The International Journal of Robotics Research
http://ijr.sagepub.com/

On the Adaptive Control of Robot Manipulators

Jean-Jacques E. Slotine and Weiping Li

The International Journal of Robotics Research 1987 6: 49
DOI: 10.1177/027836498700600303

Jean-Jacques E. Slotine
Weiping Li
Nonlinear Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

On the Adaptive Control of Robot Manipulators

Abstract

A new adaptive robot control algorithm is derived, which consists of a PD feedback part and a full dynamics feedforward compensation part, with the unknown manipulator and payload parameters being estimated online. The algorithm is computationally simple, because of an effective exploitation of the structure of manipulator dynamics. In particular, it requires neither feedback of joint accelerations nor inversion of the estimated inertia matrix. The algorithm can also be applied directly in Cartesian space.

1. Introduction

Adaptive control, as a branch of systems theory, is not yet quite mature (see, for instance, Åström 1983; 1984). Yet, the practically motivated drive to make robot manipulators capable of handling large loads in the presence of uncertainty on the mass properties of the load or its exact position in the end-effector, as well as the old "cybernetic" ideal of developing learning capabilities in machines, has spurred much research on adaptive control of robot manipulators (see, e.g., Hsia 1986, for a recent review). The nonlinearity of robot dynamics, however, makes them even more complex to analyze than the linear dynamic systems on which most of the existing adaptive control theory has been traditionally focused.

Several approaches have been considered. Some choose to ignore the dynamic complexity and fit the measured data to a second-order, linear, time-varying model, using for instance a recursive least-squares approach (see, e.g., Koivo 1986). Others do exploit the known structure of the system dynamics (e.g., Khosla and Kanade 1985; Atkeson et al. 1985; Craig et al. 1986), although they generally require estimation of joint accelerations. Another class of algorithms considers the "learning" of specific tasks through the use of feedforward signals (Arimoto et al. 1985; Atkeson et al. 1986), without explicitly updating the manipulator model itself.

In this paper a new adaptive robot control algorithm is derived, which consists of a PD feedback part and a full dynamics feedforward compensation part, with the unknown manipulator and payload parameters being estimated online. The algorithm is computationally simple, because of an effective exploitation of the particular structure of manipulator dynamics. As in Khosla and Kanade (1985) and Atkeson et al. (1985), we use the remark that the dependence of the system dynamics on the unknown parameters can be made linear in terms of a suitably selected set of robot and load parameters. However, contrary to most algorithms in the literature, there is no need to measure the joint accelerations or to invert the estimated inertia matrix.

The layout of the paper is as follows: Section 2 presents our basic adaptive structure in joint space, and in Section 3 we discuss its extension to Cartesian space control. Simulation results are presented in Section 4. Section 5 offers brief concluding remarks.

Extensive experimental results are presented in Slotine and Li (1987).

2. Adaptive Robot Controller in Joint Space

2.1. Dynamic Model of Robot Manipulators

In the absence of friction or other disturbances the dynamics of an n-link rigid manipulator can be written as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \tau, \qquad (1)$$

This research was supported in part by a grant from the Sloan Fund.

The International Journal of Robotics Research,
Vol. 6, No. 3, Fall 1987,

where $\mathbf{q}$ is the $n \times 1$ vector of joint displacements, τ is the $n \times 1$ vector of applied joint torques (or forces), $\mathbf{H}(\mathbf{q})$ is the $n \times n$ symmetric positive definite manipulator inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ is the $n \times 1$ vector of centripetal and Coriolis torques, and $\mathbf{G}(\mathbf{q})$ is the $n \times 1$ vector of gravitational torques.

Two simplifying properties should be noted about this dynamic structure. First, as remarked by several authors (e.g., Arimoto and Miyazaki 1984; Kodistcheck 1984), the matrices $\mathbf{H}$ and $\mathbf{C}$ are not independent. Specifically, given a proper definition of $\mathbf{C}$, the matrix $\dot{\mathbf{H}} - 2\mathbf{C}$ is *skew-symmetric,* as shown in Appendix II. Physically, this property can be easily understood: The derivative of the manipulator's kinetic energy $\dot{\mathbf{q}}^T\mathbf{H}\dot{\mathbf{q}}$ must equal the power input provided by the actuators and the gravitational torques:

$$\frac{1}{2}\frac{d}{dt}[\dot{\mathbf{q}}^T\mathbf{H}\dot{\mathbf{q}}] = \dot{\mathbf{q}}^T[\tau - \mathbf{G}(\mathbf{q})],$$

which implies that at all times

$$\dot{\mathbf{q}}^T(\tfrac{1}{2}\dot{\mathbf{H}} - \mathbf{C})\dot{\mathbf{q}} \equiv 0.$$

Another important property is that the dynamic structure is *linear* in terms of a suitably selected set of robot and load parameters (Khosla and Kanade 1985; Atkeson et al. 1985), as illustrated in Appendix I for a two-link manipulator.

2.2. Controller Design

The controller design problem is as follows: Given the desired trajectory $\mathbf{q}_d(t)$, and with some or all the manipulator parameters being unknown, derive a control law for the actuator torques and an estimation law for the unknown parameters such that the manipulator output $\mathbf{q}(t)$ tracks the desired trajectories after an initial adaptation process.

We derive our adaptive controller in two steps. First, in Section 2.2.1 a simple globally stable adaptive controller is obtained from a Lyapunov stability analysis. The controller strongly exploits the structure of the manipulator dynamics pointed out in the previous section. After the initial transients, however, although the adaptive controller does yield zero velocity errors, it may present nonzero position errors. We solve this problem in Section 2.2.2 by restricting the residual tracking errors to lie on a sliding surface (see Slotine 1985), thus guaranteeing asymptotic convergence of the tracking.

