the actual parameters,

²Available at <http://www.qrts.com>

TABLE I
COMPARISON OF DELAY TIMES (ms) BETWEEN THE ACTUATOR FAULT OCCURRENCE AND THE FAULT DETECTION

| | Best -Guess Estimate Case 1 | Best -Guess Estimate Case 2 | Adaptive Estimate |
|------------------------------------|--------------------------------------|--------------------------------------|----------------------|
| Free-Swinging Actuator Fault | Link 1 | Link 1 | Link 1 |
| | 751.7 | 61.3 | 50.2 |
| | Link 2 | Link 2 | Link 2 |
| Ramp Actuator Fault | Link 1 | Link 1 | Link 1 |
| | 488.9 | 72.7 | 53.2 |
| | Link 2 | Link 2 | Link 2 |
| Saturated Actuator Fault | Link 1 | Link 1 | Link 1 |
| | 57.5 | 16.7 | 15.5 |
| | Link 2 | Link 2 | Link 2 |
| | 19.1 | 10.4 | 7.8 |

and $\hat{\theta}_j$ was selected to be 20% of the parameter values for each of the two cases, respectively. For the experimental verification of the adaptive estimate prediction-error-based fault detection algorithm, we selected $\hat{\theta}_j(t=0) = 0.0$ (i.e., we assume that no prior knowledge of the mechanical parameters exists) and Γ of (28) was selected as shown below

$$\Gamma = \text{diag} \{5, 1, 1, 3, 2.25, 1.5, 2.25\}. \quad (46)$$

For each of the aforementioned experiments, the filter parameters given in (8) were selected as

$$\alpha = 1 \quad \beta = 10. \quad (47)$$

A comprehensive comparison of the delay time between the fault occurrence and the fault detection for each of the above cases was recorded in Table I. Fault histories for each of the above cases are depicted in Figs. 3–8 for the best-guess estimate case [where $\rho(t)$ was selected according to (45) (referred to as Case 1 in the figures) and (44) (referred to as Case 2 in the figures)] and the adaptive estimate case given in (39). In each of the figures, the marker “(a)” represents the residual threshold, the marker “(b)” indicates the prediction error, and the marker “(c)” indicates the instant the fault is detected. The left column of plots in Figs. 3–8 depicts the full time scale, whereas the right column of plots depicts the last 1.5 s in order to more clearly illustrate when the fault is detected. Note that we selected a time instant to inject the faults that corresponds to a time instant when all the adaptive parameter estimates had settled within the respective sliding windows (approximately 22 s); hence, the fault indicating *dead-zone* residual function given in (39) was utilized for the adaptive estimate case. If the fault had been injected prior to the parameter estimates converging then either the residual given in (19) or (36) could have been utilized. After implementing the fault indicating *dead-zone* residual function given in (36), we concluded that (19) provided superior fault detection capabilities; hence, only data utilizing (19) was recorded for comparison purposes.

From Table I, it is clear that the performance of the adaptive estimate fault detection scheme provided superior fault detec-

