Robotica (2006) volume 24, pp. 727–738. © 2006 Cambridge University Press
doi:10.1017/S0263574706002906 Printed in the United Kingdom

# Experimental results for output feedback adaptive robot control

John M. Daly* and Howard M. Schwartz

*Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada*

(Received in Final Form: July 3, 2006. First published online: August 8, 2006)

**Summary**

This paper examines three methods of adaptive output feedback control for robotic manipulators. Implementing output feedback control allows use of only the position information, which can be measured quite accurately. Velocity and acceleration measurements can get corrupted by noise. A method proposed by K. W. Lee and H. K. Khalil [Adaptive output feedback control of robot manipulators using high-gain observer, *Int. J. Control*, **6**, 869–886 (1997)] using a high-gain observer, one proposed by J. J. Craig, P. Hsu and S. S. Sastry [Adaptive control of mechanical manipulators, *Int. J. Robot. Res.*, **6**(2), 16–27 (1987)] with the addition of a linear observer that we propose, and a method proposed by R. Gourdeau and H. M. Schwartz [Adaptive control of robotic manipulators: Experimental results, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation* (Apr. 1991) pp. 8–15] using an Extended Kalman Filter are examined. The methods are implemented in simulation and experimentally on a direct-drive robot. The performance of each of the algorithms is compared.

KEYWORDS: Adaptive control; System identification; Nonlinear systems; Robot manipulators.

## 1. Introduction

The use of adaptive control strategies for robot manipulators is an area that has received interest from researchers over the last number of years. When exact knowledge of the robot parameters is unknown, or the dynamics of a manipulator may be changing over time due to varying payload masses, an adaptive strategy is useful in estimating unknown parameters and modifying the controller to minimize tracking error.

In many such adaptive control algorithms,[1,2] position, velocity, and sometimes acceleration are required for the control and/or adaptation laws. However, while the position of a robot link can be measured accurately, measurement of velocity and acceleration tends to result in noisy signals.[3] In extreme cases, these signals could be so noisy that their use in the control or adaptation would no longer be feasible. In order to overcome the problem of noisy velocity and acceleration measurements, an observer can be used to estimate these values based on position measurements only. Not only can such a method help to yield velocity and acceleration estimates with less noise than their measured values, but robot manipulator setups using this sort of approach need only to be equipped with position sensors. This can help decrease the cost of production.

Lee and Khalil[4] develop one such method of output feedback adaptive control using a high-gain observer. This observer is used to estimate position error and velocity error, and these estimated values are used in the control and adaptation laws. An acceleration signal is not required for either the control law or adaptation law. The advantage of such a high-gain observer is that the error in the observed signals tends toward zero quite rapidly, and the system is quick to recover performance similar to that achieved under full-state feedback control. A potential drawback of this method is that, due to the high gain of the observer, the presence of noise in the measurements (such as quantization error on digitized position measurements) may make estimation of the position and velocity errors inaccurate to the point of not being useful. The noise on the position measurements would be amplified many times by the high observer gain.

In order to overcome difficulties with noise in a high-gain observer, we propose an addition to the adaptive controller developed by Craig, Hsu and Sastry[1] that involves the use of a linear second-order observer to estimate position, velocity and acceleration signals. When the Computed Torque Method (CTM) is used to control a nonlinear robot manipulator, assuming perfect linearization occurs, each link of the closed-loop system can be modeled as a double integrator. We then use a simple second-order linear observer constructed based on the dynamics of the double integrator to estimate position, velocity and acceleration from position measurements. The observer gains are set large enough so that the observer poles are significantly faster than the error dynamics of the system. As a a result, the effect of observer error has a minimal impact on the control of the system. However, the observer gains must be small enough to ensure that the algorithm is still effective in the presence of noise.

The adaptive output feedback control method proposed by Gourdeau and Schwartz[5] using an Extended Kalman Filter (EKF) will also be examined. The noise rejection properties of the EKF are desirable in the presence of noise due to quantization error on the measurements. In addition, the theory behind Kalman filtering is useful in selecting the tuning parameters of the filter.[5] The EKF is used to estimate both position and velocity, as well as the unknown robot parameters. This results in a consolidated approach where a separate adaptation law is not required. However, the use of the EKF can be computationally quite expensive.

This paper will examine both simulation and experimental results for each of the three algorithms. Simulations will

* Corresponding author. E-mail: jmdaly@ieee.org

be performed using the dynamics of a two-link robot manipulator operating in the horizontal plane. The experiments will be conducted using the Carleton University direct-drive robot, a two-degree-of-freedom robot with a parallelogram linkage operating in the horizontal plane. The models used for each of the cases, simulation and experimentation, will be outlined in Section 2.

In Section 3, the algorithm proposed by Lee and Khalil will be examined in more detail. Section 4 discusses Craig's method of adaptive control with the addition of an observer, as mentioned earlier. Section 5 will examine Gourdeau and Schwartz's adaptive algorithm based on the EKF. In each of the sections, simulation results are presented and compared, for the case of no noise as well as in the presence of noise mimicking position resolver quantization error. Experimental results based on the Carleton University direct-drive robot are also presented and compared in each of the sections.

## 2. Robot dynamic models

The algorithms described in this work will be implemented in simulation as well as on an experimental robot. In each of these cases, the dynamic model used is different. The simulated robot is a two-link manipulator, with two degrees of freedom, that operates in the horizontal plane. The experimental platform is a four-link manipulator configured in a parallelogram linkage. This robot is also a two-degree-of-freedom robot operating in the horizontal plane.

Both robots are represented using equations having the following form, given for the case of a $n$-degree-of-freedom manipulator

$$T = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{1}$$

where $T$ is a $n \times 1$ vector of torques applied to the joints, $M(q)$ is the $n \times n$ mass (or inertia) matrix, $q$ is the $n \times 1$ vector of joint positions, $C(q,\dot{q})\dot{q}$ is the $n \times 1$ vector of centrifugal and Coriolis terms, and $G(q)$ is the $n \times 1$ gravity vector. The effects of friction are neglected here. The dynamics of Eq. (1) may also be expressed in linear regression form for $p$-robot parameters as

$$T = Y(q,\dot{q},\ddot{q})\theta$$

where $Y(q,\dot{q},\ddot{q})$ is an $n \times p$ matrix of known functions, and $\theta$ is a $p \times 1$ vector of robot parameters.

### *2.1. Simulated robot dynamics*

For the simulations in this work, the dynamics of a $n = 2$ degree of freedom serial link manipulator are used. The equation of dynamics for this manipulator is given by Schwartz[6] and is repeated here for convenience. The dynamics take the form of Eq. (1) but the robot operates in the horizontal plane, and as such $G(q)$ is zero. Note that a robot operating in the vertical plane could also have been used without affecting the algorithms. The controllers used for the robot will linearize and decouple the robot dynamics including the effects of gravity. There are $p = 2$ parameters to be estimated for the robot, they are

$$\theta = \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}m_1 l^2\\ m_2 l^2\end{bmatrix}$$

The mass and coriolis matrices are given as

$$M(q) = \begin{bmatrix}\theta_1 + 2\theta_2 + 2\theta_2\cos q_2 & \theta_2 + \theta_2\cos q_2\\ \theta_2 + \theta_2\cos q_2 & \theta_2\end{bmatrix} \tag{2}$$

$$C(q,\dot{q}) = \begin{bmatrix}-2\theta_2\dot{q}_2\sin q_2 & -\theta_2\dot{q}_2\sin q_2\\ \theta_2\dot{q}_1\sin q_2 & 0\end{bmatrix} \tag{3}$$

To generate the trajectory for the robot to follow, a command signal consisting of a square wave with a period of 20 s was used. The square wave has peak values of ±1 rad. This signal was pre-filtered using a critically damped second-order linear filter with a bandwidth of $\omega_n = 2.0$ rad/s. The transfer function for this filter is given as

$$G(s) = \frac{4}{s^2 + 4s + 4} \tag{4}$$

The simulations were performed using a fourth-order Runge–Kutta algorithm to numerically integrate the continuous time dynamics. Cases free of noise were run, as well as cases having uniformly distributed random noise to simulate the quantization error of a 12-b position resolver.

In all simulations, the controller dynamics were set to have a bandwidth $\omega_n$ of 2.0 rad/s, and a damping ratio $\zeta$ of 1, matching the bandwidth of the trajectory pre-filter Eq. (4). This resulted in controller feedback gains of $K_{\mathrm{p}} = 4I_{2\times 2}$ and $K_{\mathrm{d}} = 4I_{2\times 2}$. The sample period for all of the simulations was set at $T_{\mathrm{s}} = 0.001$, corresponding to a sampling frequency of 1000 Hz, which is well above the bandwidth of the closed-loop system and is faster than the observer time constants. In all cases, the robot parameter estimates were initialized to $\hat{\theta}_1 = 1.5$ and $\hat{\theta}_2 = 3$, representing 1.5 times the true parameter values. The true parameter values used in the simulated dynamics are $\theta_1 = 1$ and $\theta_2 = 2$.

