Submitted to *IEEE Transactions on Control Systems Technology*

Adaptive Robust Manipulator Trajectory Control: An Application of Unfalsified Control

Tung-Ching Tsao *Member, IEEE* and Michael G. Safonov *Fellow, IEEE*

This work was supported in part by the U.S. Air Force Office of Scientific Research under grant F49620-95-I-0095 and in part by a grant from NSC Taiwan, ROC.

A preliminary version of this paper was presented at ICARCV '96, Singapore [14].

T. C. Tsao was with the Department of Electrical Engineering, Chinese Naval Academy, Kaohsiung, Taiwan, ROC. He is now with Spectrum Astro, Inc., 3601 Aviation Blvd., Manhattan Beach, CA 90266

Michael G. Safonov is with Dept. of Electrical Engineering—Systems, University of Southern California, Los Angeles, CA 90089–2563 USA. Phone 1-213-740-4455. FAX 1-213-740-4449. Email msafonov@usc.edu.

rev. September 4, 1997

Abstract

This paper describes an application of unfalsified control theory to the design of an adaptive controller for a nonlinear robot manipulator. A nonlinear 'computed torque' control structure is employed. Four parameters representing unknown masses, inertias and other dynamical coefficients are adaptively adjusted in real-time using an linear programming technique to optimally satisfy control-law unfalsification conditions. Simulations show that the method yields significantly more precise and rapid parameter adjustments than conventional continuous parameter update rules, especially when the manipulator arm is subject to sudden random changes in mass or load properties.

I. Introduction

Unfalsified control theory [9], [10], [1] is primarily a plant-representation-free approach to controller identification, though it can benefit from plant models if available. Theoretically, unfalsified control theory provides a precise characterization of the increment in control-relevant knowledge in each new measurement, allowing robust adaptive control of both linear and nonlinear systems. But design experience with the unfalsified control is as yet very limited, and there is a need for design studies both to illustrate the use of the theory and, more importantly, to more clearly identify implementation issues that may prove to be of engineering importance.

Like the control-oriented identification methods of [4], [5], [6], [8], unfalsified control involves the testing of classes of candidate models, performance goals, and open-loop measurement data for mutual consistency. Identification is regarded as a winnowing process in which one discards so-called 'falsified' models that fail this consistency test. The distinguishing feature of unfalsified control is that the 'models' being tested are actually models of control laws and the performance goals are closed-loop control performance criteria. Safonov and Cabral [2] have shown that even this distinction fades when control-oriented identification and unfalsified control are viewed from Willems' [16] behavioral system perspective.

This paper takes a robot manipulator as an example to demonstrate how *a priori* mathematical knowledge (nonlinear plant models and uncertainty bounds) can be integrated into the essentially empirical unfalsified control theory to form a procedure that takes into consideration both prior and posterior information. The paper is organized as follows. Section II briefly reviews some mathematical results about robot manipulators and their control. Section III summarizes the key results of unfalsified control theory taken from [10]. Section IV describes how *a priori* mathematical knowledge can be merged with data in unfalsified control theory to design a robust adaptive controller. Computer simulations are provided in Section V. Finally, discussion and conclusions are in Section VI.

II. Background: Mathematical Knowledge about Manipulators

The dynamics of an ideal rigid-link manipulator can be described by the following equation,

$$H(\theta, q)\ddot{q} + C(\theta, q, \dot{q})\dot{q} + g(\theta, q) = u_a, \tag{1}$$

in which q is a real n-vector representing the rotational angles of the n links of the manipulator arm; $H(\theta, q)$ is the inertia matrix; $C(\theta, q, \dot{q})\dot{q}$ accounts for the joint friction, coupling Coriolis and centripetal forces; $g(\theta, q)$ is the torque caused by gravity; and u_a is a real n-vector whose elements are joint torques consisting of actuator outputs and external disturbances. Physical considertions ensure that the $H(\theta, q)$ is always a positive definite matrix.

Planar Two-Link Manipulator Example [13] The rigid-body dynamics of the planar, two-link manipulator shown in Figure 1 can be written in form of equation (1) with

$$\begin{aligned} H(\theta, q) &= H^T(\theta, q) = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \\ C(\theta, q, \dot{q}) &= \begin{bmatrix} -h\dot{q}_2 & -h(\dot{q}_1 + \dot{q}_2) \\ h\dot{q}_1 & 0 \end{bmatrix} \end{aligned}$$

where

$$\begin{aligned} H_{11}(\theta) &= \theta_1 + 2\theta_3 \cos q_2 + 2\theta_4 \sin q_2 \\ H_{12}(\theta) &= H_{21}(\theta) = \theta_2 + \theta_3 \cos q_2 + \theta_4 \sin q_2 \\ H_{22}(\theta) &= \theta_2 \\ h(\theta) &= \theta_3 \sin q_2 - \theta_4 \cos q_2 \\ \theta &= [\theta_1, \theta_2, \theta_3, \theta_4]^T \end{aligned}$$

and

$$\begin{aligned} q &= \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\ u_a &= \begin{bmatrix} u_{a1} \\ u_{a2} \end{bmatrix} \\ \theta_1 &= I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2 \\ \theta_2 &= I_e + m_e l_{ce}^2 \\ \theta_3 &= m_e l_1 l_{ce} \cos \delta_e \\ \theta_4 &= m_e l_1 l_{ce} \sin \delta_e . \end{aligned}$$

Here parameters with subscript 1 are related to link 1 and parameters with subscript e are related to the combination of link 2 and end effector; for example, I_1 is the inertia of link 1, m_1 is the mass of link 1, and l_{c1} indicates location of link 1 mass center, et cetera (cf. [13]). For simplicity, we shall assume that the gravity term is zero, corresponding to the case in which the manipulator arm moves only in the horizontal plane.

In view of the foregoing relations, the dynamic equation (1) for the manipulator in Figure 1, can be rewritten as

$$Y(q,\dot{q},\ddot{q})\theta + g(\theta,q) = u_a \tag{2}$$

in which $\theta \overset{\triangle}{=} [\theta_1,\theta_2,\theta_3,\theta_4]^T$ and $Y(\cdot)$ is a 2×4 matrix with elements

$$\begin{aligned}
Y_{11} &= \ddot{q}_1, \quad Y_{12} = \ddot{q}_2, \quad Y_{21} = 0, \quad Y_{22} = \ddot{q}_1 + \ddot{q}_2 \\
Y_{13} &= (2\ddot{q}_1 + \ddot{q}_2)\cos q_2 - (2\dot{q}_2\dot{q}_1 + \dot{q}_2^2)\sin q_2 \\
Y_{14} &= (2\ddot{q}_1 + \ddot{q}_2)\sin q_2 + (2\dot{q}_2\dot{q}_1 + \dot{q}_2^2)\cos q_2 \\
Y_{23} &= \ddot{q}_1\cos q_2 + \dot{q}_1^2\sin q_2 \\
Y_{24} &= \ddot{q}_1\sin q_2 - \dot{q}_1^2\cos q_2.
\end{aligned}$$

Thus, the dynamical equations are seen to be linear in the controller parameter vector θ for the above two-link example. (Indeed, it can be shown that a similar linear parameterization is possible for a general n-link manipulator [13].)