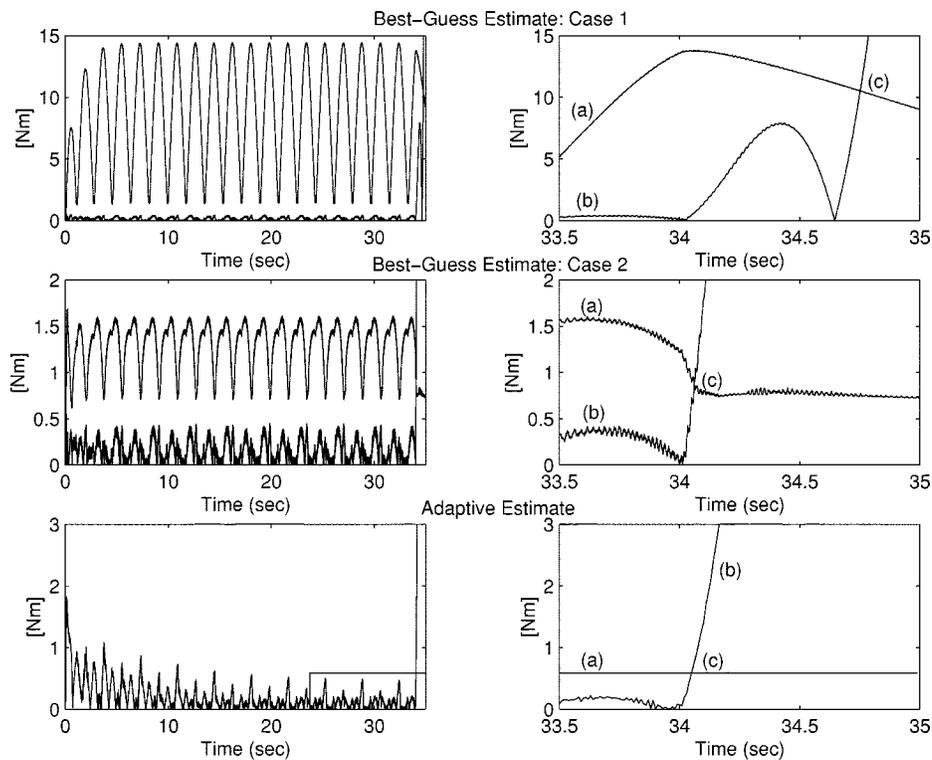


Fig. 3. Fault detection for a *free swinging* actuator fault on link 1.

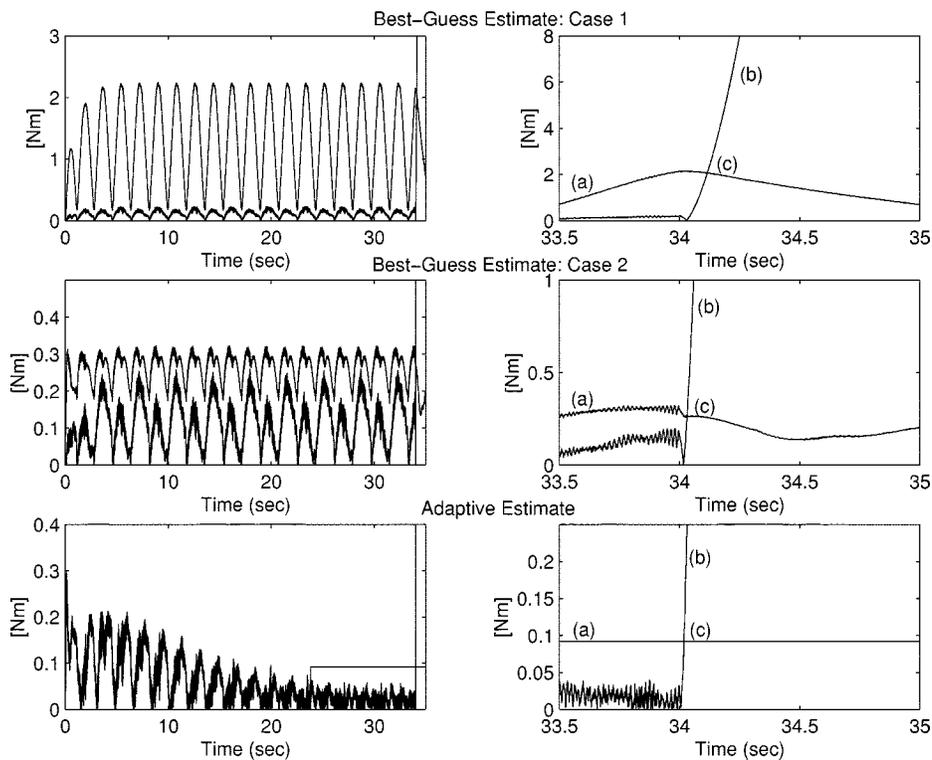


Fig. 4. Fault detection for a *free swinging* actuator fault on link 2.