2.2.1. A Globally Stable Adaptive Controller

To derive the control algorithm and adaptation law, we consider the Lyapunov function candidate

$$V(t) = \tfrac{1}{2}(\dot{\tilde{\mathbf{q}}}^T\mathbf{H}(\mathbf{q})\dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{a}}^T\Gamma\tilde{\mathbf{a}} + \tilde{\mathbf{q}}^T\mathbf{K}_p\tilde{\mathbf{q}}), \quad (2)$$

where $\mathbf{a}$ is an m-dimensional vector containing the unknown manipulator and load parameters, and $\hat{\mathbf{a}}$ is its estimate; $\mathbf{K}_P$ and Γ are symmetric positive definite matrices, usually diagonal; $\tilde{\mathbf{q}}(t) = \mathbf{q}(t) - \mathbf{q}_d(t)$ is the tracking error; and $\tilde{\mathbf{a}} = \hat{\mathbf{a}}(t) - \mathbf{a}$ denotes the parameter estimation error vector. Differentiating V yields

$$\begin{aligned}\dot{V}(t) &= \dot{\tilde{\mathbf{q}}}^T\mathbf{H}\ddot{\tilde{\mathbf{q}}} + \tfrac{1}{2}\dot{\tilde{\mathbf{q}}}^T\dot{\mathbf{H}}\dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} + \dot{\tilde{\mathbf{q}}}^T\mathbf{K}_P\tilde{\mathbf{q}} \\ &= \dot{\tilde{\mathbf{q}}}^T(\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \mathbf{H}\ddot{\mathbf{q}}_d) \\ &\quad + \dot{\tilde{\mathbf{q}}}^T[\tfrac{1}{2}(\dot{\mathbf{H}} - 2\mathbf{C}) + \mathbf{C}]\dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}} + \dot{\tilde{\mathbf{q}}}^T\mathbf{K}_P\tilde{\mathbf{q}} \\ &= \dot{\tilde{\mathbf{q}}}^T[\tau - \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_d - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d - \mathbf{G}(\mathbf{q}) + \mathbf{K}_p\tilde{\mathbf{q}}] \\ &\quad + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}},\end{aligned}$$

where we have used the property of skew-symmetry to eliminate the term $\frac{1}{2}\dot{\tilde{\mathbf{q}}}^T(\dot{\mathbf{H}} - 2\mathbf{C})\dot{\tilde{\mathbf{q}}}$. Let us define the control law as

$$\tau = \hat{\mathbf{H}}\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \hat{\mathbf{G}}(\mathbf{q}) - \mathbf{K}_P\tilde{\mathbf{q}} - \mathbf{K}_D\dot{\tilde{\mathbf{q}}}, \quad (3)$$

where the positive definite matrix $\mathbf{K}_D$ may be chosen to be time varying. Then

$$\dot{V}(t) = \dot{\tilde{\mathbf{q}}}^T[\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_d + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \tilde{\mathbf{G}}(\mathbf{q}) - \mathbf{K}_D\dot{\tilde{\mathbf{q}}}] + \tilde{\mathbf{a}}^T\Gamma\dot{\tilde{\mathbf{a}}},$$

where

$$\begin{aligned}\tilde{\mathbf{H}}(\mathbf{q}) &= \hat{\mathbf{H}}(\mathbf{q}) - \mathbf{H}(\mathbf{q}), \\ \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) &= \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \\ \tilde{\mathbf{G}}(\mathbf{q}) &= \hat{\mathbf{G}}(\mathbf{q}) - \mathbf{G}(\mathbf{q}).\end{aligned}$$

Choice (3) cancels the terms associated with the known manipulator parameters, so only the unknown manipulator parameters have to be retained and estimated in $\hat{\mathbf{a}}$. Further, since the matrices $\mathbf{H}$, $\mathbf{C}$, and $\mathbf{G}$ are linear in terms of the manipulator parameters, we can write

$$\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_d + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \tilde{\mathbf{G}}(\mathbf{q}) = \mathbf{Y}\tilde{\mathbf{a}}, \tag{4}$$

where $\mathbf{Y} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$ is an $n \times m$ matrix, and therefore

$$\dot{V}(t) = -\dot{\tilde{\mathbf{q}}}^{\mathrm{T}}\mathbf{K}_D\dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{a}}^{\mathrm{T}}[\Gamma\dot{\tilde{\mathbf{a}}} + \mathbf{Y}^{\mathrm{T}}\dot{\tilde{\mathbf{q}}}].$$

This suggests choosing the adaptation law such that

$$\Gamma\dot{\tilde{\mathbf{a}}} + \mathbf{Y}^{\mathrm{T}}\dot{\tilde{\mathbf{q}}} = \mathbf{0};$$

that is

$$\dot{\hat{\mathbf{a}}} = -\Gamma^{-1}\mathbf{Y}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)\dot{\tilde{\mathbf{q}}}. \tag{5}$$

Note that $\dot{\hat{\mathbf{a}}} = \dot{\tilde{\mathbf{a}}}$, since the unknown parameters $\mathbf{a}$ are constants. The resulting expression of $\dot{V}$ is

$$\dot{V}(t) = -\dot{\tilde{\mathbf{q}}}^{\mathrm{T}}\mathbf{K}_D\dot{\tilde{\mathbf{q}}} \leq 0. \tag{6}$$

Therefore the control law (3) and the adaptation law (5) yield a globally stable adaptive controller.

Expression (6) implies that the steady-state joint velocity error is zero. However, it does not necessarily guarantee that the steady-state position error is also zero. We now modify the previous adaptive scheme in order to solve this potential problem.