### *2.2. Carleton university direct-drive robot dynamics*

All control algorithms in this work have been implemented experimentally using the Carleton University direct-drive robot. This robot is a $n = 2$ degree of freedom robot, consisting of four links in a parallelogram linkage, which operates in the horizontal plane. The dynamics of this robot are derived by Gourdeau and Schwartz[7] and are now presented. These dynamics take the form of Eq. (1), but since the robot operates in the horizontal plane $G(q)$ is zero. In this case there are $p = 3$ robot parameters to be estimated, they are

$$\theta = \begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix} = \begin{bmatrix}I_0 + I_2 + m_2 l_4^2 + m_3 l_2^2\\ I_1 + I_3 + m_2 l_1^2 + m_3 l_3^2\\ m_2 l_1 l_4 + m_3 l_2 l_3\end{bmatrix}$$

where $I_0$ through $I_3$ represent inertia terms, $l_1$ through $l_4$ are the lengths of each of the links, and $m_2$ and $m_3$ are the masses of links 2 and 3, respectively.

The mass and coriolis matrices are given as

$$M(q) = \begin{bmatrix} \theta_1 & \theta_3 \cos(q_1 - q_2) \\ \theta_3 \cos(q_1 - q_2) & \theta_2 \end{bmatrix} \quad (5)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & \theta_3 \dot{q}_2 \sin(q_1 - q_2) \\ -\theta_3 \dot{q}_1 \sin(q_1 - q_2) & 0 \end{bmatrix} \quad (6)$$

The links of the robot are driven by two brushless DC servo motors. Each motor is supplied with current by a servo amplifier, which is powered by a high-voltage power supply.[8] Each motor has a resolver for measuring position, and a tachometer for measuring velocity. The resolver signals are converted by the amplifiers into 12-b digital words. These 12-b words are then sent to the computer through digital input/output ports of a digital interface card. The analog tachometer signals are sent to an analog/digital card inside the computer, which samples and digitizes the velocity measurements using 12 bits of resolution.

To drive the motors, the torques computed by the controller are scaled to a 12-b number, and are then output to the digital/analog converter where they are converted to a ±5-V signal that is sent to the amplifiers. The amplifiers take these signals and convert them to 20-KHz Pulse Width Modulated (PWM) signals, which drive the motors. The torque constant for each motor was measured at 1.7 Nm/V, giving a torque saturation value of 8.5 Nm (based on a ±5-V maximum control signal).[9]

The computer used to control the robot has an Intel Pentium-II CPU running at 300 MHz, with 128 MB of RAM. All experiments were run using a sample period of $T_s = 0.002$, giving a sample frequency of 500 Hz.

When running the experiments, a trajectory consisting of a combination of sines and cosines at different frequencies is selected. It is based on a trajectory used by Warshaw[8] and is given as

$$y_1(t) = \cos(2\omega t) - \cos(4\omega t) \quad (7)$$

$$y_2(t) = (\pi/2) - 1.9 + \sin(\omega t) + \sin(2\omega t) \quad (8)$$

where $\omega$ was chosen to be 1.257 rad/s.

This signal was pre-filtered using a critically damped second-order linear filter with a bandwidth of $\omega_n = 2.0$ rad/s. A filter with this bandwidth was selected to ensure that the torques required to track the trajectory were kept within the torque saturation value of 8.5 Nm. The transfer function for this filter is given by Eq. (4). The filtered trajectories used for each link can be seen in Fig. 1.

In each of the experiments, the controller closed-loop dynamics were set to be critically damped and given a bandwidth $\omega_n = 6.0$ rad/s. This yielded feedback gains of $K_p = 36 I_{2\times2}$ and $K_d = 12 I_{2\times2}$. For all experiments, the robot parameter estimates were initialized to $\hat{\theta}_1 = 0.2$, $\hat{\theta}_2 = 0.2$, and $\hat{\theta}_3 = 0.05$.

Filtering the original trajectory provided access to the desired velocity and desired acceleration signals corresponding to the filtered trajectory. In addition, the range of the filtered positions matched well with the achievable range of motion for each link. This resulted in a path that the robot was able to follow without encountering points of singularity or obstacles such as its own aluminium structure.

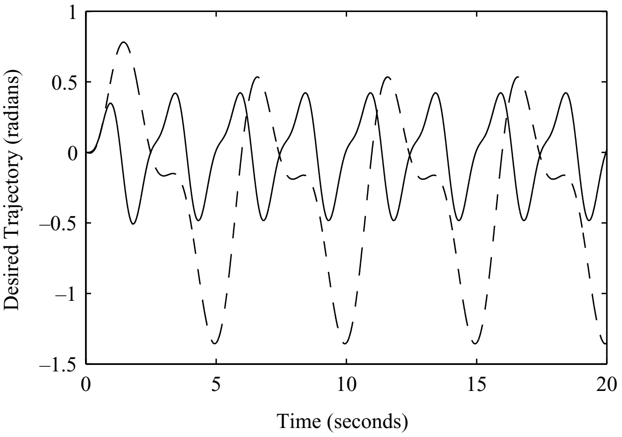

Fig. 1. Pre-filtered trajectories used with experiments on the direct-drive robot for link 1 (solid line) and link 2 (dashed line).

In the implementation of all controllers, a Proportional-Derivative (PD) controller was used initially to move the robot links close to their origin (0 rad). This was done to minimize initial position error for the adaptive controllers.

## 3. Adaptive control using a high-gain observer

In the case of adaptive control using a high-gain observer, Lee and Khalil[4] begin by designing a globally bounded state feedback controller and then introduce the observer to estimate position and velocity error. It is noted that such a high-gain observer may result in peaking in its transient behaviour.[4] This peaking could be transmitted to the manipulator, and could lead to instabilities. In order to overcome this, a modification is made which saturates the control inputs above some pre-defined level.

On the basis of the dynamic model Eq. (1), Lee and Khalil propose the control law

$$T = \hat{M}(q)\ddot{q}_d + \hat{C}(q, \dot{q}_r)\dot{q}_d + \hat{G}(q) - K_d \dot{e} - K_p e \quad (9)$$

for full-state feedback. The desired trajectory is given by $q_d$, an $n \times 1$ vector. Here, $e = q - q_d$, $\dot{q}_r = \dot{q} - \lambda e$, with $\lambda = \lambda_0/(1 + ||e||)$, $\lambda_0 > 0$. The notation $||e||$ is defined as $||e|| = (e^T e)^{1/2}$. $\hat{M}(\cdot)$, $\hat{C}(\cdot)$ and $\hat{G}(\cdot)$ represent estimates (based on robot parameter estimates) of each of the actual matrices. $K_d$ and $K_p$ are positive definite symmetric constant matrices. Their values will determine the response of the error dynamics.

The derivation of the adaptation law is based on Lyapunov stability and is given by Lee and Khalil.[4] This adaptation law requires *a priori* bounds on the parameter estimates. These bounds are represented by the set

$$\Theta = \{\theta | a_i \le \theta_i \le b_i, 1 \le i \le p\}$$

Now let

$$\Theta_\delta = \{\theta | a_i - \delta \le \theta_i \le b_i + \delta, 1 \le i \le p\}$$

where $\delta > 0$. The adaptation rule then contains a parameter projection feature to ensure that the parameter estimates remain in $\Theta_\delta$ whenever the initial parameter estimates are in $\Theta$. The adaptation rule is given as

$$\dot{\hat{\theta}}_i = \begin{cases} \gamma_{ii}\phi_i & \text{if } a_i < \hat{\theta}_i < b_i \text{ or} \\ & \text{if } \hat{\theta}_i \ge b_i \text{ and } \phi_i \le 0 \text{ or} \\ & \text{if } \hat{\theta}_i \le a_i \text{ and } \phi_i \ge 0 \\ \gamma_{ii}\left(1 + \dfrac{b_i - \hat{\theta}_i}{\delta}\right)\phi_i & \text{if } \hat{\theta}_i \ge b_i \text{ and } \phi_i \ge 0 \\ \gamma_{ii}\left(1 + \dfrac{\hat{\theta}_i - a_i}{\delta}\right)\phi_i & \text{if } \hat{\theta}_i \le a_i \text{ and } \phi_i \le 0 \end{cases} \tag{10}$$

where $\phi_i$ is the $i$th element of $\phi = -Y_r^{\mathrm{T}}(q, \dot{q}_r, \dot{q}_d, \ddot{q}_d)s$. $Y_r$ is the same function as $Y$ given in the linear regression form of the dynamics, but it has different arguments. See Lee and Khalil[4] for a more complete treatment. The vector $s$ is defined as $s = \dot{e} + \lambda e$. The scalar $\gamma_{ii}$ is the $i$th diagonal of a positive diagonal matrix $\Gamma$ which represents the adaptation gain. The $n \times 1$ vector $\hat{\theta}$ is the estimate of the parameter vector $\theta$.