tion capabilities. That is, provided each of the adaptive estimates have approximately settled to a constant, the time delay before an injected fault was detected was significantly less than the robust fault detection scheme. In addition, it is clear from Table I that the time delay before an injected fault was detected was sig-

nificantly decreased by utilizing the less conservative bounding function $\rho_i(t)$ given in (22) and (23) [versus the bounding function given in (21)]; hence, the performance of the proposed best-guess estimate fault detection algorithm may be improved by selecting the bounding function $\rho_i(t)$ differently from (21)

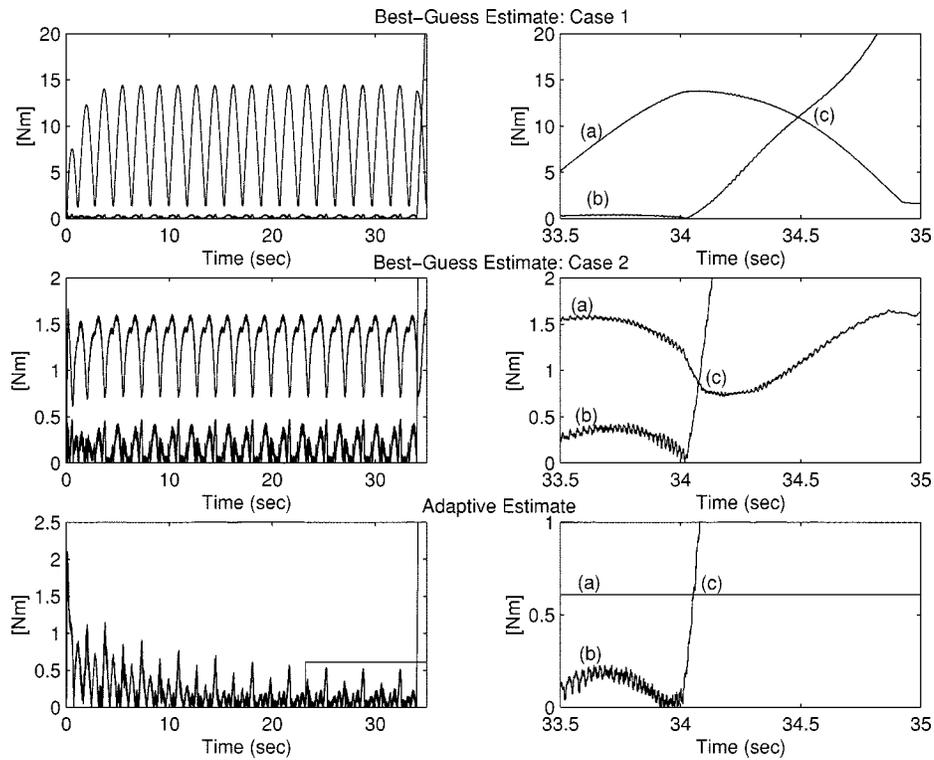


Fig. 5. Fault detection for a *ramp* actuator fault on link 1.