For manipulator trajectory control, the "computed torque" control method (e.g. [7]) is commonly used to deal with the nonlinearity of the dynamic equation. The following is an example of a computed torque control law (see Figure 2):

$$\begin{aligned}
u = & \Big(H(\theta,q)[\ddot{\tilde{q}} + 2\lambda\dot{\tilde{q}} + \lambda^2\tilde{q}] \\
& + H(\theta,q)\ddot{q} + C(\theta,q,\dot{q})\dot{q} + g(\theta,q) \Big) && (3) \\
= & \; \tilde{u} + Y(q,\dot{q},\ddot{q})\theta + g(\theta,q) && (4)
\end{aligned}$$

where

$$\begin{aligned}
\tilde{u} &= H(\theta,q)[\ddot{\tilde{q}} + 2\lambda\dot{\tilde{q}} + \lambda^2\tilde{q}] && (5) \\
\tilde{q} &= q_d - q && (6)
\end{aligned}$$

Here q_d is desired trajectory and $\tilde{q}$ is tracking error. The actual joint torque u_a is related to the control signal u by

$$u_a = G_a(s)u + d$$

where $G_a(s)$ represents uncertain actuator dynamics and and d is an uncertain disturbance.

The real variable $\lambda > 0$ is a design parameter which determines the speed at which the tracking error converges to zero. If there are no disturbances (i.e., $d(t) = 0$), no actuator dynamics (i.e., $G_a(s) = 1$), and no other modeling errors, then $u = u_a$ and the application of control law (3) to the idealized robotic manipulator system described by (1) gives

$$H(\theta,q(t))[\ddot{\tilde{q}}(t) + 2\lambda\dot{\tilde{q}}(t) + \lambda^2\tilde{q}(t)] = 0. \tag{7}$$

Because the inertia matrix $H(\theta, q)$ is strictly positive definite for all q, the above equality implies that tracking error $\tilde{q}$ decreases to zero as fast as $e^{-\lambda t}$. If an external disturbance is introduced (i.e., $d(t) \neq 0$), then (7) becomes

$$H(\theta, q(t))[\ddot{\tilde{q}}(t) + 2\lambda\dot{\tilde{q}}(t) + \lambda^2\tilde{q}(t)] = -d(t). \tag{8}$$

The above implies that the robot manipulator tracking error will eventually fall into a region of size proportional to the magnitude of $d(t)$; and, the control law (3) is stabilizing. But this may be guaranteed only for the *idealized* situation in which the parameters are exactly known, the actuators have no dynamics, there is no friction, and the links are completely rigid. Evidently, good performance may still be possible when these idealized assumptions do not hold, at least for some values of the assumed parameter vector θ. Unfalsified control theory provides a rapid and precise means for determining which, if any, of values of θ remain suitable for control of the actual non-idealized physical system, based on a real-time analysis of evolving real-time plant data.

III. Background: Unfalsified Control

The theory of unfalsified control is described in [10]. It is essentially an adaptive control theory that permits learning by a process of elimination. The theory concerns the general feedback control configuration in Figure 3. As always in control theory, the goal is to determine a control law K for the plant P such that the closed-loop system response, say T, satisfies certain given specifications. Unfalsified control theory is concerned with the case in which the plant is either unknown or is only partially known and one wishes to fully utilize information from measurements in selecting the control law K. In the theory of unfalsified control, learning takes place when new information in measurement data enables one to eliminate from consideration one or more candidate controllers.

The three elements that define the unfalsified control problem are (1) plant measurement data, (2) a class of candidate controllers, and (3) a performance specification, say T_{spec}, consisting of a set of admissible 3-tuples of signals (r, y, u). More precisely, we have the following.

Definition [10] : A controller K is said to be **falsified** by measurement information if this information is sufficient to deduce that the performance specification $(r, y, u) \in \mathsf{T}_{spec}\ \forall r \in \mathcal{R}$ would be violated if that controller were in the feedback loop. Otherwise, the control law K is said to be **unfalsified**. ■

To put plant models, data and controller models on an equal footing with performance specifications, these like T_{spec} are regarded as sets of 3-tuples of signals (r, y, u) — that is, they are regarded as relations in $\mathcal{R} \times \mathcal{Y} \times \mathcal{U}$. For example, if $P : \mathcal{U} \to \mathcal{Y}$ and $K : \mathcal{R} \times \mathcal{Y} \to \mathcal{U}$ then

$$\begin{aligned} \mathsf{P} &= \left\{ (r, y, u) \;\middle|\; y = Pu \right\} \\ \mathsf{K} &= \left\{ (r, y, u) \;\middle|\; u = K \begin{bmatrix} r \\ y \end{bmatrix} \right\}. \end{aligned}$$

And, if $J(r,y,u)$ is a given loss-function that we wish to be non-positive, then the performance specification T_{spec} would be simply the set

$$T_{spec} = \left\{ \ (r,y,u) \ \middle| \ J(r,y,u) \leq 0 \ \right\}. \tag{9}$$

On the other hand, experimental information from a plant corresponds to *partial* knowledge of the plant P. Loosely, data may be regarded as providing a sort of an "interpolation constraint" on the graph of P — i.e., a 'point' or set of 'points' through which the infinite-dimensional graph of dynamical operator P must pass.

Typically, the available measurement information will depend on the current time, say τ. For example, if we have complete data on (u,y) from time 0 up to time $\tau > 0$, then the *measurement information* is characterized by the set [10]

$$P_{data} \triangleq \left\{ \ (r,y,u) \in \mathcal{R} \times \mathcal{U} \times \mathcal{Y} \ \middle| \ P_\tau \begin{bmatrix} (u - u_{data}) \\ (y - y_{data}) \end{bmatrix} = 0 \ \right\} \tag{10}$$

where P_τ is the familiar time-truncation operator of input-output stability theory (cf. [11], [17]), viz.,

$$[P_\tau x](t) \triangleq \begin{cases} x(t), & \text{if } 0 \leq t \leq \tau \\ 0, & \text{otherwise.} \end{cases}$$

The main result of unfalsified control theory is the following theorem which gives necessary and sufficient conditions for past open-loop plant data P_{data} to falsify the hypothesis that controller K can satisfy the performance specification T_{spec}.