The high-gain observer used to estimate the error dynamics will now be given. First, define $x_1 = e$ and $x_2 = \dot{e}$. The equations for the observer are given as

$$\dot{\hat{x}}_1 = \hat{x}_2 + \frac{1}{\epsilon}L_1(x_1 - \hat{x}_1) \tag{11}$$

$$\begin{aligned} \dot{\hat{x}}_2 = {} & \frac{1}{\epsilon^2}L_2(x_1 - \hat{x}_1) - \ddot{q}_{\mathrm{d}} - \hat{M}^{-1}(\hat{x}_1, q_{\mathrm{d}}) \\ & \times [\hat{C}(\hat{x}, q_{\mathrm{d}}, \dot{q}_{\mathrm{d}})(\hat{x}_2 + \dot{q}_{\mathrm{d}}) + \hat{G}(\hat{x}_1, q_{\mathrm{d}})] \\ & + \hat{M}^{-1}(\hat{x}_1, q_{\mathrm{d}})T^s(\hat{x}, q_{\mathrm{d}}, \dot{q}_{\mathrm{d}}, \ddot{q}_{\mathrm{d}}, \hat{\theta}) \end{aligned} \tag{12}$$

where $L_1 = \mathrm{diag}\{\alpha_{1i}\}$ and $L_2 = \mathrm{diag}\{\alpha_{2i}\}$, $i = 1, \ldots, n$. The high-gain of the observer comes from $\epsilon$, which is a small positive parameter. The values for matrices $L_1$ and $L_2$ are chosen such that,

$$\bar{A} = \begin{bmatrix} -L_1 & I \\ -L_2 & 0_{n\times n} \end{bmatrix}$$

is Hurwitz. $T^{\mathrm{s}}$ represents the saturated input torques, which are saturated at some pre-determined threshold.

In implementation of this algorithm, $\hat{e}$ and $\dot{\hat{e}}$ are used in the control and adaptation laws, replacing $e$ and $\dot{e}$, which would be used for full-state feedback control.

### 3.1. Simulation results for Lee and Khalil's method

The observer parameters $L_1$ and $L_2$ were both set equal to $I_{2\times 2}$ to ensure the matrix $\bar{A}$ is Hurwitz. The parameter $\epsilon$ which is directly related to observer gain was set to $\epsilon = 0.001$, as per one of the simulations performed by Lee and Khalil.[4] Initial position and velocity error were both set to zero, meaning that the robot starts at rest in its zero position. The adaptation gain $\Gamma$ was set to $I_{p\times p}$ for unity adaptation gain. Figure 2 illustrates the tracking error of the algorithm without any noise. It reaches a peak error of 0.2662 rad on link 1, and 0.1114 rad on link 2. The large peak error can be attributed to the relatively inaccurate initial estimates of the robot parameters, and a fairly small adaptation gain, which slows parameter convergence. The error in parameter estimation over time can be seen in Fig. 3. It is seen here that the parameters are relatively slow to converge. A small adaptation gain was chosen in order to compare this algorithm with the one presented in Section 4, which requires a small adaptation gain to prevent the simulation from diverging.

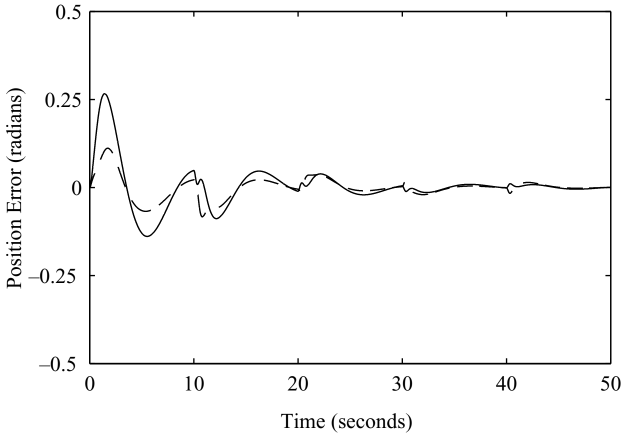

Fig. 2. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Lee and Khalil's method in simulation, with no noise on position measurements.

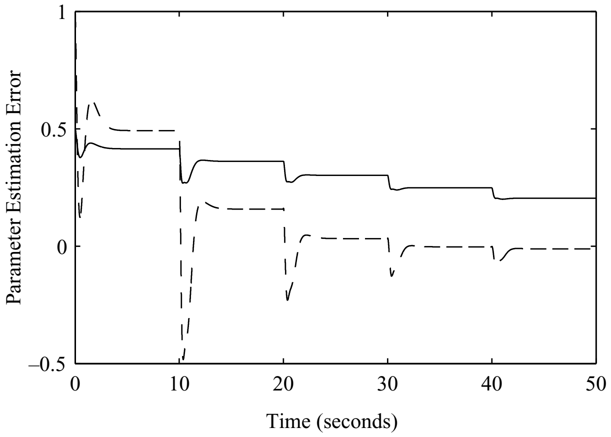

Fig. 3. Parameter estimation error using Lee and Khalil's adaptation law in simulation with no noise in the system, for $\theta_1$ (solid) and $\theta_2$ (dashed).

Adding noise to the position measurements to simulate a 12-b resolver resulted in comparable tracking performance, as seen in Fig. 4. Here the maximum tracking error was 0.2973 radians on link 1, and 0.1352 radians on link 2. However, due to the high-gain observer, velocity estimates were quite noisy, and this resulted in a very noisy control signal that could make implementation on a real platform difficult.

In order to better gauge the performance of the algorithm by itself, the adaptation gain was increased to $\Gamma = 50I$, and the results for tracking error are quite good. The much quicker adaptation of parameters resulted in significantly lower peak tracking error, at 0.0173 rad for link 1, and 0.0274 rad for link 2. However, when 12-b resolver noise was added to the position measurements with this value for $\Gamma$ the simulation diverged. This shows excellent theoretical performance, but performance that may not be achievable in practice.

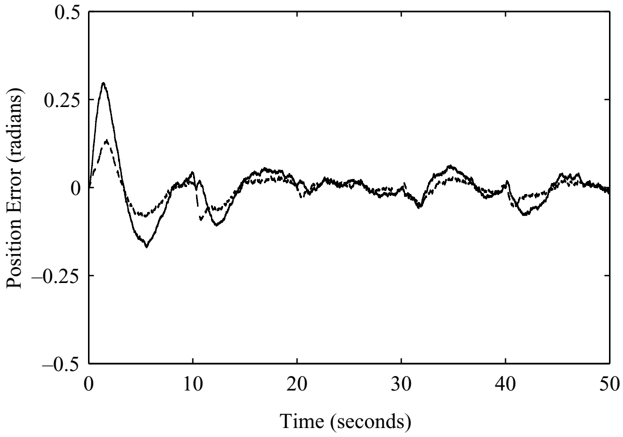

Fig. 4. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Lee and Khalil's method in simulation with position error equivalent to quantization by a 12-b resolver.

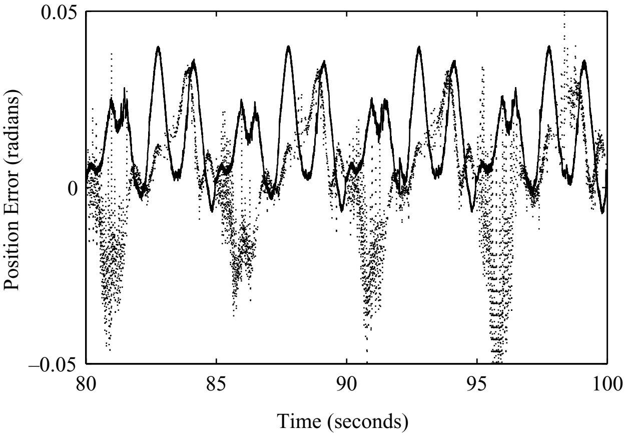

Fig. 5. Position error for joint 1 (solid line) and joint 2 (dotted line) after 80 s using Lee and Khalil's adaptation law with $\Gamma = 0.3$ on the direct-drive robot.

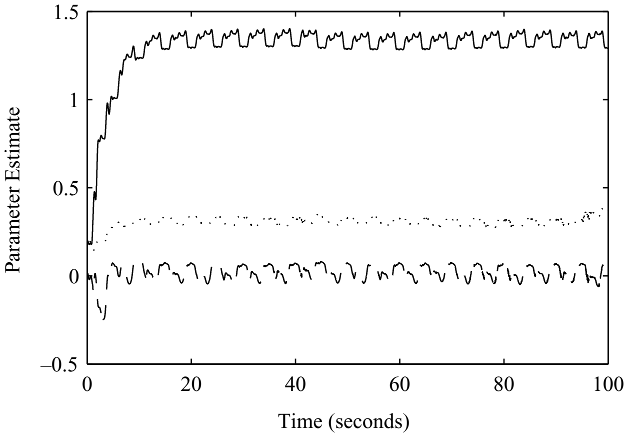

Fig. 6. Parameter estimates over time for $\theta_1$ (solid), $\theta_2$ (dotted) and $\theta_3$ (dashed) using Lee and Khalil's adaptation law with $\Gamma = 0.3$ on the direct-drive robot.

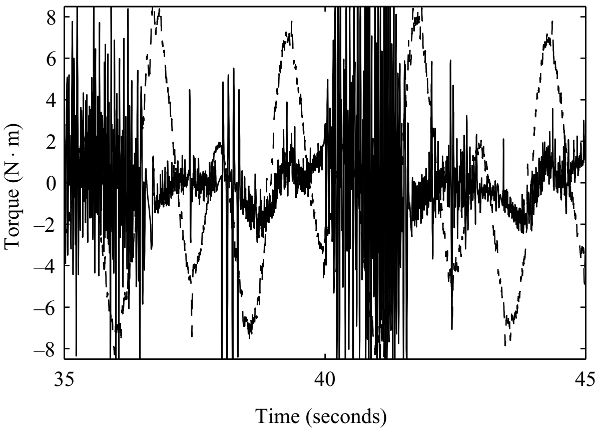

Fig. 7. Computed torques used to drive the robot links for link 1 (dashed) and link 2 (solid). These torques are computed using Lee and Khalil's control algorithm on the direct-drive robot for $\Gamma = 0.3$.