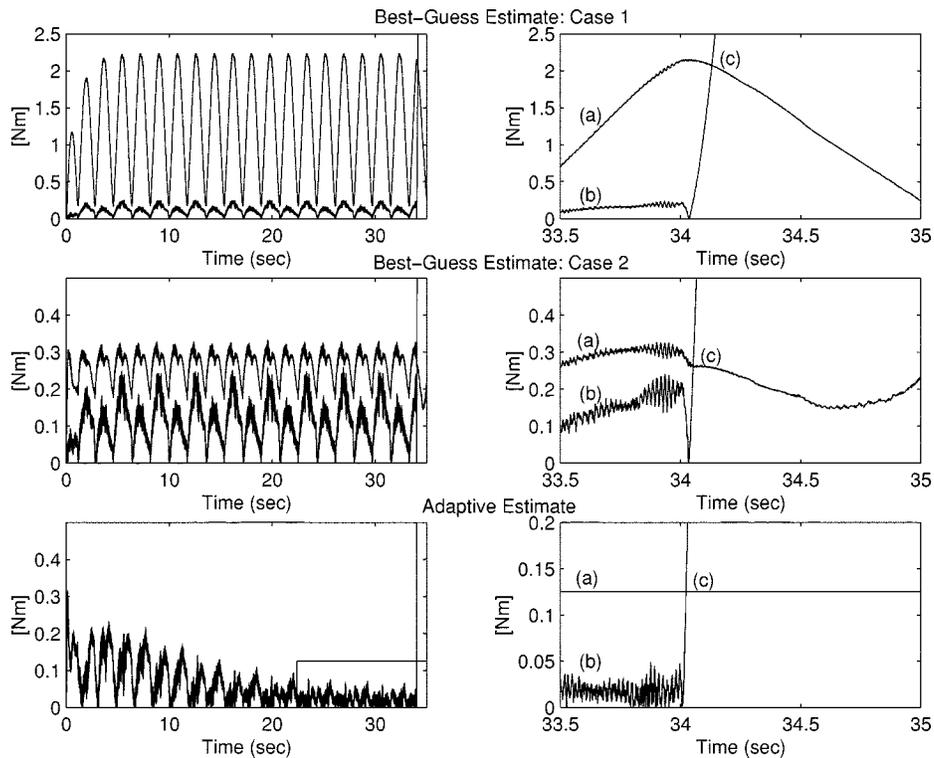


Fig. 6. Fault detection for a *ramp* actuator fault on link 2.

to (23) (i.e., an experimentally determined constant threshold could be utilized).

Remark 7: Note that the disturbances such as signal noise, numerical roundoff error, etc. can be incorporated in the dynamic model of the robot manipulator given in (1) as an

additive bounded disturbance. Assuming that the upper and lower bounds of the additive bounded disturbance are known, it is straightforward to include the bounding terms γ_2 and γ_3 in the theoretical development of the constant best-guess estimate fault detection scheme.