Unfalsified Control Theorem [10] : A control law K is unfalsified by measurement information P_{data} if, and only if, for each triple $(r_0, y_0, u_0) \in P_{data} \cap K$, there exists at least one pair $(\hat{u}_0, \hat{y}_0)$ such that $(r_0, \hat{y}_0, \hat{u}_0) \in P_{data} \cap K \cap T_{spec}$. ■

The unfalsified control theorem says simply that controller falsification can be tested by computing an intersection of certain sets of signals. It turns out that for the robot manipulator example considered in this paper, this involves a linear programming computation, as the is shown in the next section. A noteworthy feature of the unfalsified control theory is that a controller need not be in the loop to be falsified. Broad classes of controllers can be falsified with open-loop plant data or even data acquired while other controllers were in the loop.

IV. Unfalsified Robot Manipulator Control

We now consider the two-link manipulator trajectory control problem as an application of unfalsified control theory. We shall demonstrate how the *a priori* mathematical knowledge and the *a posteriori* data can be combined in the context of unfalsified control theory to produce a robust adaptive controller. In the unlikely event that the manipulator conforms exactly to the theoretical ideal so that its dynamics are exactly described by (1) with known parameters, and joint torque is exactly as commanded (i.e., joint actuator transfer funtion $G_a(s) = 1$), then the application of the control law (3) will yield satisfactory performance. However a real physical manipulator will have many other factors that cannot be characterized by (1) such as link flexibility and the effects of actuator dynamics, saturation, friction, mechanical backlash and so forth. A mathematical model is never able to describe every detail of a physical system, so there is always a gap between the model and reality. Such a gap may sometimes be fortuitously bridged when the aforementioned factors are "negligible," but unfalsified control theory provides a more robust methodology that ensures that this gap will be overcome whenever possible. We shall show below how the theory directly uses real-time data to quickly and accurately assess the appropriateness of various control laws of the form (3) on a given physical manipulator.

Now, assume the following conditions occur:

C1. Prior Knowledge: Our mathematical knowledge about manipulators in general and prior observation of our particular manipulator's characteristics have caused us to believe that the use of a control law of the general form (3) could result in the performance described by (8), and that (1) and (2) should hold.

C2. Parameter Uncertainty: Parameters such as inertia, location of mass center and so forth cannot be correctly known in advance, due to possible changes of operating conditions or load mass, or due to other unknown causes.

C3. Data: The actuator input commands (u_1, u_2) and the manipulator's output angles (q_1, q_2), velocities $(\dot{q}_1, \dot{q}_2)$, and accelerations $(\ddot{q}_1, \ddot{q}_2)$ are directly measurable.

For this scenario, the unfalsified control method [10] can be applied by taking the reference signal r, measurement signal y and control input signal u to be

$$\begin{aligned} r &= q_d \\ y &= [q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2]^T \\ u &= [u_1, u_2]^T \end{aligned}$$

and, at each time τ, the measurement data is

$$\begin{aligned} u_{data} &= P_\tau u \\ y_{data} &= P_\tau y. \end{aligned}$$

Control Law and Unfalsified Controller Parameter Set

Based on Condition C1 above, the set of admissible control laws and performance specification are selected as follows:

$$\mathsf{K} = \left\{ \; \mathit{K}(\theta) \; \middle| \; \theta \in \mathrm{I\!R}^m \; \right\} \tag{11}$$

$$\text{with } \mathit{K}(\theta) = \left\{ \; (r, y, u) \; \middle| \; u = K_\theta(r, y) \; \right\}$$

$$T_{spec}(\theta) = \left\{ \; (r, y, u) \; \middle| \; J_\theta(r(t), y(t), u(t)) \leq 0 \; \forall t \leq \tau \; \right\} \tag{12}$$

where

$$K_\theta(r, y) \triangleq u(\theta, q, \dot{q}, \ddot{q}) \tag{13}$$

$$J_\theta(r(t), y(t), u(t)) \triangleq \mathrm{abs}(\tilde{u}) - \bar{d} \tag{14}$$

where, in (14), $\bar{d}(t) \geq 0$ is a given time-function and, in (13), $u(\theta, q, \dot{q}, \ddot{q})$, $\tilde{u}(\theta, q, \tilde{q}, \dot{\tilde{q}}, \ddot{\tilde{q}})$ and $\tilde{q}$ are given by (3)-(6). In (12), $J_\theta(r(t), y(t), u(t)) \leq 0$ means each entry of $J_\theta(r(t), y(t), u(t))$, a vector, is less than or equal to zero and $\mathrm{abs}([x_1, \ldots, x_n]^T)$ denotes $[|x_1|, \ldots, |x_n|]^T$. Based on condition C3, the measured data u_{data}, y_{data} consists of past values of commanded joint control torques u and sensor output signals $q, \dot{q}, \ddot{q}$, respectively. In this case, the measurement information set P_{data} is given in terms of the data (u_{data}, y_{data}) by

$$P_{data} \triangleq \left\{ \; (r, y, u) \in \mathcal{U} \times \mathcal{Y} \times \mathcal{U} \; \middle| \; P_\tau u = u_{data}, P_\tau y = y_{data} \; \right\} \tag{15}$$

The notation $\Theta(\tau)$ denotes the set of unfalsified values θ at time τ. Based on the unfalsified control theory, each element $\theta \in \Theta(\tau)$ corresponds to a control law $\mathbf{K}(\theta))$ given by (13).

Notice that the specification $T_{spec}(\theta)$ is *θ-dependent.* It is through this dependence that prior knowledge of the manipulator model influences the adaptation process. In this example, the performance goal is to make $\tilde{u}(\theta, q, \tilde{q}, \dot{\tilde{q}}, \ddot{\tilde{q}})$ small. Not only does this ensure that the control signal u will not be appreciably larger than need be, but it also ensures that the θ-dependent idealized rigid-body manipulator model will be a reasonably good fit to the observed data. The unfalsified control process simultaneously identifies both the plant model and the controller when it identifies $\hat{\theta}$.