### *3.2. Experimental results for Lee and Khalil's method*

Several choices were made for the setting of the parameters of the algorithm when conducting the experiments. Given that a real platform is used, instead of simulated dynamics, control is lost over the noise processes in the system. As a result, some care had to be taken when parameters were selected.

As in simulation, the observer parameters $L_1$ and $L_2$ were both set equal to $I_{2\times 2}$ to ensure the matrix $\bar{A}$ is Hurwitz. The parameter $\epsilon$, related to the observer gain, was set to $\epsilon = 0.01$. This value is an order of magnitude larger than the value used for the simulations, however it is still one of the values used in Lee and Khalil's paper.[4] Such a value will also result in a high-gain observer.

Throughout the experiments, a variety of adaptation gains $\Gamma$ were used in order to see the effect of adaptation gain on controller performance.

Figure 5 illustrates the tracking performance of this controller for $\Gamma = 0.3$ after 80 s. The tracking error here is quite small. In this experiment, it peaks at 0.1325 rad for link 1, and −0.0785 rad for link 2. After the parameters have reached a steady-state, the tracking error is bounded between 0.040 and −0.007 rad for link 1. Link 2 error is bounded between 0.040 and −0.060 rad. Figure 6 shows convergence of the estimated parameters over time. Convergence happens rapidly, which helps to reduce tracking error more quickly. The high gain of the observer allows observer error to converge to a very small value quite rapidly. The small error in observed signals leads to small tracking error.

While this performance appears to be quite desirable, a problem is sometimes encountered in implementation. It was found that this algorithm excites unmodeled high-frequency dynamics in the robot. While the robot is running, significant vibrations are observed in its links. At times, the whole platform of the robot shakes. This can be seen graphically by examining the control torques produced by this algorithm in Fig. 7. At the points in the graph where the link 2 control torque (solid) saturates, the robot experiences much shaking. One can observe the high-frequency content of the link 2 control torque. The high-gain observer of this algorithm, which works to minimize observer error thus improving tracking performance, is also responsible for amplifying the system noise. This contributes to the high-frequency saturating control signal that excites the robot's high-frequency dynamics.

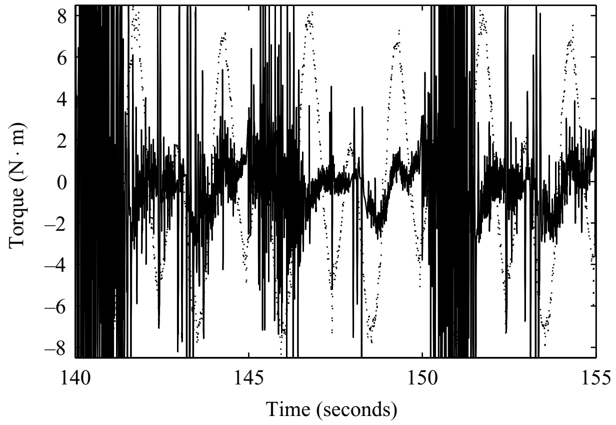

Fig. 8. Computed torques used to drive the robot links for link 1 (dashed) and link 2 (solid). These torques are computed using Lee and Khalil's control algorithm on the direct-drive robot for $\Gamma = 0.05$.

Experiments were performed with various values of $\Gamma$ to determine if the adaptation gain has an effect on the high-frequency excitations. Figure 8 shows the torques computed for an experiment with $\Gamma = 0.05$, much smaller than before. (When compared with the torques computed in other algorithms, such as Fig. 16 for Craig's algorithm with our observer and Fig. 23 for Gourdeau and Schwartz's algorithm, it should be noted that the torques computed for link 2 are much smaller in those cases, and do not approach the saturation point.) Again, though, the torque for link 2 saturates and contains high-frequency components. During this experiment the robot shook significantly, although tracking error remained quite good.

The cause of this high-frequency excitation appears to be due initially to a difference between measured and estimated positions. It is this error term that drives the observer, and is scaled by the high observer gains. Such disturbances as noise on the position measurements could be responsible for this behaviour. It is interesting to note that these vibrations occur primarily with link 2. The position resolver for the motor attached to link 2 of the direct-drive robot appears to produce slightly noisier position measurements than that of link 1. However, the other algorithms tested do not respond to the increased noise in this fashion. This demonstrates this algorithm's intolerance to noise on the measurements.

Another point of interest is that the occurrence of these high-frequency excitations is not predictable. The same experiment was run several times over, and in one instance the robot shook considerably, but in another the robot only vibrated slightly. The lack of predictability of these vibrations further complicates the implementation of this algorithm in practice. In one experiment, $\Gamma$ was set to $\Gamma = 2$ and left to run for a period of time. Beginning at 50 s, the computed torque for link 2 started saturating and the robot vibrated significantly. At 90 s into the experiment, the algorithm diverged. It is not sufficient for the robot to track the position well if it vibrates so significantly as to cause divergence of the experiment.

In another instance, $\Gamma$ was set to $\Gamma = 2$ again, and the experiment was run for 150 s. Figure 9 shows the tracking error for this run. At a steady state, the tracking error for link 1 was bounded between 0.042 and −0.015 rad. The tracking error for link 2 was bounded between 0.042 and −0.030 rad. Examining Fig. 10, it is apparent that, while the torques contain noise, there is little saturation observed. This translates to minimal vibration in the robot links as the experiment is run.

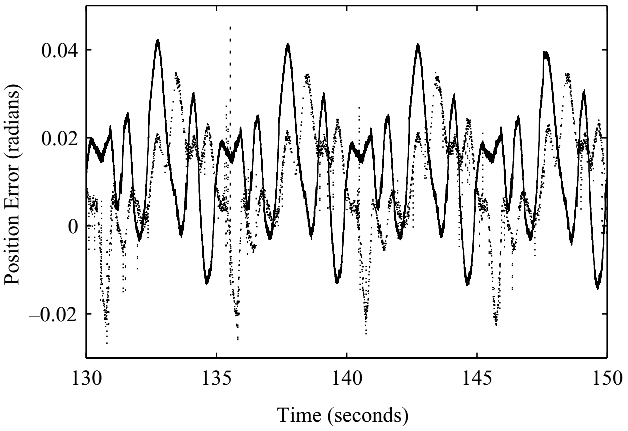

Fig. 9. Position error for joint 1 (solid line) and joint 2 (dotted line) after 130 s using Lee and Khalil's adaptation law with $\Gamma = 2$ on the direct-drive robot.

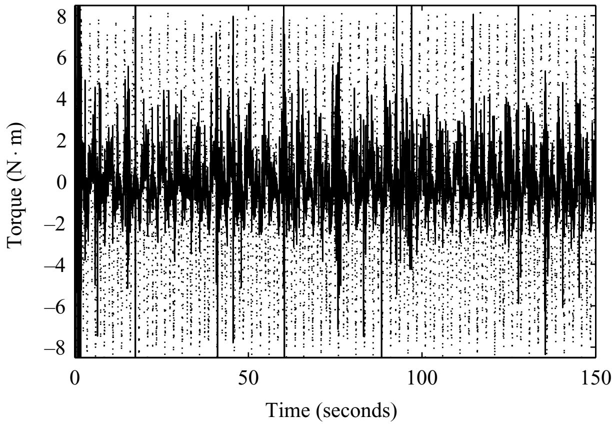

Fig. 10. Computed torques used to drive the robot links for link 1 (dashed) and link 2 (solid). These torques are computed using Lee and Khalil's control algorithm on the direct-drive robot for $\Gamma = 2$.

## 4. Adaptive control using Craig's algorithm and a Linear observer

The adaptive control method proposed by Craig, Hsu and Sastry[1] was implemented for comparison. However, instead of using measured position, velocity and acceleration of the robot a linear observer was constructed to feed back estimates of those signals, based on measured position. This observer has its poles placed so that they are much faster than the error dynamics, but small enough that the response to noise in the system will remain reasonable. The control law is given by

$$T = \hat{M}(q)\ddot{q}^{*} + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) \qquad (13)$$

where $\hat{M}(\cdot)$, $\hat{C}(\cdot)$ and $\hat{G}(\cdot)$ represent estimates of the mass matrix, coriolis matrix and vector of gravity terms. The term

$\ddot{q}^*$ is defined as

$$\ddot{q}^* = \ddot{q}_\mathrm{d} + K_\mathrm{v}\dot{E} + K_\mathrm{p}E \tag{14}$$

The servo error $E$ is an $n \times 1$ vector that is defined as

$$E = q_\mathrm{d} - q$$

which is negative to the way error is defined in Section 3. $K_\mathrm{v}$ and $K_\mathrm{p}$ are constant diagonal gain matrices,[1] which determine the location of the poles of the error dynamics.