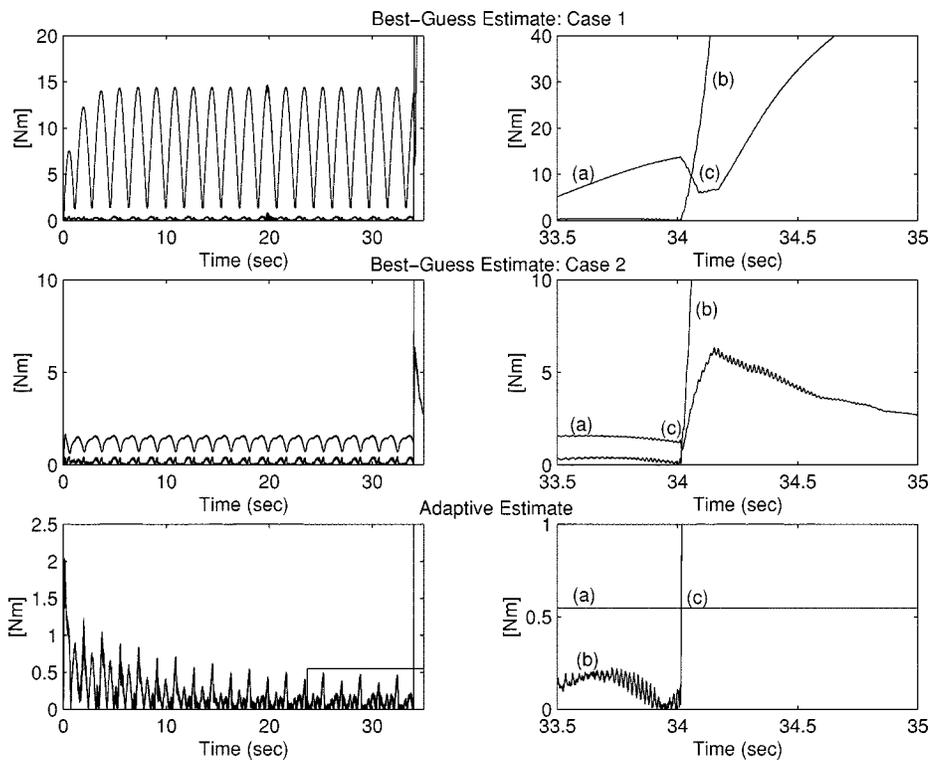


Fig. 7. Fault detection for a *saturated* actuator fault on link 1.

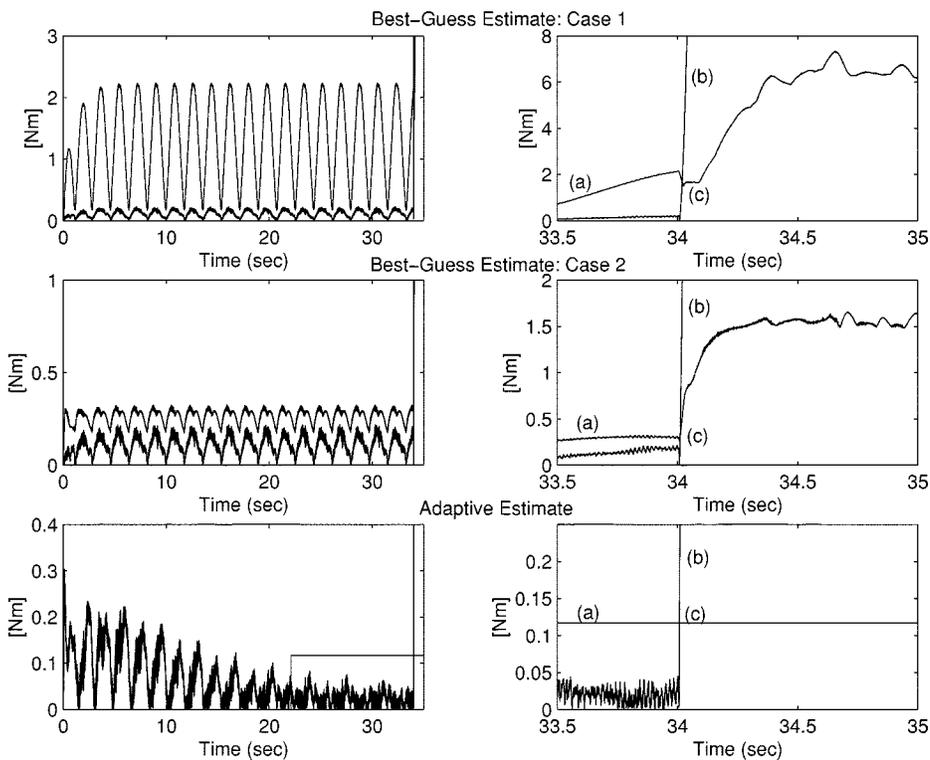


Fig. 8. Fault detection for a *saturated* actuator fault on link 2.

VI. CONCLUSION

In conclusion, we provide a fault detection method for robot manipulators, for exact model knowledge and parameter uncertainty cases. Two methods are presented for fault detection

under parameter uncertainty: best-guess parameter estimates and a dynamic prediction error driven parameter update law. Conditions were given for all three fault detection schemes and a strategy was developed for detecting smaller faults after the parameter estimates have converged. One of the advantages of

the proposed fault detection scheme is that it is independent from the controller; hence, the fault detection scheme can be utilized in conjunction with any controller (provided link position and velocity measurements are available). The effectiveness of the proposed fault detection methods was illustrated through experimental results.