2.2.2. Elimination of Steady-State Position Errors

Undesirable steady-state position errors can be eliminated if we restrict them to lie on a sliding surface

$$\dot{\tilde{\mathbf{q}}} + \Lambda\tilde{\mathbf{q}} = \mathbf{0},$$

where Λ is a constant matrix whose eigenvalues are strictly in the right-half complex plane. Formally, we achieve this by replacing the desired trajectory $\mathbf{q}_d(t)$ in the above derivation by the virtual "reference trajectory"

$$\mathbf{q}_r = \mathbf{q}_d - \Lambda\int_0^t \tilde{\mathbf{q}}\,dt. \tag{7a}$$

Accordingly, $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$ are replaced by

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda\tilde{\mathbf{q}}, \tag{7b}$$

$$\ddot{\mathbf{q}}_r = \ddot{\mathbf{q}}_d - \Lambda\dot{\tilde{\mathbf{q}}}. \tag{7c}$$

If we define

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}}_r = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \dot{\tilde{\mathbf{q}}} + \Lambda\tilde{\mathbf{q}},$$

the control law and adaptation law become

$$\tau = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{G}}(\mathbf{q}) - \mathbf{K}_D\,\mathbf{s}, \tag{8}$$

$$\dot{\hat{\mathbf{a}}} = -\Gamma^{-1}\mathbf{Y}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\,\mathbf{s}. \tag{9}$$

Note that the matrix $\mathbf{Y}$ is now a function of $\dot{\mathbf{q}}_r$ and $\ddot{\mathbf{q}}_r$ rather than $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$. We can again demonstrate global convergence of the tracking by now using the Lyapunov function

$$V(t) = \tfrac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{H}\mathbf{s} + \tfrac{1}{2}\tilde{\mathbf{a}}^{\mathrm{T}}\Gamma\,\tilde{\mathbf{a}}, \tag{10}$$

instead of (2), which yields

$$\dot{V}(t) = -\mathbf{s}^{\mathrm{T}}\mathbf{K}_D\mathbf{s} \leq 0, \tag{11}$$

instead of (6). Note that control law (8) does not contain a term in $\mathbf{K}_P$, since the position error $\tilde{\mathbf{q}}$ is already included in $\dot{\mathbf{q}}_r$. Expression (11) shows that the output errors converge to the sliding surface

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \Lambda\tilde{\mathbf{q}} = \mathbf{0}. \tag{12}$$

This in turn implies that $\tilde{\mathbf{q}} \rightarrow 0$ as $t \rightarrow \infty$. Thus, the adaptive controller defined by (8) and (9) is globally asymptotically stable and guarantees zero steady-state error for joint positions.

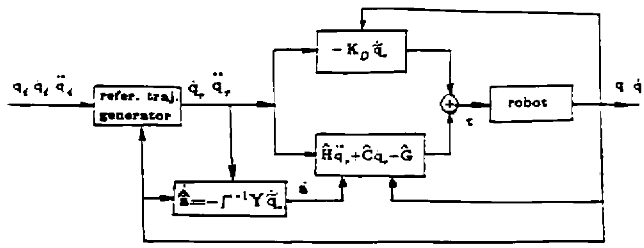
The previous proof of tracking convergence may seem somewhat unorthodox to readers not familiar with sliding control theory. Let us detail the basic features. First, the vector $\mathbf{s}$ conveys information about boundedness and convergence of $\mathbf{q}$ and $\dot{\mathbf{q}}$, since *the definition of* $\mathbf{s}$ *can also be viewed as a stable first-order differential equation in* $\tilde{\mathbf{q}}$, with $\mathbf{s}$ as an input. Thus, for bounded initial conditions, boundedness of $\mathbf{s}$ implies boundedness of $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ and, therefore, of $\mathbf{q}$ and $\dot{\mathbf{q}}$;

similarly, one can easily show that if $\mathbf{s}$ tends to $\mathbf{0}$ as $t \to \infty$, so do $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$. Second, the function V is actually a quasi-Lyapunov function, in our case simply a positive continuous function of time. Let us now detail the proof itself. Since $\dot{V}$ is negative or zero and V is lower bounded (by zero), V tends to a constant as $t \to \infty$ and therefore remains bounded for $t \in [0, \infty]$. Given the definition (10) of V, this in turn implies, since $\mathbf{H}$ is uniformly positive definite (i.e., $\mathbf{H} \geqslant h\mathbf{I}$ for some strictly positive h), that $\mathbf{s}$ is bounded and, therefore, that $\mathbf{q}$ and $\dot{\mathbf{q}}$ are bounded; it also implies that $\tilde{\mathbf{a}}$ is bounded and, therefore, that $\hat{\mathbf{a}}$ is bounded. From the system dynamics this then makes $\dot{\mathbf{s}}$ bounded, and thus $\mathbf{s}$ is *uniformly continuous* on $t \in [0, \infty]$. Assuming that the (perhaps time-varying) matrix $\mathbf{K}_D$ is chosen to be uniformly continuous (as is typically the case, for instance, with $\mathbf{K}_D$ constant, or with $\mathbf{K}_D = \lambda\hat{\mathbf{H}}$), $\dot{V}$ is then uniformly continuous on $t \in [0, \infty]$; therefore, since V is bounded on that time interval and $\dot{V}$ is of constant sign ($\dot{V} \leqslant 0$), *$\dot{V}$ tends to zero as $t \to \infty$*. Assuming that $\mathbf{K}_D$ is uniformly positive definite (as is again the case if $\mathbf{K}_D$ is chosen to be constant, or if $\mathbf{K}_D = \lambda\hat{\mathbf{H}}$), this implies from (11) that $\mathbf{s} \to \mathbf{0}$ as $t \to \infty$, and therefore that $\dot{\tilde{\mathbf{q}}} \to \mathbf{0}$ as $t \to \infty$.