The following describes how the unfalsified controller parameter set $\Theta(t)$ can be obtained through set intersections. According to (13), a control law K_θ with parameter vector θ results in the control signal u given by the computed torque control law (3). Hence, the set $P_{data} \cap \mathbf{K}(\theta)$ consists of those points (r, y, u) satisfying for all $t \leq \tau$

$$\begin{aligned} r(t) &= \hat{q}_d(\theta)(t) \\ y(t) &= y_{data}(t) \\ u(t) &= u_{data}(t) \end{aligned}$$

where, for each θ, $\hat{q}_d(\theta)$ is a solution to $u_{data} = K_\theta(\hat{q}_d(\theta), y_{data})$, viz.,

$$\begin{aligned} &\ddot{\hat{q}}_d(\theta) + 2\lambda\dot{\hat{q}}_d(\theta) + \lambda^2 \hat{q}_d(\theta) \\ &\qquad = H(\theta, q)^{-1} \Big(u + H(\theta, q)(2\lambda\dot{q} + \lambda^2 q) \\ &\qquad\qquad - C(\theta, q, \dot{q})\dot{q} - g(\theta, q) \Big) . \end{aligned} \tag{16}$$

Note that, given only the measured plant input-output data and a parameter vector θ, the right hand side of the above can be determined. For this "fictitious" reference signal $\hat{q}_d(\theta)$, the performance of every control law can be examined even if the data was not produced by this control law. According to the unfalsified control theory, the unfalsified controller parameter set $\Theta(\tau)$ at current-time τ can be expressed as a set intersection:

$$\Theta(\tau) = \bigcap_{0 \leq t \leq \tau} \Omega(t) \tag{17}$$

where

$$\Omega(t) \stackrel{\triangle}{=} \left\{ \theta \;\middle|\; \mathrm{abs}(\tilde{u}(q(t), \dot{q}(t), \ddot{q}(t), \theta)) \leq \bar{d}(t), \theta \in \Theta_0 \right\} \tag{18}$$

$$\tilde{u}(q(t), \dot{q}(t), \ddot{q}(t), \theta) \stackrel{\triangle}{=} Y(q(t), \dot{q}(t), \ddot{q}(t))\theta \tag{19}$$

where $Y(q(t), \dot{q}(t), \ddot{q}(t))$ is defined in (2), the vector inequality in (18) holds component-wise, and $\mathrm{abs}([x_1, \ldots, x_n]^T)$ denotes $[|x_1|, \ldots, |x_n|]^T$. In (18), the set $\Theta_0 \subset \mathrm{I\!R}^m$ is an *a priori* estimate the range of controller parameter vectors that are considered to be candidates for achieving the performance specification $T_{spec}(\theta)$.

Parameter Update Law

The goal is to compute at each time τ controller parameter vector $\hat{\theta}(\tau)$ such that $\hat{\theta}(t) \in \Theta(\tau)\ \forall \tau$. The corresponding controller $K_\theta(r, y, u)$ may then be inserted in the control loop. The strategy for choosing $\hat{\theta}(\tau)$ is as follows: The value of $\hat{\theta}(\tau)$ is held constant until such time as it is falsified by the newest data. That is, $\hat{\theta}(\tau)$ remains constant as time τ increases until such time as $\hat{\theta}(\tau^-) \notin \Omega(\tau)$, (Here τ^- denotes the time just an instant prior to time τ.) At the instant τ when this occurs $\hat{\theta}(\tau^-) \notin \Theta(\tau)$, and $\hat{\theta}(\tau)$ must switch to a new value $\hat{\theta}(\tau) \in \Theta(\tau)$.

An important theoretical point to note is that each of the sets $\Omega(\tau)$ defined in (18) is a convex polytope, bounded by the intersection of two pairs of hyperplanes in $\mathrm{I\!R}^4$. And, the intersection of finitely many convex polytopes is always a convex polytope too. The computation of an element of a convex polytope is, it turns out, a linear programming problem for which there are many good computational algorithms. Thus, the computation of an unfalsified $\hat{\theta}(t) \in \Theta(t)$ at a switching time involves using a linear programming algorithm.

We have opted for a $\hat{\theta}$ parameter update law that produces new controller parameter vector $\hat{\theta}$ that is 'optimal' in the sense that it as far as possible from the boundary of the current unfalsified set $\Theta(\tau)$; i.e.,

$$\hat{\theta}(\tau) = \arg \max_{\theta \in \Theta(\tau)} \mathrm{dist}(\theta, \partial\Theta(\tau)) \tag{20}$$

where $\partial\Theta(\tau)$ denotes the boundary of the set $\Theta(\tau)$. Specifically, we compute $\hat{\theta}$ as the solution to the following linear programming problem (whose computational complexity proportional to the number of past measurements being considered [15]):

$$\hat{\theta}(\tau) = \arg \max_{\theta \in \Theta(\tau)} \delta \tag{21}$$

subject to $\forall\ 0 \leq t \leq \tau$

$$\delta \geq 0 \tag{22}$$

$$-Y(q(t), \dot{q}(t), \ddot{q}(t))\theta + \bar{d}(t) + \delta R(t) \geq 0 \tag{23}$$

$$Y(q(t), \dot{q}(t), \ddot{q}(t))\theta - \bar{d}(t) - \delta R(t) \geq 0 \tag{24}$$

where $R(t) \in \mathbb{R}^2$ is given by

$$R(t) \ \stackrel{\Delta}{=} \ \begin{bmatrix} ||Y_1(q(t), \dot{q}(t), \ddot{q}(t))|| \\ ||Y_2(q(t), \dot{q}(t), \ddot{q}(t))|| \end{bmatrix} \tag{25}$$

where $Y_i(\cdot)$, $(i = 1, 2)$ denotes i-th row of the matrix $Y(\cdot)$ defined by (2). Here the maximal δ, say $\hat{\delta}$, is the radius of the largest ball that fits inside the convex polytope $\Theta(\tau)$ and $\hat{\theta}$ is its center. That is, $\hat{\theta}$ is a point in $\Theta(\tau)$ that is as far as possible from $\partial\Theta(\tau)$ and $\hat{\delta}$ the distance of $\hat{\theta}$ from $\partial\Theta(\tau)$.

Besides the batch-type approach linear programming (21), a recursive algorithm for (21) is also possible because the unfalsified controller parameter set $\Theta(\tau)$ is the intersection of degenerate ellipsoids (regions between "parallel" hyper planes), the recursive algorithm of minimal-volume outer approximation by Fogel and Huang [3] can be useful for the calculation of the intersections. Further, convergence of time for the linear program may be much reduced because at each switching time the value $(\hat{\delta}, \hat{\theta}(\tau^-))$ is a feasible point for all constraints $(t < \tau)$ in (22)-(24) and only the constraints (23)-(24) corresponding to $t = \tau$ are violated.