APPENDIX

In order to rewrite (7) in terms of the linear parameterization given in (9), we first note that (1) can be written in the following form [8]:

$$\tau = \dot{h} + g \quad (48)$$

where

$$\dot{h} = \frac{d}{dt}(M(q)\dot{q}) \quad (49)$$

and

$$g = -\dot{M}(q)\dot{q} + N(q, \dot{q}) + u_{-1}(t - T)\zeta(q, \dot{q}). \quad (50)$$

After substituting (48) into (7), we obtain the following expression:

$$\begin{aligned} \tau_f = & \dot{f}(t) * \{M(q(t))\dot{q}(t)\} + f(0)M(q(t))\dot{q}(t) \\ & - f(t)M(q(0))\dot{q}(0) + f(t) \\ & * \left\{ -\dot{M}(q(t))\dot{q}(t) + N(q(t), \dot{q}(t))\dot{q}(t) \right. \\ & \left. + u_{-1}(t - T)\zeta(q, \dot{q}) \right\} \end{aligned} \quad (51)$$

where the facts that

$$f * \{\dot{h} + g\} = f * \dot{h} + f * g \quad (52)$$

and

$$f * \dot{h} = \dot{f} * h + f(0)h - fh(0) \quad (53)$$

have been utilized. Hence, based on (51), it is straightforward to conclude that (7) can be rewritten in the structure given in (9).

REFERENCES

- [1] F. Caccavale and I. D. Walker, "Observer-based fault detection for robot manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 2881–2887.
- [2] C. Chan, K. Cheung, Y. Wang, and W. Chan, "On-line fault detection and isolation of nonlinear systems," in *Proc. American Control Conf.*, 1999, pp. 3980–3984.
- [3] *Direct Drive Manipulator Research and Development Package Operations Manual*. Berkeley, CA: Integrated Motion Inc., 1992.
- [4] 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," in *Proc. IEEE Int. Conf. Robotics and Automation*, April 2000, pp. 3628–3634.
- [5] J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, 1996, pp. 460–467.
- [6] B. Freyermuth, "An approach to model-based fault diagnosis of industrial robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 1991, pp. 1350–1356.

- [7] 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.
- [8] F. Lewis, C. Abdallah, and D. Dawson, *Control of Robot Manipulators*. New York: MacMillan, 1993.
- [9] A. Neumaier, *Interval Methods for Systems of Equations*. New York: Cambridge Univ. Press, 1990.
- [10] C. J. 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.
- [11] S. I. Roumeliotis, G. Sukhatme, and G. A. Bekey, "Fault detection and identification in a mobile robot using multiple-model estimation," in *Proc. IEEE Int. Conf. Robotics and Automation*, Lueven, Belgium, 1998, pp. 2223–2228.
- [12] H. Schneider and P. M. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, no. 3, pp. 274–282, 1996.
- [13] 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. Robotics and Automation*, Detroit, MI, 1999, pp. 861–866.
- [14] M. Soika, "Grid based fault detection and calibration of sensors on mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Albuquerque, NM, 1997, pp. 2589–2594.
- [15] D. Sreevijayan, S. Tosunoglu, and D. Tesar, "Architectures for fault tolerant mechanical systems," in *Proc. IEEE Mediterranean Electrotechnical Conf.*, Antalya, Turkey, 1994, pp. 1029–1033.
- [16] Y. Ting, S. Tosunoglu, and D. Tesar, "A control structure for fault tolerant operation of robotic manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, pp. 684–690.
- [17] A. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Trans. Neural Networks*, vol. 8, no. 6, pp. 1410–1419, 1997.
- [18] —, "Robust nonlinear fault diagnosis in input-output systems," *Int. J. Contr.*, vol. 68, no. 2, pp. 343–360, 1997.
- [19] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robotic fault detection and fault tolerance: A survey," *Rel. Eng. Syst. Safety*, vol. 46, pp. 139–158, 1994.
- [20] —, "A dynamic fault tolerance framework for remote robots," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 477–490, Aug. 1995.
- [21] H. Wang, Z. Huang, and S. Daley, "On the use of adaptive updating rules for actuator and sensor fault diagnosis," *Automatica*, vol. 33, no. 2, pp. 217–225, 1997.
- [22] 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.
- [23] 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. Automat.*, vol. 9, pp. 675–684, Oct. 1993.
- [24] J. Wunnenberg and P. M. Frank, "Dynamic model based incipient fault detection concept for robots," in *Proc. IFAC 11th Triennial World Congress*, Tallinn, Estonia, USSR, 1990, pp. 61–66.
- [25] F. Zanaty, "Consistency checking techniques for the space shuttle remote manipulator system," *SPAR J. Eng. Technol.*, vol. 2, no. 1, pp. 40–49, 1993.
- [26] X. Zhang, T. Parisini, and M. Polycarpou, "Robust parameteric fault detection and isolation for nonlinear systems," in *Proc. 38th Conf. Decision and Control*, 1999, pp. 3102–3107.