The structure of the adaptive controller given by (8) and (9) is sketched in Fig. 1. The controller consists of two parts. The first part consists of three feedforward terms corresponding to inertial, centripetal and Coriolis, and gravitational torques. The second part contains two terms representing PD feedback. The required inputs to the controller are the desired joint position $\mathbf{q}_d$, velocity $\dot{\mathbf{q}}_d$, and acceleration $\ddot{\mathbf{q}}_d$ from the trajectory planner, and the required measurements are the joint position $\mathbf{q}$ and velocity $\dot{\mathbf{q}}$. Contrary to several algorithms in the literature (e.g., Craig et al. 1986), there is no need for measuring the joint accelerations $\ddot{\mathbf{q}}$ or for inverting the estimated inertia matrix. Note that if measurements of joint accelerations were indeed explicitly available online, one could easily show (Slotine 1986) that the effect of parametric uncertainty on performance could in principle be made arbitrarily small by simply increasing the value of the acceleration gain, without using adaptation; however, this procedure would be extremely sensitive to imprecision on the joint acceleration measurement, which then essentially would enter as a pure disturbance added to $\ddot{\mathbf{q}}$.

Note from Fig. 1 that the integral term $\int_0^t \tilde{\mathbf{q}}\, dt$ of (7)

Fig. 1. Structure of the joint space adaptive controller.



need not be actually computed, since only $\dot{\mathbf{q}}_r$ and $\ddot{\mathbf{q}}_r$ (not $\mathbf{q}_r$) are *explicitly* used in the control law. Therefore, the formal definition of $\mathbf{q}_r$ is, in effect, equivalent to adding a feedback loop.

2.3. Discussion

In this section we discuss implementation aspects, computational efficiency, and strategies that combine adaptation on certain parameters with robustness to uncertainty on others and to disturbances.

2.3.1. *Implementation Aspects*

Since the load is usually fixed with respect to the last link, it can be regarded as part of that link. In practice, the parameters of the robot itself can be measured or estimated beforehand (Khosla and Kanade 1985; Atkeson et al. 1985), so only the parameters of the load are unknown. Models of Coulomb and viscous friction may also be included in (1), and the corresponding coefficients can be identified similarly.

Although convergence of the trajectory tracking is guaranteed in the previous derivation, the parameter estimates themselves do not necessarily converge to their exact values. Intuitively, to guarantee parameter convergence, the desired trajectory must be "sufficiently rich" so that only the true set of parameters can yield exact tracking. A formalization of this concept in the context of robot control and the generation of trajectories that speed up parameter convergence constitute interesting research topics in themselves (Morgan and Narendra 1977; Craig et al. 1986).

We stop updating a given parameter when it reaches its known bounds, and we resume updating as soon as the corresponding derivative changes signs. This intuitively motivated procedure can easily be shown to preserve convergence of the tracking.

2.3.2. Computational Efficiency

In the practical implementation of the previous adaptive controller, the matrices $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{G}}$ may be updated at a low rate, whereas a high update rate is used for $\dot{\mathbf{q}}_r$, $\ddot{\mathbf{q}}_r$, and $\mathbf{s}$, since typically the error terms vary much faster than the dynamic coefficient matrices (see, e.g., Khatib 1986). Further, the matrix $\mathbf{Y}$, whose calculation is naturally coupled to the dynamics computation, can also be updated at the slow rate, since the choice of the adaptation gain matrix Γ is generally such that the adaptation process is slower than the control bandwidth.

Because of the presence of $\dot{\mathbf{q}}_r$ in the second term of control law (8), however, the controller cannot be implemented directly with fast recursive formulations, such as the Newton-Euler method, and, therefore, requires explicit computations of $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{G}}$. The same is true of adaptation law (9). We now introduce a recursive Newton-Euler method as an alternative way of implementing the control and adaptation laws. This Newton-Euler formulation can be seen as an approximation of the previous development, for which new stability conditions are derived.

Assume that the second term $\hat{\mathbf{C}}\dot{\mathbf{q}}_r$ in (8) is approximated by $\hat{\mathbf{C}}\dot{\mathbf{q}}$. Then we can compute the first three terms in (8) by a recursive Newton-Euler method, based on the parameters obtained from the adaptation law. The resulting control torque is

$$\tau = \hat{\mathbf{H}}(\mathbf{q})(\ddot{\mathbf{q}}_d + \Lambda\dot{\tilde{\mathbf{q}}}) + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{G}}(\mathbf{q}) - \mathbf{K}_D\mathbf{s}, \quad (13)$$

which is computed through a number of operations proportional to the number of links. Accordingly, the same approximation is made in the calculation of the matrix $\mathbf{Y}$, namely,

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\hat{\mathbf{a}} = \hat{\mathbf{H}}\ddot{\mathbf{q}} + \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{G}}. \quad (14)$$

Let us examine the effects of these approximations. We have

$$\dot{\hat{\mathbf{a}}} = -\Gamma^{-1}\mathbf{Y}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\mathbf{s}, \quad (15)$$

with now

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\mathbf{a}} = \tilde{\mathbf{H}}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \tilde{\mathbf{G}}. \quad (16)$$

From (10),

$$\dot{V}(t) = \mathbf{s}^{\mathrm{T}}[\tau - \mathbf{H}\ddot{\mathbf{q}}_r - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{G}] + \tilde{\mathbf{a}}^{\mathrm{T}}\Gamma\dot{\hat{\mathbf{a}}}.$$

Thus from (13), (15), and (16), we obtain

$$\begin{aligned}
\dot{V}(t) &= \mathbf{s}^{\mathrm{T}}[\hat{\mathbf{H}}\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{G}} - \mathbf{K}_D\mathbf{s} - \mathbf{H}\ddot{\mathbf{q}}_r - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{G}] \\
&\quad - \tilde{\mathbf{a}}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\mathbf{s} \\
&= \mathbf{s}^{\mathrm{T}}[\tilde{\mathbf{H}}\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}\dot{\mathbf{q}}_r + \tilde{\mathbf{G}} - \mathbf{K}_D\mathbf{s} + \mathbf{C}\dot{\mathbf{q}} - \mathbf{C}\dot{\mathbf{q}}_r] \\
&\quad - \mathbf{s}^{\mathrm{T}}[\tilde{\mathbf{H}}\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}\mathbf{q} + \tilde{\mathbf{G}}] \\
&= -\mathbf{s}^{\mathrm{T}}[\mathbf{K}_D - \mathbf{C}]\mathbf{s} = -\mathbf{s}^{\mathrm{T}}[\mathbf{K}_D - \tfrac{1}{2}\dot{\mathbf{H}}]\mathbf{s},
\end{aligned}$$

using the skew-symmetry of the matrix $(\dot{\mathbf{H}} - 2\mathbf{C})$. Therefore, the stability of this recursive formulation of the adaptive controller is guaranteed as long as $\mathbf{K}_D$ is chosen large enough (perhaps time varying) to satisfy $\mathbf{K}_D > \frac{1}{2}\dot{\mathbf{H}}$.