For the full derivation of the adaptation law, see Craig, Hsu and Sastry's paper[1] and the paper by Schwartz, Warshaw and Janabi[10] The adaptation law is given as

$$\dot{\hat{\theta}} = \Gamma Y(q, \dot{q}, \ddot{q})^\mathrm{T} \hat{M}^{-1}(q) E_1 \tag{15}$$

where $E_1$ is the filtered servo error, and is given by

$$E_1 = \dot{E} + \Psi E$$

Here $\Psi$ is an $n \times n$ diagonal matrix, with $\psi_i > 0$ as entries along the diagonal. The $\psi_i$ are chosen to make the transfer function

$$\frac{s + \psi_i}{s^2 + k_{\mathrm{v}i}s + k_{\mathrm{p}i}}$$

strictly positive real (SPR). See Craig, Hsu and Sastry's paper[1] for further information about SPR systems.

Additionally, the adaptation law includes reset conditions on the parameters to ensure they lie within the bounds. This is similar in concept to the parameter projection principle of Lee and Khalil's adaptation law given by Eq. (10). The reset conditions are given as

$$\begin{aligned} \hat{\theta}_i(t^+) &= a_i, \quad \text{if } \hat{\theta}_i(t) \le a_i - \delta, \\ \hat{\theta}_i(t^+) &= b_i, \quad \text{if } \hat{\theta}_i(t) \ge b_i + \delta, \end{aligned} \tag{16}$$

After feedback linearization, using a computed torque method such as the one given, each link of the robot manipulator can be thought of as a double integrator.[6] We make use of the certainty equivalence principle in assuming perfect feedback linearization and decoupling based on the control defined in Eq. (13). On that basis we propose using a second-order linear observer for each joint. From this observer, position, velocity and acceleration will be estimated. In state-space form, the equation of the observer for the $i$th link can be expressed as

$$\begin{bmatrix} \dot{\hat{q}}_i \\ \ddot{\hat{q}}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{q}_i \\ \dot{\hat{q}}_i \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} (q_i - \hat{q}_i) \tag{17}$$

The values $K_1$ and $K_2$ represent the observer gains. For the input, $v$, to the observer, we use the same input as we give to the control law, that is $\ddot{q}^*$. By integrating the left-hand side of (17), the estimates of position and velocity are obtained. In order to obtain the acceleration estimate, a copy of the $\ddot{\hat{q}}_i$ term is kept. The control and adaptation laws can then be implemented with the estimates obtained.

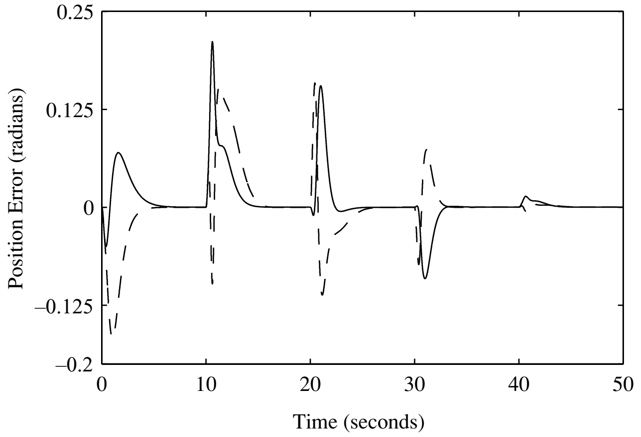

Fig. 11. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Craig's method with our observer in simulation, with no noise on position measurements.

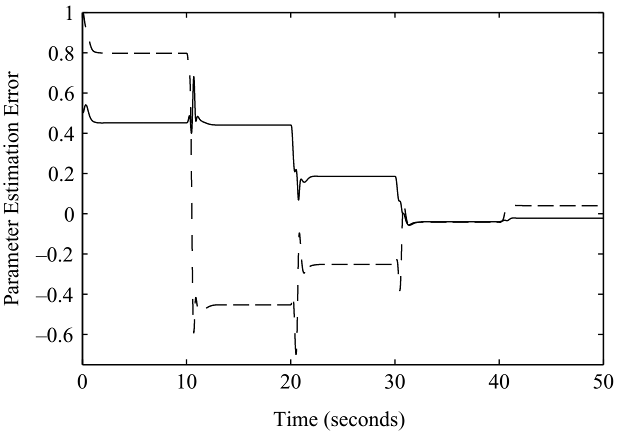

Fig. 12. Parameter estimation error using Craig's method with our observer in simulation with no noise in the system, for $\theta_1$ (solid) and $\theta_2$ (dashed).

One issue that presents itself with this method is the need to invert the mass matrix for the adaptation law. As the number of links increases in the robot manipulator, inversion of the mass matrix becomes increasingly computationally intensive.

### *4.1. Simulation results for Craig's method with a Linear observer*

Figure 11 represents the tracking error for this method when implemented in simulation without any measurement noise added. The value for $\Psi$ was set to $\Psi = I$. The observer gains were set to $K_1 = 20$ and $K_2 = 300$, placing the poles for the observer at $s = -10 \pm 10\sqrt{2}j$. This is much faster than the dynamics of the closed-loop system, but the gains are significantly smaller than those used for Lee and Khalil's method in Section 3.

The tracking error for this simulation reached maximum values of 0.2115 rad for link 1, and 0.1676 rad for link 2, with no noise in the system. The error on the parameter estimates can be seen in Fig. 12. The adaptation gain was set to $\Gamma = I$, values much larger than this resulted in divergence of the

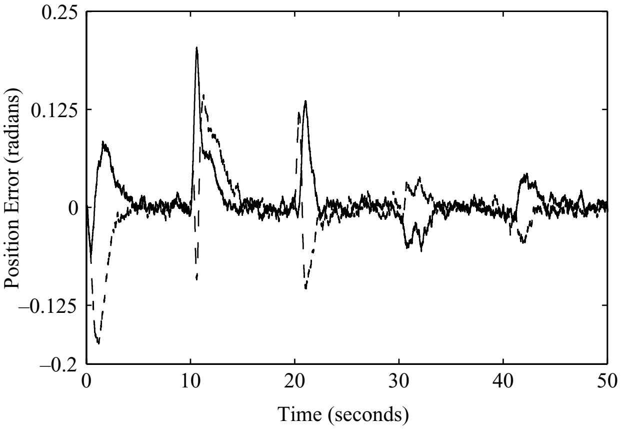

Fig. 13. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Craig's method with our observer in simulation, with position error equivalent to quantization by a 12-b resolver.

simulation. It can be seen that after 50 s of simulation, the parameter estimates are both quite close to their true values.

When noise was added to the system to simulate 12-b quantization error, the maximum error did not change significantly, and while the error became much noisier, the results were comparable. This is illustrated in Fig. 13. In this case, the maximum tracking error was 0.2040 rad for link 1, and 0.1744 rad for link 2. The computed torque signal resulting from the controller contained some noise, but it was much smaller than the torques computed in Section 3 for noisy position measurements.

### 4.2. Experimental results for Craig's method with a Linear observer

In the experiments with this algorithm on the direct-drive robot, the parameters were chosen to allow a good comparison with the other algorithms. The controller feedback gains were set as outlined in Section 2. For the experiments, the observer gains were set to $K_1 = 20$ and $K_2 = 500$, placing the observer poles at $s = -10 \pm 20j$. These poles are much faster than the dynamics of the closed-loop system. It was determined experimentally that using underdamped observer poles allowed for smaller observer gains and yielded performance comparable to a set of critically damped observer poles placed farther into the negative real portion of the $s$-plane. An experiment was performed with observer poles at $s = -25$, giving observer gains of $K_1 = 50$ and $K_2 = 625$, and observer error was not significantly different than with the observer poles placed at $s = -10 \pm 20j$. For all experiments, the value of $\Psi$ was set to $\Psi = I$. Several values were used for the adaptation gain $\Gamma$ to compare performance.

Figure 14 shows the tracking error for this method when implemented with $\Gamma = 0.1$. When run experimentally, values for $\Gamma$ any larger than this result in unacceptable performance of the robot during the experiment. The tracking error in this case reached maximum values of $-0.3810$ rad for link 1, and $-0.3583$ rad for link 2. After convergence of the parameter estimates, the error remained bounded between 0 and $-0.045$ rad for link 1. For link 2, the tracking error remained bounded between 0.080 and $-0.066$ rad. The parameter estimates converged to steady-state values as seen in Fig. 15.

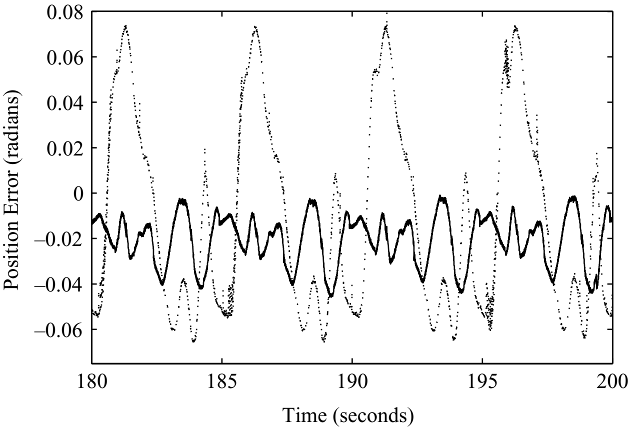

Fig. 14. Position error for joint 1 (solid line) and joint 2 (dotted line) after 180 s using Craig's method with our observer, with $\Gamma = 0.1$ on the direct-drive robot.