Warren E. Dixon (S'94–M'00) was born in 1972 in York, PA. He received the B.S. and Ph.D. degrees in 1994 and 2000, respectively, from the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, and the M.E. degree in 1997 from the Department of Electrical and Computer Engineering, University of South Carolina, Columbia, SC.

After completing the Ph.D. degree, he was selected as an Oak Ridge National Laboratory Wigner Fellow where he currently works in the Robotics and Process Systems Division. His recent publications have been in the following areas: mobile robotics, fault detection, adaptive and robust control, learning control, amplitude limited control, output feedback control, visual-servoing, and the control of underactuated systems.



Ian D. Walker (S'84–M'89) received the B.Sc. degree in mathematics from the University of Hull, U.K., in 1983. He received the M.S. and Ph.D. degrees, both in electrical engineering, from the University of Texas at Austin, in 1985 and 1989, respectively.

From 1989 to 1997, he was with the Department of Electrical and Computer Engineering, Rice University, Houston, TX, where he held the positions of Assistant and Associate Professor. Since 1997, he has been an Associate Professor in the Department

of Electrical and Computer Engineering, Clemson University, Clemson, SC. His research interests are in the areas of robotics, particularly robot hands and grasping, robot reliability and fault detection, and kinematically redundant robots.

Dr. Walker is a member of Beta Alpha Phi, Eta Kappa Nu, Phi Kappa Phi, and Tau Beta Pi.

John P. Hartraft (M') was born in Homestead, FL, in 1976. He received the B.S. degree from Bob Jones University, Greenville, SC, in 1998, and the M.S. degree from the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, in 2000.

He is currently a Test Engineer with Michelin Americas Research and Development Corporation. As a graduate student at Clemson University, his research focused on nonlinear model based control of robotic systems.

Darren M. Dawson (S'89–M'90–SM'94) was born in Macon, GA, in 1962. He received the associate degree in mathematics from Macon Junior College in 1982 and the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, in 1984. In 1987, he returned to the Georgia Institute of Technology where he received the Ph.D. degree in electrical engineering in March 1990.

He was with Westinghouse as a Control Engineer from 1985 to 1987. During his pursuit of the Ph.D. degree, he also served as a Research/Teaching Assistant. In July 1990, he joined the Electrical and Computer Engineering Department where he currently holds the position of Professor. Currently, he leads the Robotics and Manufacturing Automation Laboratory, which is jointly operated by the Electrical and Mechanical Engineering Departments. His main research interests are in the fields of nonlinear based robust, adaptive, and learning control with application to electromechanical systems including robot manipulators, motor drives, magnetic bearings, flexible cables, flexible beams, high-speed transport systems, mobile robots, underactuated systems, and aerospace systems.