2.3.3. Combining Adaptation with Robustness

In practice, we may simplify the algorithm by not explicitly estimating all unknown parameters. Some parameters may have relatively minor importance in the dynamics, in which case we may choose to make the controller robust to the uncertainty on these parameters rather than explicitly estimating them online. Similarly, some geometric parameters may already be known with reasonable precision or may have been estimated through sorting devices or visual information. Further, the controller must be robust to residual time-varying disturbances, such as stiction or torque ripple.

We categorize the unknown parameters $\mathbf{a}$ into two groups: group $\mathbf{a}_E$ contains the parameters estimated online; group $\mathbf{a}_R$ contains the parameters not estimated online. A sliding control term is then incorporated into the torque input (8) to account for the effects of uncertainties on the parameters in $\mathbf{a}_R$ and of disturbances.

Assume, without loss of generality, that only the first α unknown parameters are to be actually estimated:

$$\mathbf{a} = [\mathbf{a}_E^{\mathrm{T}} \quad \mathbf{a}_R^{\mathrm{T}}]^{\mathrm{T}},$$

$$\text{with } \mathbf{a}_E = \{a_j\}^{\mathrm{T}}_{j=1,\dots,\alpha}, \quad \mathbf{a}_R = \{a_j\}^{\mathrm{T}}_{j=\alpha+1,\dots,m},$$

and let, correspondingly, $\mathbf{Y} = [\mathbf{Y}_E \quad \mathbf{Y}_R]$. Assume that the uncertainties on $\mathbf{a}_R$, as well as the disturbance torques d_i reflected to the manipulator joints, are bounded:

$$|\tilde{a}_j| \leq A_j, \quad j = \alpha + 1, \ldots, m; \qquad |d_i(t)| \leq D_i(t), \quad i = 1, \ldots, n.$$

Add a sliding control term to torque input (8):

$$\tau = \hat{\mathbf{H}}\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\dot{\mathbf{q}}_r + \hat{\mathbf{G}} - \mathbf{K}_D\mathbf{s} - \mathbf{k} \operatorname{sgn}(\mathbf{s}), \qquad (17)$$

where the notation $\mathbf{k} \operatorname{sgn}(\mathbf{s})$ stands for the $n \times 1$ vector of components $k_i \operatorname{sgn}(s_i)$, with the k_i yet to be specified. With $\mathbf{a}_E$ and Γ_E in place of $\mathbf{a}$ and Γ in the Lyapunov function (10), we obtain

$$\dot{V}(t) = -\mathbf{s}^T[\mathbf{K}_D\mathbf{s} + \mathbf{Y}_R\tilde{\mathbf{a}}_R - \mathbf{k} \operatorname{sgn}(\mathbf{s})] + \tilde{\mathbf{a}}_E^T[\Gamma_E\dot{\hat{\mathbf{a}}}_E + \mathbf{Y}_E^T\mathbf{s}].$$

Since

$$\mathbf{Y}_R\tilde{\mathbf{a}}_R - \mathbf{k} \operatorname{sgn}(\mathbf{s}) = \left\{ \sum_{j=\alpha+1}^{m} Y_{ij}\tilde{a}_j + d_i(t) - k_i \operatorname{sgn}(s_i) \right\}^T_{i=1,\ldots,n},$$

we let

$$k_i = \sum_{j=\alpha+1}^{m} |Y_{ij}|A_j + D_i + \eta_i, \qquad i = 1, \ldots, n,$$

$$\dot{\hat{\mathbf{a}}}_E = -\Gamma_E^{-1}\mathbf{Y}_E^T\mathbf{s},$$

where the η_i are positive constants. This yields

$$\dot{V}(t) \leq -\sum_{i=1}^{n} \eta_i|s_i| - \mathbf{s}^T\mathbf{K}_D\mathbf{s} \leq 0. \qquad (18)$$

The system trajectories are thus guaranteed to reach sliding surface $\mathbf{s} = \mathbf{0}$, and therefore convergence of the tracking is achieved.

Further, to avoid undesirable control chattering, we can use saturation functions sat (s_i/ϕ_i) in place of the switching function sgn (s_i), with the ϕ_i representing the thicknesses of the corresponding "boundary layers." Similarly to Slotine (1984), $\mathbf{s}$ is then guaranteed to converge to the boundary layers, with corresponding small tracking errors; further, the ϕ_i can be modulated based on bandwidth considerations. Similarly to Slotine and Coetsee (1986), parameter adaptation must then be stopped when the system trajectories are inside the boundary layers; indeed, by definition, disturbances and errors on $\mathbf{a}_R$ can drive the trajectories anywhere in the boundary layers without this providing any information about the estimation error on $\mathbf{a}_E$. This procedure also has the advantage of avoiding long-term drift of the estimated parameters.

Note from (18) that $\mathbf{K}_D\mathbf{s}$ can be eliminated from control input (17), since the sliding control action makes it unnecessary; however, this term must be kept in a Newton–Euler implementation of the algorithm to compensate for the approximation of $\hat{\mathbf{C}}\dot{\mathbf{q}}_r$ by $\hat{\mathbf{C}}\dot{\mathbf{q}}$, as discussed earlier. It may also be retained in order to accelerate convergence. Note that fixed-parameter sliding control is obtained if none of the unknown parameters is explicitly estimated ($\alpha = m$).