V. Computer Simulation

Simulations are performed to demonstrate the performance of the unfalsified control method. The two-link manipulator given in Example is used in the simulations; furthermore, in order to test the robustness capability of the unfalsified control method, first order transfer functions of different bandwidth are used to simulate the actuator dynamics, viz.,

$$G_a(s) = \frac{1}{\tau s + 1}$$

where $1/(2\pi\tau)$ is the actuator bandwidth in hertz; the values $\tau = 0$, $\tau = 1/(10\pi)$ and $\tau = 1/(40\pi)$ were used in the simulations displayed in the plots. In the simulation, the following robot manipulator parameters are used

$$m_1 = 1,\, l_1 = 1,\, m_e = 2,\, \delta_e = 30^\circ,$$

$$I_1 = .12,\, l_{c1} = .5,\, I_e = 0.25,\, l_{ce} = 0.6$$

so that the exact parameter vector (in the absence of actuator) is $\theta^* \ \stackrel{\Delta}{=} \ [\theta_1, \theta_2, \theta_3, \theta_4]^T \ = [3.34, 0.97, 1.0392, 0.6]^T$. The scenario of the simulation is that the end effector mass m_e changes back and forth between 2 and 20 periodically with period 0.5 *sec*, so does its inertia I_e between 0.25 and 2.5, so that the parameter vector changes between $[3.34, 0.97, 1.0392, 0.6]^T$ and $[30.07, 9.7, 10.3923, 6]$ periodically with period 0.5 *sec* accordingly. The magnitudes of parameter vectors are unknown to the controller. The desired trajectory used is

$$q_{d1}(t) = 30°(1 - \cos 2\pi t), \quad q_{d2}(t) = 45°(1 - \cos 2\pi t) \ .$$

The external torque disturbance acting on the two joints are $\sin 20\pi t$ and $2 \sin 13\pi t$, respectively. At time $t = 0$, the system is initially at rest with joint angles $q_1(0) = q_2(0) = 0.4$ rad.

For comparison, two control methods are examined. One is the unfalsified control method. The other is the adaptive control method by Slotine *et al.* [12]. The Slotine *et al.* controller is

$$u = Y_{slot}\hat{\theta} + K_D\dot{\tilde{q}} + \Lambda\tilde{q} \tag{26}$$

where $\hat{\theta}$ is the estimated parameter vector of θ, K_D is positive definite, and Λ is a positive definite matrix. Similar to (2), Y_{slot} satisfies

$$H(\theta, q)\ddot{q}_r + C(\theta, q, \dot{q})\dot{q}_r + g(\theta, q) = Y_{slot}(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\theta,$$

in which $\dot{q}_r = \dot{q}_d + \Lambda\tilde{q}$. The parameter update law for $\hat{\theta}$ is

$$\dot{\hat{\theta}} = -\Gamma Y_{slot}^T s \tag{27}$$

in which $\Gamma > 0$. The parameters used in the simulation are $K_D = 100I_2$, $\Lambda = 20I_2$, and $\Gamma = \text{diag}([.03, .05, .1, .3])$.

For the unfalsified control method, Θ_0 taken as a solid square box centering at origin with each edge of length 200. To avoid the complexity, the choice of Θ_0 does not take into account the condition $\hat{H}(\theta, q) > \delta I$, $\forall q$, $\forall\theta \in \Theta_0$, as required by the theorems of [14]; simulation results show boundedness of signal can still be obtained without this singularity condition. For simplicity, the computation delay time ν_i is taken as constant for all i with $\nu_i = 10^{-3}$ sec [14]. The parameter λ used in the control law (3) is $\lambda = 20$. The linear programming parameter update law (21) is used, in which constant $\bar{d}(t) = [2, 4]^T$ is the bound on the effect of external disturbance in the performance specifation T_{spec}. In our simulation, the "correct" parameter vector changes periodically every 0.5 seconds starting at $\tau = 0$ and at these time the parameter update law (17) is reset by discarding previous data, resetting the parameter update clock time to $\tau = 0$ and setting $\Theta(0) = \Theta_0$.

In the simulation, both control methods use $\hat{\theta}(0) = [.5, .5, .5, .5]^T$ as initial guess for the parameter estimate, and the results are shown in Figure 5. Simulations in both cases were attempted for each of the three values of actuator time-constant $\tau = 0, 1/(40\pi), 1/(10\pi)$. However, an unstable initially transient with Slotine *et al.*'s method was too severe to permit the simlation to be performed with for actuator bandwidths smaller than 10 hertz (i.e., $\tau \geq 2\pi/10$). Thus five time histories appear in each plot in Figure 5, three for the unfalsified methods and two for the method of Slotine *et al.*.

Plots ((a))-(b)) in Figure 4 each show the five time histories for the joint angle tracking errors $\tilde{q}_1$ and $\tilde{q}_2$, respectively. The smallest amplitude error corresponds to the unfalsified control approach with infinite actuator bandwidth ($\tau = 0$). Next amplitude are the unfalsified-controltracking errors for $\tau = 1/(40\pi)$ and $\tau = 1/(10\pi)$, respectively Increasing amplitude tracking-errors are shown for the Slotine *et al.* controller with $\tau = 0$ and $\tau = 1/(40\pi)$, respectively. Though not shown, other simulations that we performed showed the tracking error with changes in the manipulator end effector mass to be also smaller with our unfalsified controller than with the Slotine *et al.* controller.

Also shown in Figure 4 are the actuator torques required for both the unfalsified control ($\tau = 0,\ 1/(40\pi)$ and $1/(10\pi)$) and the Slotine *et al.* controller ($\tau = 0,\ 1/(40\pi)$). Surprisingly, the three smaller amplitude actuator signals shown correspond to the unfalsified controller, even though these control signals also produce smaller the tracking errors as shown in Figure 5 (a) -(b) . The plots show that the unfalsified controller is able to achieve a more precise and “sure-footed” control over the arm's response without any appreciable increase in control energy.

Figures 5 (a) -(d) show the estimated parameters $\hat{\theta}_i(t)$ $(i = 1, \ldots, 4)$. The two sluggishly smooth traces in each of the four plots are for the Slotine *et al.* controller with $\tau = 0$ and $\tau = 1/(40\pi)$. The crisp “square-wave” response shown in the four plots are for the unfalsified controller. There is no perceptable difference in unfalsified controller response due to variations in the actuator time-constant; the three values of $\tau = 0,\ 1/(40\pi)$ and $1/(10\pi)$ produce time-histories that coincide with the “square-wave” plots shown in the figure. The simulations show that the Slotine *et al.* controller controller cannot accurately track the “correct” parameters. Attempts by us to improve this situation by adjusting Slotine's parameter Γ proved unsuccessful.