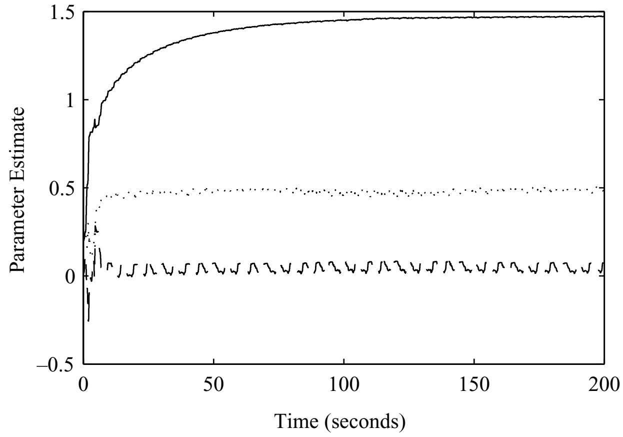

Fig. 15. Parameter estimates over time for $\theta_1$ (solid), $\theta_2$ (dotted) and $\theta_3$ (dashed) using Craig's method with our observer, with $\Gamma = 0.1$ on the direct-drive robot.

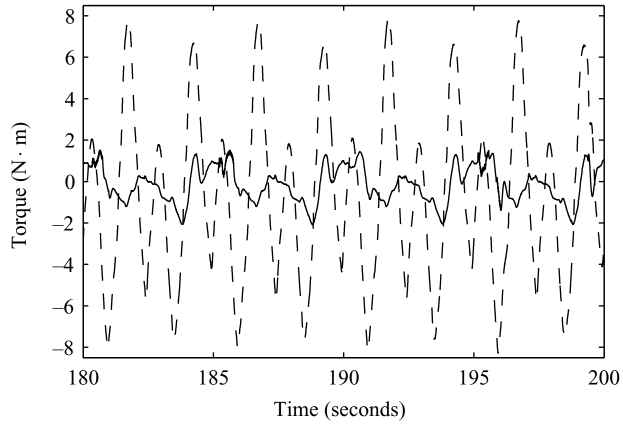

Fig. 16. Computed torques used to drive the robot links for link 1 (dashed) and link 2 (solid). These torques are computed using Craig's method with our observer on the direct-drive robot.

It is interesting to note the quality of the torque signals used to command the robot. From Fig. 16 it is apparent that the computed torques do not contain large quantities of noise. It

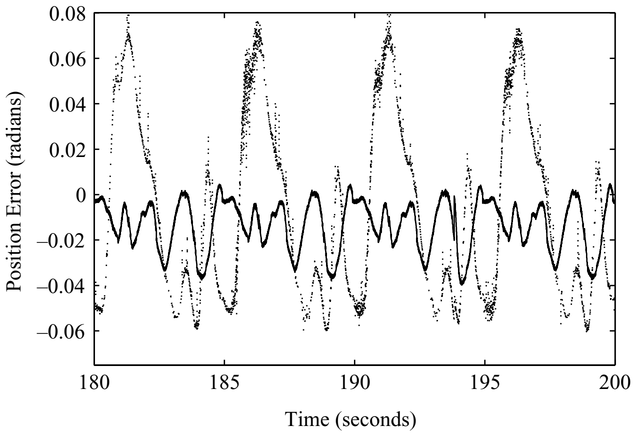

Fig. 17. Position error for joint 1 (solid line) and joint 2 (dotted line) after 180 s using Craig's method with our observer, with $\Gamma = 0.05$ on the direct-drive robot.

is this characteristic that allows smooth tracking of the links of the robot, without causing vibrations as it travels.

While the performance of this experiment is quite good, the choice of $\Gamma$ is close to the largest possible value before unacceptable performance is observed. As a result, another experiment was performed with an adaptation gain of $\Gamma = 0.05$, half of the previous value. The tracking error for this experiment is shown in Fig. 17. In this case, the tracking error reached maximum values of $-0.4234$ rad for link 1, and $-0.3282$ rad for link 2. After 200 s, the tracking error remained bounded between 0.005 and $-0.040$ rad for link 1. For link 2, the tracking error remained bounded between 0.080 and $-0.060$ rad. This is a positive result, since very similar performance is achieved over time with a lower adaptation gain. Using this lower adaptation gain keeps the system much further from the point of divergence of the experiment.

## 5. Adaptive control using an extended Kalman filter

A third method of adaptive output feedback control is to use an EKF. This method is developed by Gourdeau and Schwartz.[5] A notable difference between this method and those presented in Section 3 and Section 4 is that here the EKF is used to estimate unknown robot parameters, as well as position and velocity. This method effectively combines the functionality associated with the adaptation laws and observers of the previous methods into one EKF. An advantage of this method over the previous methods is that the theory behind Kalman filtering provides guidelines for setting the design parameters of the controller.[5]

The control law used in this algorithm is the same as that given in Eqs. (13) and (14).

The use of an EKF requires linearizing the nonlinear system model about the estimate of the states, using a first-order Taylor series expansion. In order to begin that process, a nonlinear state vector is defined as $x = [q^{\mathrm{T}}\dot{q}^{\mathrm{T}}\theta^{\mathrm{T}}]^{\mathrm{T}}$ and its derivative is defined as $\dot{x} = f(x, u)$ where $u(t) = T$, the input torque. This yields

$$\dot{x} = f(x, u) = \begin{bmatrix} \dot{q} \\ \ddot{q} \\ 0 \end{bmatrix} \tag{18}$$

where $\ddot{q}$ is obtained by rearranging the terms of (1) to give

$$\ddot{q} = M^{-1}(q)(T - C(q, \dot{q})\dot{q} - G(q)) \tag{19}$$

Note that the vector $\theta$ is constant so its derivative $\dot{\theta}$ is zero. In order to account for input disturbances and parameter variations, the model may be extended as follows

$$\dot{x} = f(x(t), t) + G(x(t), t)w(t) \tag{20}$$

$$G(x) = \begin{bmatrix} 0 & 0 \\ M^{-1}(q, \theta) & 0 \\ 0 & I \end{bmatrix} \tag{21}$$

$$w = \begin{bmatrix} w_1^T & w_2^T \end{bmatrix}^{\mathrm{T}} \tag{22}$$

where $w_1$ and $w_2$ are random variables. The $w_1$ term is the noise associated with the input to create a disturbance, and $w_2$ is the noise associated with $\dot{\theta}$ to cause parameter variation. The system measurements may be described as

$$z(t) = Hx(t) + v(t) \tag{23}$$

$$H = [I \quad 0 \quad 0] \tag{24}$$

for the case of only using position measurements. To implement the EKF, a perturbation model is needed. The reader is referred to Gourdeau and Schwartz's work[5] for derivation of the perturbation model. The equations of dynamics for the EKF are given as

$$\dot{\hat{x}}(t) = f(\hat{x}(t), t) + P(t)H^{\mathrm{T}}R^{-1}(t)\{z(t) - H\hat{x}(t)\} \tag{25}$$

$$\begin{aligned}\dot{P}(t) = {} & \frac{\partial f}{\partial \hat{x}}P(t) + P(t)\frac{\partial f}{\partial \hat{x}}^{\mathrm{T}} + G(x(t), t)Q(t)G^{\mathrm{T}}(x(t), t) \\ & - P(t)H^{\mathrm{T}}R^{-1}(t)HP(t)\end{aligned} \tag{26}$$

where $\hat{x}(t)$ is the estimated state vector, and $P(t)$ is the error covariance matrix. The matrix $R(t)$ is a matrix of measurement variances.[11] In the case of an $n$-link manipulator, there would be $n$ position measurements, and $R(t)$ would have dimensions $n \times n$. $Q(t)$ is made up of two diagonal matrices, $Q_1(t)$ and $Q_2(t)$. The matrix $Q_1(t)$ represents the confidence in the dynamic model and has dimensions $n \times n$, while $Q_2(t)$ represents the speed at which the parameter vector is estimated to vary and has dimensions $p \times p$.[11]

### 5.1. Simulation results for adaptive control using an extended Kalman filter

To simulate the EKF, the filter parameters were set as $R(t) = 0.0001I$, $Q_1(t) = 0.01I$, $Q_2(t) = 0.0025I$ and $P(0) = \mathrm{diag}\{0.0001, 0.0001, 0.0001, 0.0001, 0.01, 0.01\}$. These parameters used for simulation were chosen based on the values used in Gourdeau and Schwartz's paper,[5] except that the initial error covariance for the parameter estimates is set larger in this case. This is due to greater initial error on the parameter estimates in this simulation, than in their paper.