3. Extension to Cartesian Space Control

In this section we extend the previous joint space adaptive controllers to task space. To this effect, for a nonredundant manipulator, we simply replace the reference trajectories in (7b) and (7c) by

$$\dot{\mathbf{q}}_r = \mathbf{J}^{-1}[\dot{\mathbf{x}}_d + \Lambda(\mathbf{x}_d - \mathbf{x})], \qquad (19a)$$

and, accordingly,

$$\ddot{\mathbf{q}}_r = \mathbf{J}^{-1}\{[\ddot{\mathbf{x}}_d + \Lambda(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})] - \dot{\mathbf{J}}\dot{\mathbf{q}}_r\}, \qquad (19b)$$

so that

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}}_r = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \mathbf{J}^{-1}[\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d + \Lambda\tilde{\mathbf{x}}].$$

The same control and adaptation laws (8) and (9) are then used, again with (10) as the Lyapunov function. Following the same derivation as before, we obtain

$$\begin{aligned}\dot{V}(t) &= -\mathbf{s}^T\mathbf{K}_D\mathbf{s} \\ &= -[\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d + \Lambda\tilde{\mathbf{x}}]^T\mathbf{J}^{-T}\mathbf{K}_D\mathbf{J}^{-1}[\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d + \Lambda\tilde{\mathbf{x}}] \\ &\leq 0,\end{aligned}$$

Fig. 2. Two-link manipulator carrying a large unknown load.

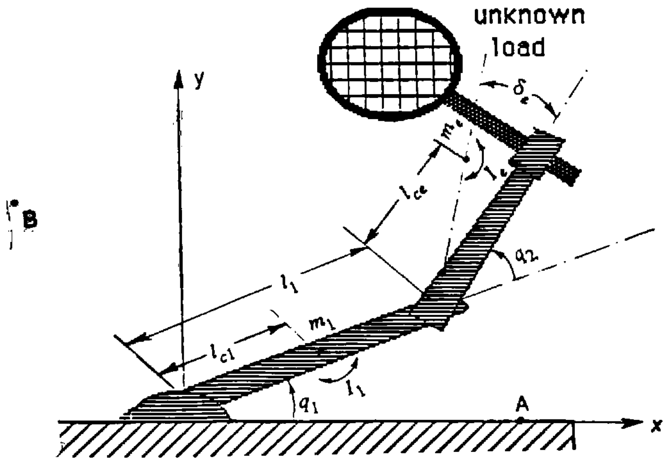
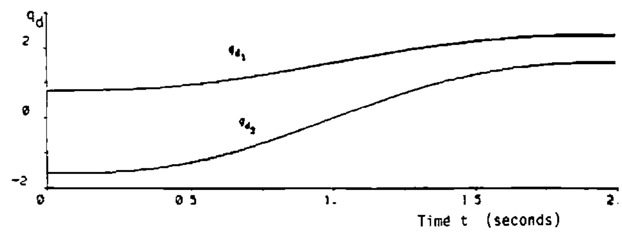


Fig. 3. Desired joint trajectories for Examples 1 and 2.



which implies convergence to

$$\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d + \Lambda\tilde{\mathbf{x}} = \mathbf{0}. \tag{20}$$

Using the kinematic relation $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, we recognize expression (20) as the equation of the sliding surface $\dot{\tilde{\mathbf{x}}} + \Lambda\tilde{\mathbf{x}} = \mathbf{0}$, which in turn guarantees that $\tilde{\mathbf{x}} \rightarrow \mathbf{0}$ as $t \rightarrow \infty$. Therefore, the previous adaptive controller is globally stable and guarantees zero steady-state, Cartesian space, position error.

Note from (19a) and (19b) that only the desired trajectories in Cartesian space $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, and $\ddot{\mathbf{x}}_d$ have to be given (i.e., explicit inverse kinematics is not necessary). The quantities to be measured are joint positions $\mathbf{q}$ and joint velocities $\dot{\mathbf{q}}$. End-effector position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ can be obtained from the direct kinematics, and therefore do not need to be explicitly measured. Also, note that the inverse Jacobian $\mathbf{J}^{-1}$ appears in (19a) and (19b), and therefore singularity points should be avoided (see Khatib 1986 for a relaxation of this condition).

4. Simulation Results

We present computer simulations using the two-link planar manipulator considered in Appendix I, carrying a large load of unknown mass properties (Fig. 2). The two links are identical uniform beams, with actuators mounted at the joints. In the simulations the unknown load actually has the same geometry as the links but is twice as heavy. For simplicity, the parameters of the robot itself are assumed to be exactly known. The parameters to be adapted are α, β, ϵ, and η, whose true values are $\alpha = 6.7$, $\beta = 3.4$, $\epsilon = 3.0$, and $\eta = 0$. The initial estimates of the load mass properties assume that the load is identical to the second link. The corresponding initial parameter estimates are $\hat{\alpha} = 4.1$; $\hat{\beta} = 1.9$, $\hat{\epsilon} = 1.7$, and $\hat{\eta} = 0$. In the simulation plots the estimates of the first three parameters are normalized by the true values, and $\hat{\eta}$ is normalized by 3 (the true value of ϵ), since η is itself zero.

Example 1: Comparison with conventional controllers

The task is to move the load from position A to position C, as indicated in Fig. 2. Three controllers are used: (1) PD controller, (2) PD + full dynamics feedforward compensation, and (3) adaptive controller given by (3) and (5). The desired joint trajectories are chosen to be fifth-order polynomials and are shown in Fig. 3. The matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ are chosen to be identical for all three controllers, with $\mathbf{K}_P = 800I$ and $\mathbf{K}_D = 160I$. The results are plotted in Fig. 4 for controller a, Fig. 5 for controller b, and Fig. 6 for controller c. The maximum joint position errors are about 7.5° for controller a, 3° for controller b, and only about 0.5° for the adaptive controller. The maximum actuator torques are smaller for the adaptive controller than for controllers a and b. The parameter estimates do not converge to their exact values, since the desired trajectory is not persistently exciting. Also, as anticipated in Section 2.2.1, the joint position errors do not exactly converge to zero, a problem that we now remedy using the development of Section 2.2.2.