Finally, the number of floating point operations (flops) required for each update of $\hat{\theta}(\tau)$ by the unfalsified controller's linear programming routine (solved by using the Matlab *Optimization Toolbox* function `lp.m`) is plotted in Figure 6. The figure shows the times τ at which the controller gain $\hat{\theta}(\tau)$ was falsified and the number of flops (floating point operations) required to solve the linear program (21)–(24) in order to compute a new, as yet unfalsified controller at each of these times. As the Figure shows, between six and eighteen such falsifications occurred in each 0.5 second interval between controller resets. The average computational load during the five second simulation was about 0.8 kflops/msec. Currently, state-of-the-art Pentium-based computers are capable of about 60 kflops/msec. Thus, though the computational burden of our unfalsified controller may seem large, it is well within the capacity of standard microprocessors.

VI. Discussion and Conclusion

Using manipulator trajectory control as an example, we have demonstrated how prior knowledge and the posterior data can be merged within the context of unfalsified control theory to form a synthesis procedure for a nonlinear adaptive controller. Unfalsified control is used to adaptively adjust four parameters of a ‘computed torque’ manipulator arm controller. The parameters represent unknown dynamical coefficients of an ideal frictionless, rigid-link manipulator. For non-ideal manipulators whose dynamics are not necessarily well-approximated by the ideal, robust performance is ensured by application of the unfalsified control method which, using real time data, adjusts the controller parameters so as to ensure that ability of the current controller $K(\hat{\theta}(t))$ to meet the performance goal is unfalsified by the accumulated past measurement data. Convergence is assured provided only that there exists a controller with the specified "computed-torque" structure which meets the performance specification. The "gridding" of controller parameter space which has been typical of previous applications of unfalsified control is avoided by characterizing the evolution of the optimal unfalsified parameter vector $\hat{\theta}(t)$ as the solution to a linear program specified in terms of performance goals and evolving measurement data.

References

[1] T. F. Brozenec and M. G. Safonov. Controller identification. In *Proc. American Control Conf.*, Albuquerque, NM, June 4–6, 1997. IEEE Press, New York.

[2] F. B. Cabral and M. G. Safonov. Robustness oriented controller identification. In *Proc. IFAC Symp. on Robust Control Design, ROCOND '97*, Budpapest, Hungary, June 25-27, 1997. Elsevier Science, Oxford, England.

[3] E. Fogel and Y. F. Huang. On the value of information in system identification–bounded noise case. *Automatica*, 18(2):229–238, 1982.

[4] R. L. Kosut. Adaptive uncertainty modeling: On-line robust control design. In *Proc. American Control Conf.*, pages 245–250, Mineapolis, MN, June 1987. IEEE Press, New York.

[5] R. L. Kosut, M. K. Lau, and S. P. Boyd. Set-membership identification of systems with parametric and nonparametric uncertainty. *IEEE Trans. on Automatic Control*, AC-37(7):929–941, July 1992.

[6] J. M. Krause. Stability margins with real paramenter uncertainty: Test data implications. In *Proc. American Control Conf.*, pages 1441–1445, Pittsburgh, PA, June 21-23, 1989. IEEE Press, New York.

[7] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul. Resolved-acceleration control of mechanical manipulators. *IEEE Trans. on Automatic Control*, AC-25(3):468–474, 1980.

[8] K. Poolla, P. Khargonekar, J. Krause A. Tikku, and K. Nagpal. A time-domain approach to model validation. In *Proc. American Control Conf.*, pages 313–317, Chicago, IL, June 1992. IEEE Press, New York.

[9] M. G. Safonov and T.-C. Tsao. The unfalsified control concept: A direct path from experiment to controller. In B. A. Francis and A. R. Tannenbaum, editors, *Feedback Control, Nonlinear Systems and Complexity*, pages 196–214. Springer-Verlag, Berlin, 1995.

[10] M. G. Safonov and T. C. Tsao. The unfalsified control concept and learning. *IEEE Trans. on Automatic Control*, to appear 1997.

[11] I.W. Sandberg. On the L_2-boundedness of solutions of nonlinear functional equations. *Bell System Technical Journal*, 43(4):1581–1599, July 1964.

[12] J. J. E. Slotine and W. Li. Adaptive manipulator control: A case study. *IEEE Trans. on Automatic Control*, 33(11), 1988.

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

[14] T.-C. Tsao and M. G. Safonov. Adaptive robust manipulator trajectory control — An application of unfalsified control method. In *Proc. Fourth International Conference on Control, Automation, Robotics and Vision*, Singapore, December 3–6, 1996.

[15] A. Vicino and M. Milanese. Optimal inner bounds of feasible parameter set in linear estimation with bounded noise. *IEEE Trans. on Automatic Control*, 36(6):759–763, 1991.

[16] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. on Automatic Control*, AC-36(3):259–294, March 1991.

[17] G. Zames. On the input–output stability of time-varying nonlinear feedback systems — Part I: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Trans. on Automatic Control*, AC-15(2):228–238, April 1966.