Overall, the results for the simulation were good, with the exception of the peak tracking error in the transient period. Results for the tracking error in a noise-free simulation are

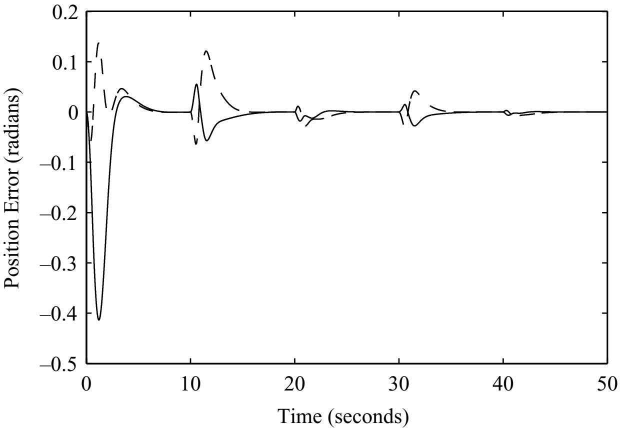

Fig. 18. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Gourdeau and Schwartz's method in simulation, with no noise on position measurements.

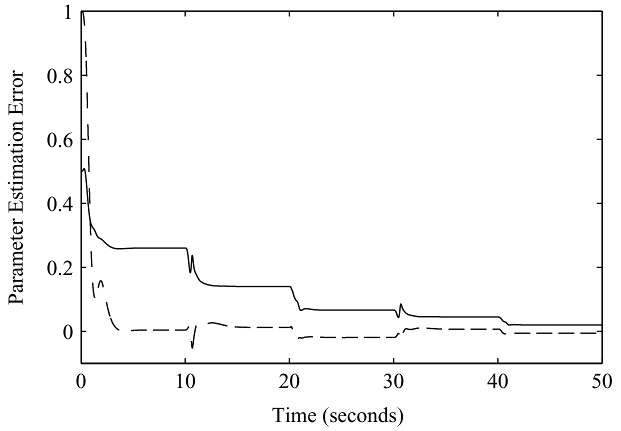

Fig. 19. Parameter estimation error using Gourdeau and Schwartz's EKF in simulation with no noise in the system, for $\theta_1$ (solid) and $\theta_2$ (dashed).

illustrated in Fig. 18. The maximum tracking error in this case is 0.4135 rad for link 1, and 0.1372 rad for link 2. The maximum tracking error on link 1 is much larger than in the other cases, and is due to large initial error on the observed velocity signals. However, the error is quick to decrease and enter the range of error of the other algorithms. Also, the parameters are quick to converge using this approach. Figure 19 shows the error in the parameter estimates.

When noise was added to the position measurements to simulate 12-b quantization error, the results were similar to the noise-free case. Figure 20 demonstrates the tracking error of the robot in this case. The maximum tracking error was 0.4208 rad for link 1, and 0.1436 rad for link 2. An important note here is that when noise was introduced, the control signal generated contained much less noise than in the case of the other two algorithms. This is an important fact for implementation, since a control signal as free of noise as possible is desired to drive the motors, so as not to excite high frequency and unmodeled dynamics.

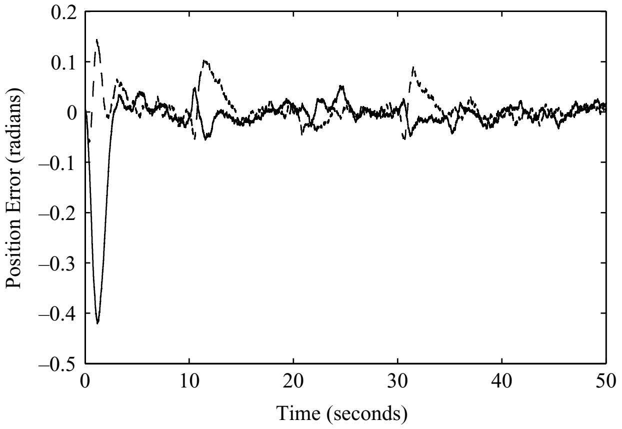

Fig. 20. Error between desired trajectory and simulated trajectory for joint 1 (solid line) and joint 2 (dashed line) using Gourdeau and Schwartz's method in simulation with position error equivalent to quantization by a 12-b resolver.

### *5.2. Experimental results for adaptive control using an extended Kalman filter*

When running the adaptive controller based on the EKF on the direct-drive robot, it was found that proper setting of the filter parameters was crucial to correct operation of the EKF. Without proper setting of the filter parameters, a divergence in the error on the estimated values of the state is observed. Since the dynamic model being used is an assumed model of reality, it is possible that some discrepancy between the assumed model and the true dynamics is responsible for this divergence in error.[12]

In order to ensure proper operation of the EKF, the matrix $Q_1(t)$ must be set correctly. This matrix represents the magnitude of disturbances cause by unmodeled dynamics.[11] This effectively relates to the confidence in the dynamic model. A small value of $Q_1(t)$ suggests a high confidence in the model. However, in these experiments such a small value caused a divergence in estimation error. As the value of $Q_1(t)$ was increased, the EKF achieved increasingly better performance.

The experiment was run with the filter parameters set to $R(t) = 0.0002I$, $Q_1(t) = 0.8I$, $Q_2(t) = 0.001I$ and $P(0) = 0.1I$. The choice of these variances can be a challenge and is one of the drawbacks of the method. We base our design on a trial and error tuning process. These parameters were chosen experimentally in order to yield satisfactory results. The tracking error from this experiment can be seen in Fig. 21. In this case, the maximum tracking error reached −0.4942 rad for link 1, and −0.4187 rad for link 2. Once the parameter estimates had converged, the tracking error remained bounded between 0.018 and −0.130 rad for link 1. The steady-state error bounds for link 2 were 0.140 and −0.140 rad. The convergence of estimated robot parameters can be seen in Fig. 22. Note that it was possible to achieve faster parameter convergence with this algorithm than when using Craig's algorithm with our observer, which must use lower adaptation gain in order to prevent divergence of the experiment. As such, a steady state with Gourdeau and Schwartz's algorithm is reached after a shorter period of time.

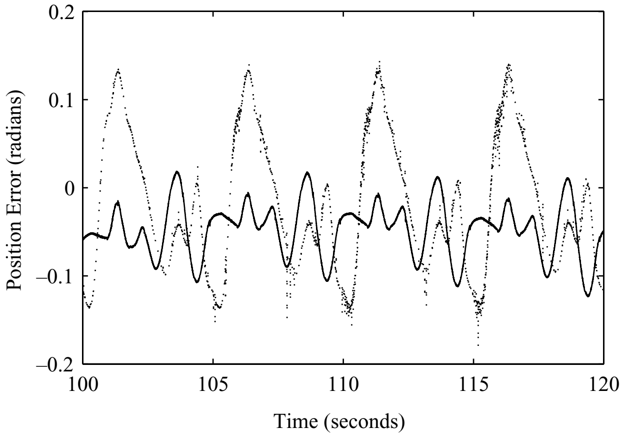

Fig. 21. Position error for joint 1 (solid line) and joint 2 (dotted line) after 100 s using Gourdeau and Schwartz's EKF on the direct-drive robot.

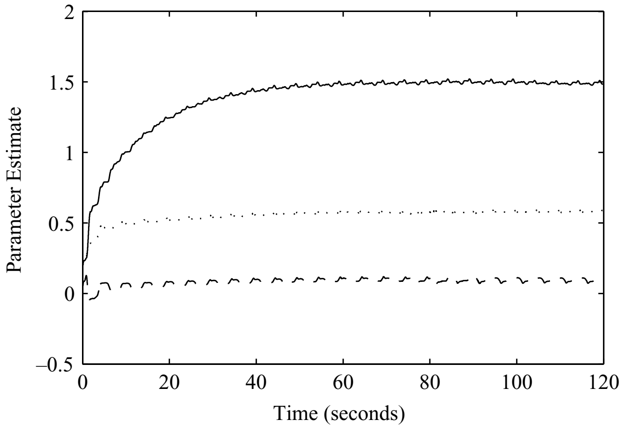

Fig. 22. Parameter estimates over time for $\theta_1$ (solid), $\theta_2$ (dotted) and $\theta_3$ (dashed) using Gourdeau and Schwartz's EKF on the direct-drive robot.

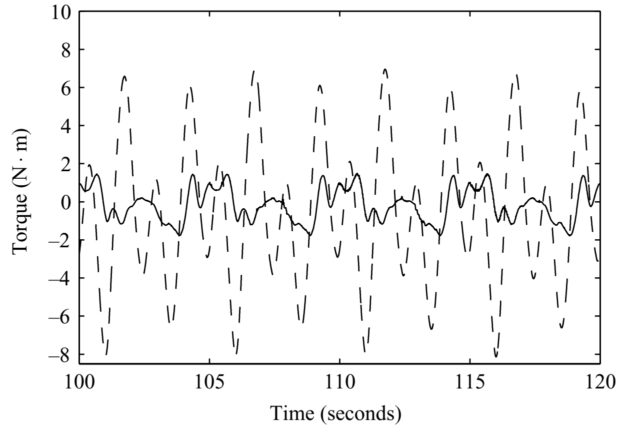

Fig. 23. Computed torques used to drive the robot links for link 1 (dashed) and link 2 (solid). These torques are computed using Gourdeau and Schwartz's control algorithm on the direct-drive robot.