Fig. 4. *PD controller in Example 1.*

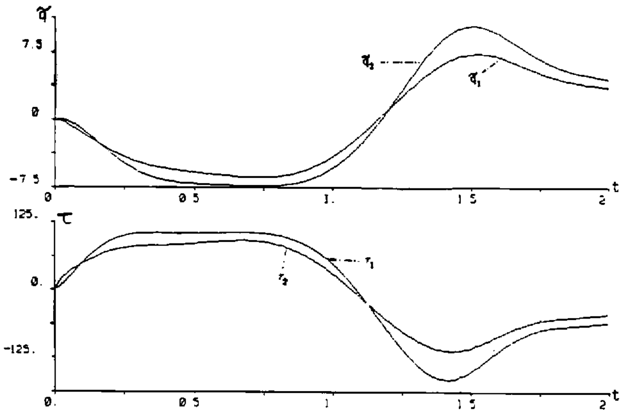
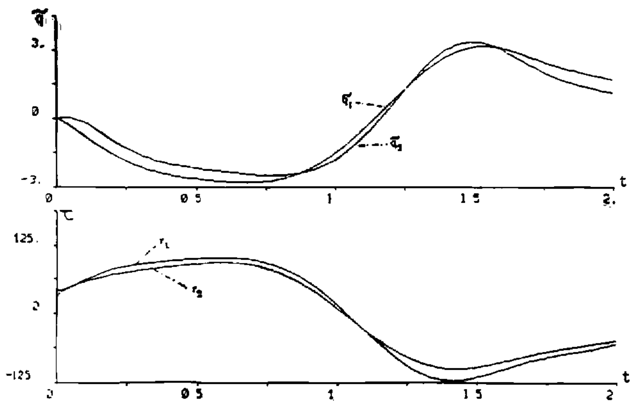


Fig. 5. *PD + full dynamics feedforward controller in Example 1.*



Example 2: Elimination of steady-state position error

The adaptive controller given by (7) and (8) is simulated with the same parameters as in Example 1, and $\Lambda = 30I$. The joint position errors now converge to zero (Fig. 7). We also note that the maximum joint position errors have been reduced to only 0.08° without significant increase in actuator torques.

A smaller value of Λ is also simulated. With $\Lambda = 5I$, the product of $\mathbf{K}_D$ and Λ is the same as $\mathbf{K}_P$ of controller c in Example 1; however, the resulting maximum position errors are only 0.12°, and convergence to zero is observed.

Example 3: Parameter convergence

In this example the desired trajectory is chosen to be

$$\theta_{d1} = a_1 \sin(t) + a_2 \cos(t) + a_3 \sin(3t) + a_4 \cos(3t) + a_5 \sin(5t) + a_6 \cos(5t),$$
$$\theta_{d2} = b_1 \sin(2t) + b_2 \cos(2t) + b_3 \sin(4t) + b_4 \cos(4t) + b_5 \sin(6t) + b_6 \cos(6t).$$

The coefficients a_i and b_i are chosen to make the desired trajectory satisfy the initial and final conditions on position, velocity, and acceleration. The same adaptive controller as in Example 2 is used. Although it may not be necessary to have six frequency components for the desired trajectory to be persistently exciting, this example demonstrates that sufficiently rich desired trajectories do yield convergence of the parameter estimation (Fig. 8).

Example 4: Cartesian space adaptive controller

The same task as that in previous examples is performed by the adaptive Cartesian space controller of Section 3. The desired path is now a straight line from A to B in Fig. 2. A fifth-order polynomial is constructed for the desired displacement along the path, which has zero velocities and accelerations at the start and the end of the path. The feedback gains and all other parameters are the same as before. The performance of this controller (Fig. 9) is similar to that of the joint space adaptive controller. The steady-state Cartesian position errors are zero, and the maximum Cartesian path errors in the x- and y-directions are about 8×10^{-4} m.

Extensive experimental results (Slotine and Li 1987) confirm these simulations.

5. Concluding Remarks

It is of interest to further investigate specific choices of the adaptation gain matrix Γ that yield optimal convergence rates while still avoiding the excitation of high-frequency unmodeled dynamics (such as structural resonant modes, actuator dynamics, or sampling effects). This may involve employing a time-varying Γ, based, e.g., on a Gauss–Newton algorithm. Although

Fig. 6. Adaptive controller (3), (5).

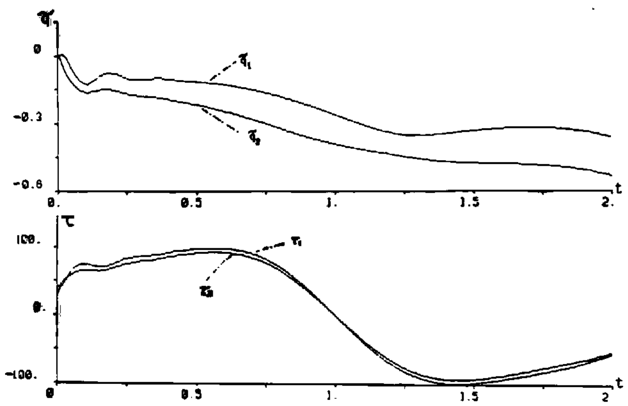
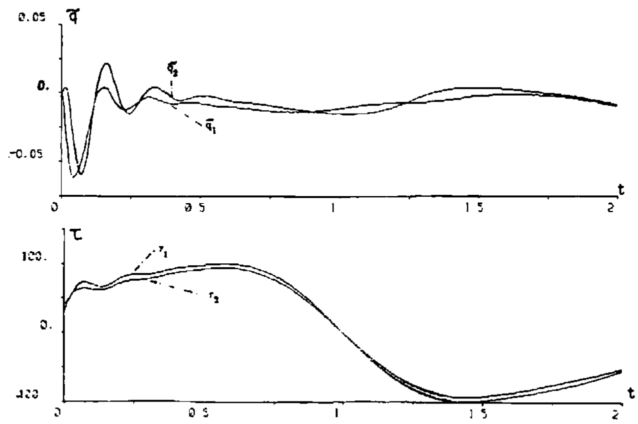