List of Figures

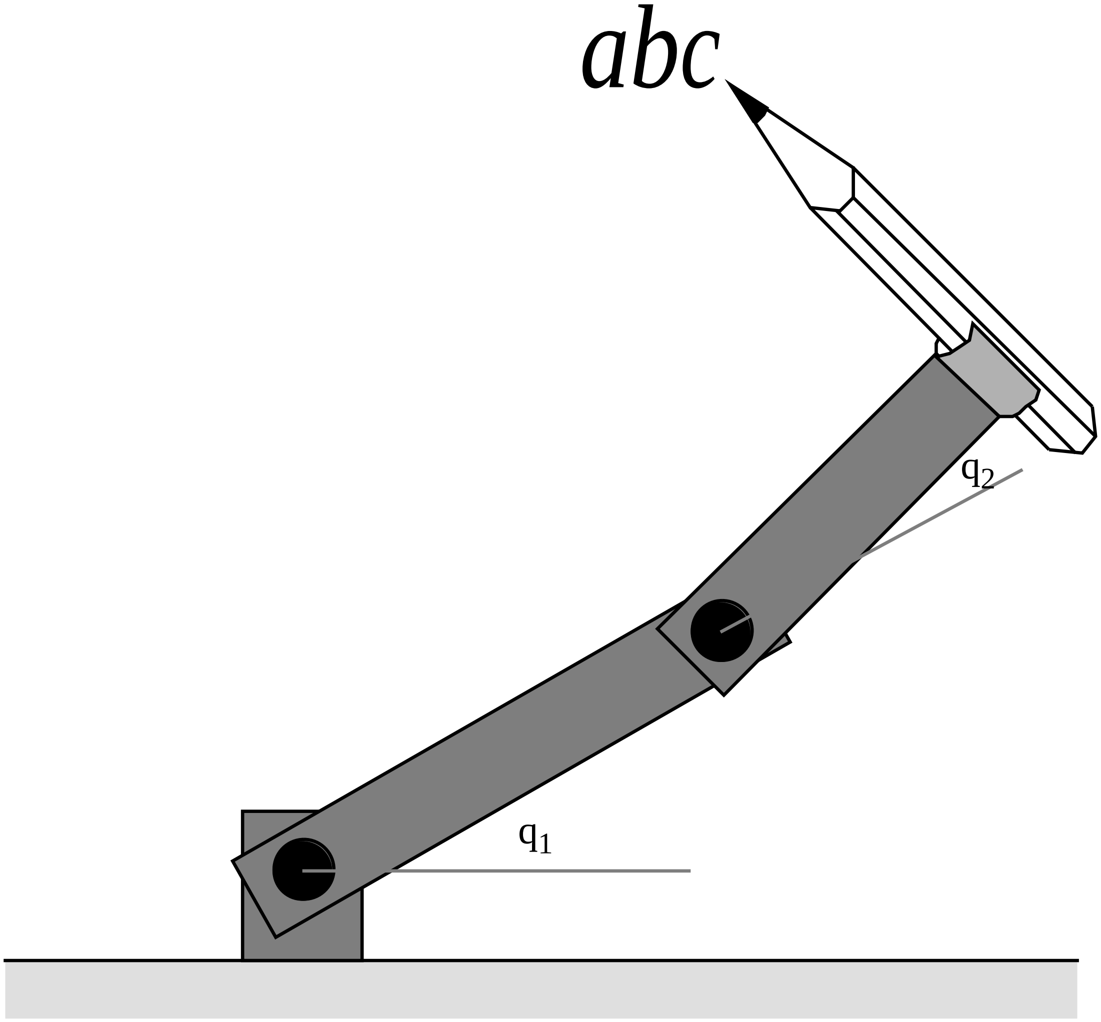


Fig. 1. A two-link robot manipulator with joint angles $q_1(t)$ and $q_2(t)$.

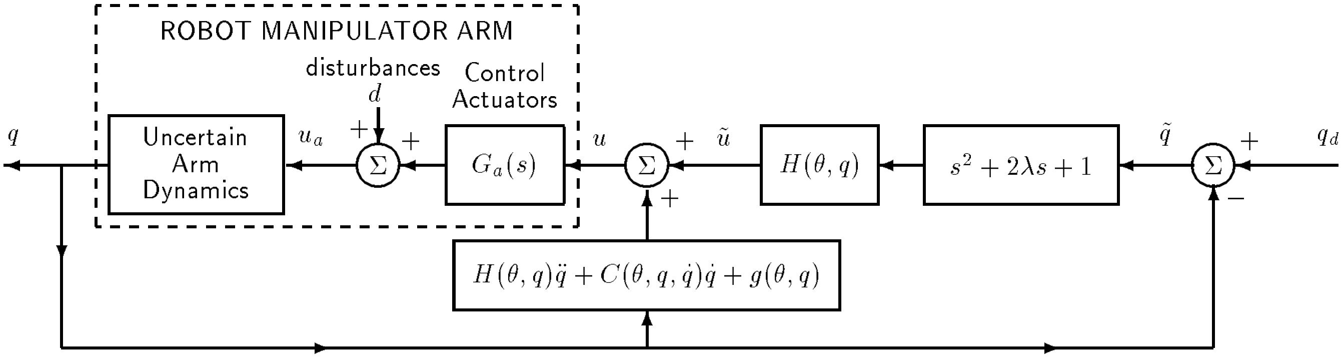


Fig. 2. Computed torque manipulator control configuration.

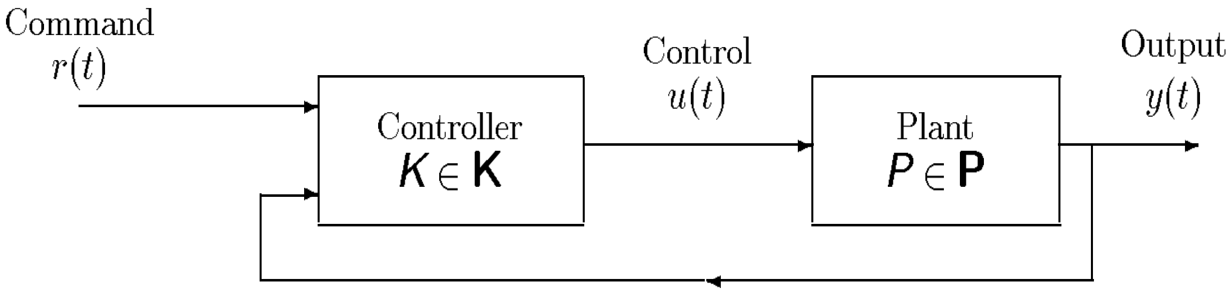
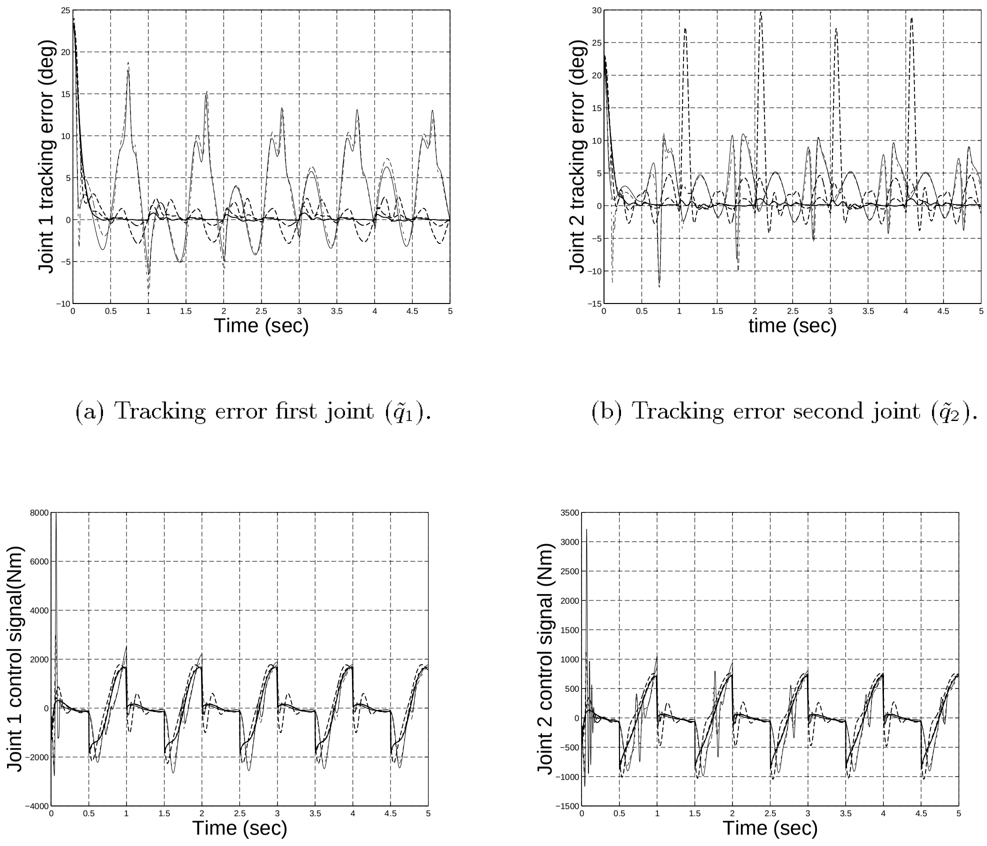


Fig. 3. Feedback control system.