Figure 23 shows the control torque produced by this algorithm. It is apparent that this algorithm produces smooth control inputs to command the robot, resulting in minimal robot vibration, and smooth travel of the links. However, it is important to note that this algorithm yields much higher tracking error than in the other algorithms tested. This is likely due to the larger estimation error with the EKF than with the observers of other algorithms. For link 1, the error on the position estimate at steady state is bounded by 0.030 and −0.006 rad. The error on the position estimate for link 2 at steady state is bounded by 0.020 and −0.025 rad. In comparison, the observer error when using Craig's algorithm with our observer (run with $\Gamma = 1$) is bounded by 0.007 and −0.003 rad for link 1 at steady state. For link 2, Craig's algorithm with our observer yields bounds on the observer error as 0.015 and −0.015 rad.

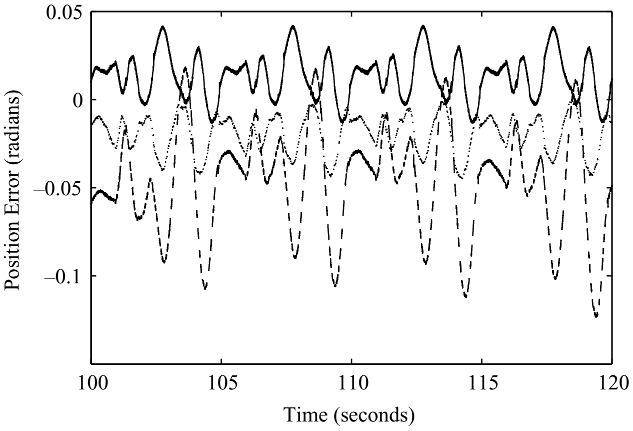

Fig. 24. Link 1 position error for each of the three algorithms studied – Lee and Khalil's algorithm (solid line), Craig's algorithm with our observer (dotted line) and Gourdeau and Schwartz's algorithm (dashed line). All results are from the direct-drive robot.

In Gourdeau and Schwartz's algorithm, as $R(t)$ is decreased, the performance of the EKF improves – the error on the position and velocity estimates decreases, as does tracking error. However, the value of $R(t) = 0.0002I$ used in this experiment is close to the lower limit of achievable values of $R(t)$ before the experiment diverges. As a result, the tracking error presented here is the smallest that has been achieved in experiments on the direct-drive robot for this algorithm.

## 6. Conclusion

Three methods of output feedback adaptive control have been presented in this paper. The systems of equations for each of the methods have been given. All methods were demonstrated in simulation using two-degree-of-freedom serial manipulator dynamics, in both noise-free simulations and simulations involving 12-b quantization error. Experiments for all of the methods were also performed using the Carleton University direct-drive robot, a two-degree-of-freedom manipulator operating in the horizontal plane.

Figure 24 shows the position error on link 1 of the direct-drive robot over time for each of the three algorithms studied. Figure 25 shows the position error on link 2 of the direct-drive robot over time for the same algorithms. From these graphs it is clear that Lee and Khalil's algorithm consistently demonstrates the lowest tracking error. Craig's algorithm with our observer follows Lee and Khalil's algorithm in tracking performance, with slightly greater tracking error.

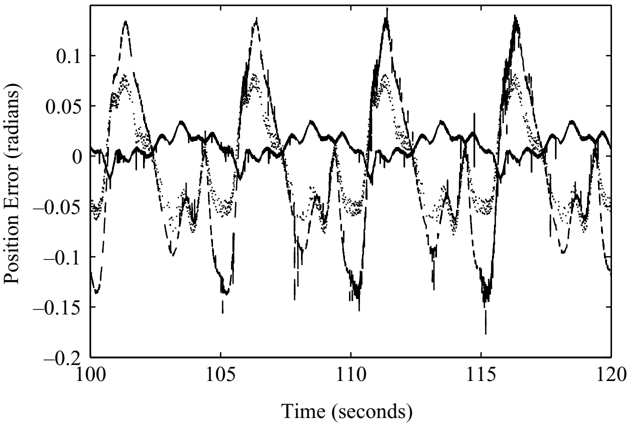

Fig. 25. Link 2 position error for each of the three algorithms studied – Lee and Khalil's algorithm (solid line), Craig's algorithm with our observer (dotted line) and Gourdeau and Schwartz's algorithm (dashed line). All results are from the direct-drive robot.

Gourdeau and Schwartz's method shows the largest tracking error in both of the robot links.

Lee and Khalil's method is an excellent theoretical approach. The high observer gain allows fast convergence of the observer error, and it is quick to approach performance of full-state feedback control.[4] With a very large adaptation gain of $\Gamma = 50I$ this method yielded the smallest peak trajectory error in simulation, and quick convergence of parameters to their true values. However, use of the high-gain observer makes this system quite sensitive to noise. In the presence of noise, the simulations diverge for large values of $\Gamma$, limiting the achievable performance of the algorithm. Also, the significant error on the observed signals contributed to a very noisy control signal that is likely to present difficulty in practical implementations of the algorithm. The experimental results confirmed this, as the noisy control signals contributed to significant vibrations in the robot links. The high frequencies in the control signal excite unmodeled system dynamics. While tracking performance is very good in the experiments, the problem of vibrating links limits the practical usefulness of this algorithm.

The method of adaptive control proposed by Craig, with the observer that we propose, yielded results in simulation comparable to Lee and Khalil's for the case of small $\Gamma$. The performance of this method without noise was not as desirable as Lee and Khalil's. The tracking error for this method took significantly longer to decay, and reached similar peak values. However, when noise was introduced in both cases, this method yielded comparable results and, importantly, a much less noisy control signal. This is of concern for implementation on a real robot. Also, the adaptation law caused parameter error to converge towards zero in a quicker period of time than with Lee and Khalil's method using $\Gamma = 1$. In experimentation, results were very positive for this algorithm. It yielded small tracking error and fairly clean control signals. While the tracking error was not as small as that of Lee and Khalil's algorithm, it was much smaller than that of Gourdeau and Schwartz's algorithm. The parameter estimates converged to a steady state. However, it is important to note that in order to prevent divergence, the adaptation gain cannot be set arbitrarily large.

Finally, implementation of Gourdeau and Schwartz's Extended Kalman Filter for estimation of position, velocity and robot parameters also yielded promising results in simulation. While in all cases, the initial tracking error was much larger than the other methods, the error decayed faster than in the other approaches. Also, the robot parameter estimates were quick to approach their true values. Introduction of noise did not have a significant effect on tracking error, but it should be noted that the control signal that resulted in this case was noticeably cleaner than with the other methods. This is a very desirable result for practical implementation. However, when implemented in practice, this method had larger tracking error than either of the others. This could be due to inaccuracy of the robot model with respect to the real world, and the sensitivity of the EKF to that discrepancy.[12] However, robot parameter convergence was quite quick with this method, bringing the tracking error to its steady-state value fairly quickly.

Having examined each of the algorithms both in simulation and through experiments on the Carleton University direct-drive robot, an idea of the relative performance of each algorithm has been obtained. While Lee and Khalil's method yields the lowest tracking error, it is very sensitive to noise and excites the high-frequency dynamics of the robot. Gourdeau and Schwartz's method does not yield low tracking error, but its ability to estimate the parameters is quite good. Craig's method, with the linear observer we propose, yields low tracking error and good parameter convergence, while producing a fairly clean control signal.

## References

1. J. J. Craig, P. Hsu and S. S. Sastry, "Adaptive control of mechanical manipulators," *Int. J. Robot. Res.* **6**(2), 16–27 (1987).
2. J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.* **6**(3), 49–59 (1987).
3. H. Hajjir and H. M. Schwartz, "An Adaptive Nonlinear Output Feedback Controller for Robot Manipulators," *Proceedings of the American Control Conference*, San Diego, California (Jun. 1999) vol. **3**, pp. 1520–1524.
4. K. W. Lee and H. K. Khalil, "Adaptive output feedback control of robot manipulators using high-gain observer," *Int. J. Control* **6**, 869–886 (1997).
5. R. Gourdeau and H. M. Schwartz, "Adaptive control of robotic manipulators using an extended Kalman filter," *J. Dyn. Syst. Meas. Control* **115**, 203–208 (Mar. 1993).
6. H. M. Schwartz, "Model reference adaptive control for robotic manipulators without velocity measurements," *Int. J. Adapt. Control Signal Process.* **8**, 279–285 (1994).
7. R. Gourdeau and H. M. Schwartz, "Adaptive control of robotic manipulators: Experimental results," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacra mento, California (Apr. 1991) pp. 8–15.
8. G. D. Warshaw, Investigations of Adaptive Control for a Direct Drive Robotic Manipulator *Master's Thesis* (Ottawa, Ontario: Carleton University, Jan. 1990).
9. G. Warshaw, Sampled-Data Robot Adaptive Control *PhD Dissertation* (Ottawa, Ontario: Carleton University, Jan. 1994).
10. H. M. Schwartz, G. Warshaw and T. Janabi, "Issues in robot adaptive control," *Proceedings of the American Control Conference*, San Diego, California, (1990) pp. 2797–2805.
11. R. Gourdeau, Adaptive Control of Robotic Manipulators *PhD Dissertation* (Ottawa, Ontario: Carleton University, Apr. 1991).
12. A. Gelb, ed., *Applied Optimal Estimation* (MIT Press, Cambridge, MA, 1974).