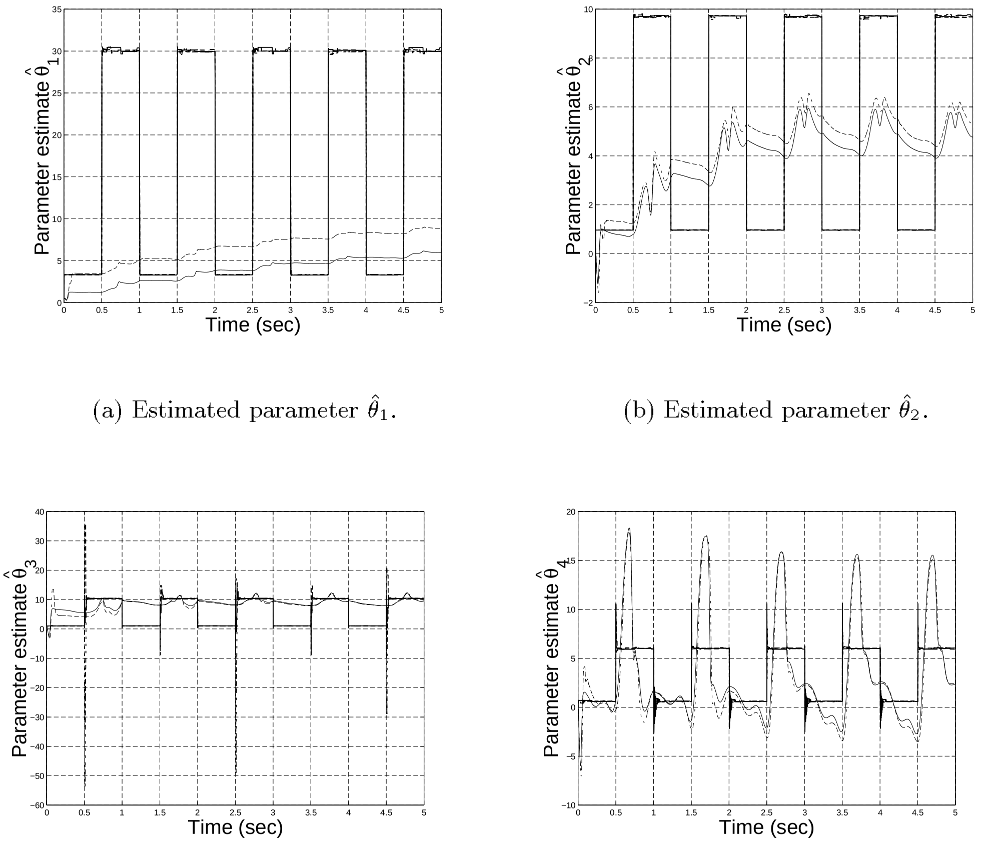
(a) Tracking error first joint ($\tilde{q}_1$).

(b) Tracking error second joint ($\tilde{q}_2$).

(c) First joint command (u_1).

(d) Second joint torque command (u_2).

Fig. 4. Comparison of tracking errors $\tilde{q}$ and control signals u indicates that the unfalsified controller produces a quicker, more precise response, without increased control effort.



(a) Estimated parameter $\hat{\theta}_1$.

(b) Estimated parameter $\hat{\theta}_2$.

(c) Estimated parameter $\hat{\theta}_3$.

(d) Estimated parameter $\hat{\theta}_4$.

Fig. 5. Simulation results for unfalsified controller (thick solid line: ideal actuator, thick dashed line: 20 hz actuator dynamics, thick dotted line: 5 hz actuator dynamics) and Slotine's controller (thin solid line: ideal actuator, thin dashed line: 20 hz actuator dynamics).

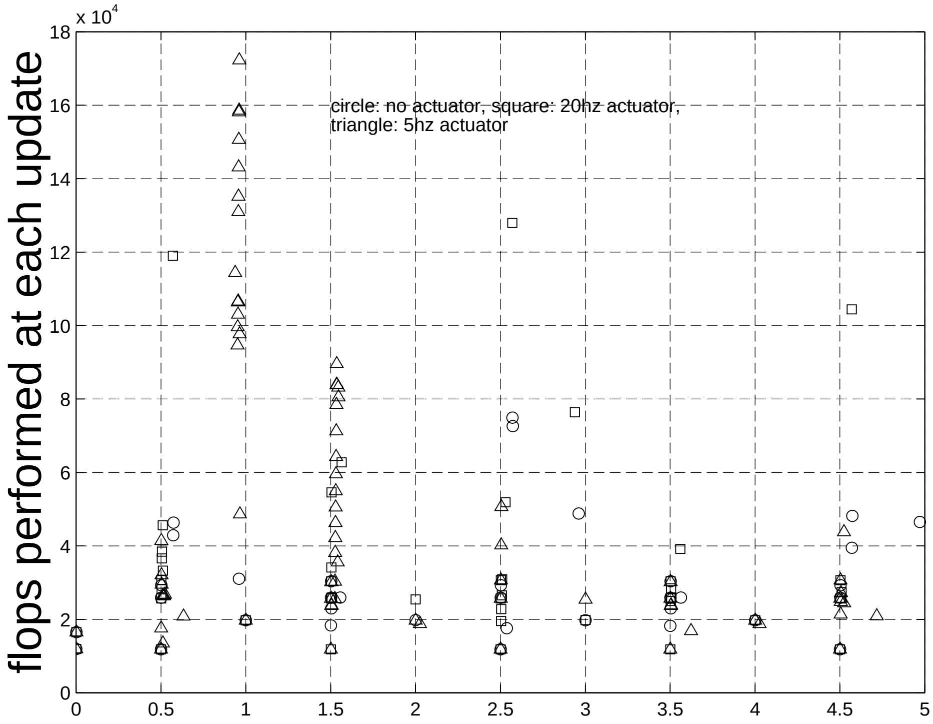


Fig. 6. Floating point operations per millisecond by linear program parameter update law.