Fig. 7. Adaptive controller with steady-state position error eliminated.



in principle an approach similar to that of Slotine and Coetsee (1986) could be used to this effect, we believe that in this instance it may be more effective to try again to take full advantage of the specific structure of the manipulator dynamics. This will be the object of a separate study.

Further, in the more general context of control system design for physical nonlinear systems, we believe that the approach that consists of modifying, through feedback, the system's natural energy function rather than its explicit expanded dynamics is worthy of further investigation in its own right.

Appendix I: Two-Link Manipulator with Large Unknown Load

A two-link planar manipulator carrying an unknown payload is shown in Fig. 2. The second link, with the payload attached, can be regarded as an augmented link with four unknown parameters, namely, mass m_a, moment of inertia I_e, the distance l_{ce} of its mass center to the second joint, and the angle δ_e relative to the original second link. The dynamics of the manipulator with payload can then be written as

$$\begin{bmatrix} \alpha + 2\epsilon\cos(q_2) + 2\eta\sin(q_2) & \beta + \epsilon\cos(q_2) + \eta\sin(q_2) \\ \beta + \epsilon\cos(q_2) + \eta\sin(q_2) & \beta \end{bmatrix}\begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \end{bmatrix} + \begin{bmatrix} \epsilon Y_1 + \eta Y_2 + (\alpha - \beta + e_1)e_2\cos(q_1) \\ \epsilon Y_3 + \eta Y_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

where

$$\begin{aligned} Y_1 &= -2\sin(q_2)\dot{q}_1\dot{q}_2 - \sin(q_2)\dot{q}_2^2 + e_2\cos(q_1 + q_2), \\ Y_2 &= 2\cos(q_2)\dot{q}_1\dot{q}_2 + \cos(q_2)\dot{q}_2^2 + e_2\sin(q_1 + q_2), \\ Y_3 &= \sin(q_2)\dot{q}_1^2 + e_2\cos(q_1 + q_2), \\ Y_4 &= -\cos(q_2)\dot{q}_1^2 + e_2\sin(q_1 + q_2), \\ e_1 &= m_1 l_1 l_{c1} - I_1 - m_1 l_{m1}^2, \\ e_2 &= g/l_1, \end{aligned}$$

where g is the acceleration of gravity, and the four unknown parameters α, β, ϵ, and η are functions of the unknown physical parameters:

$$\begin{aligned} \alpha &= I_1 + m_1 l_{c1}^2 + I_e + m_e l_{ce}^2 + m_e l_1^2, \\ \beta &= I_e + m_e l_{ce}^2, \\ \epsilon &= m_e l_1 l_{ce}\cos(\delta_e), \\ \eta &= m_e l_1 l_{ce}\sin(\delta_e). \end{aligned}$$

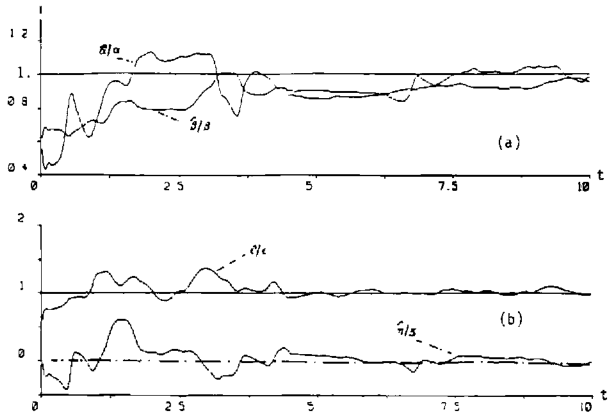
Conversely, the four unknown physical parameters are uniquely determined by α, β, ϵ, and η.

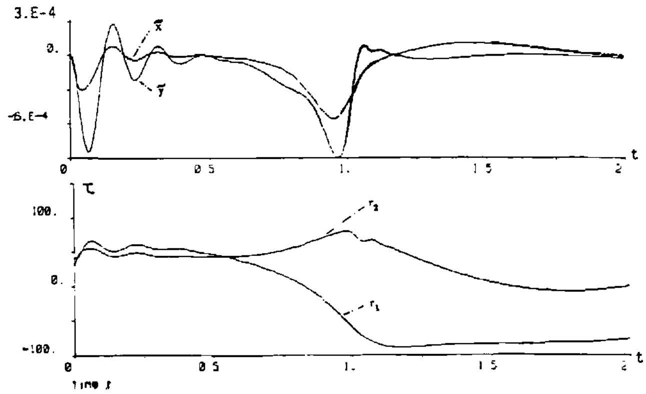
Appendix II: The Matrix $\dot{\mathbf{H}} - 2\mathbf{C}$

We show here that, with a proper definition of the matrix $\mathbf{C}$, the matrix $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew-symmetric, thus making more precise the result obtained earlier from conservation of energy.

Fig. 8. Showing the convergence of the estimates for persistently exciting trajectories: (a) normalized $\hat{\alpha}$ and $\hat{\beta}$; (b) normalized $\hat{\epsilon}$ and $\hat{\eta}$.

Fig. 9. Adaptive controller in Cartesian space.





The ith element of the vector $\mathbf{C}\dot{\mathbf{q}}$ is (see, e.g., Asada and Slotine 1986)

$$\sum_{j=1}^{n} C_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_j\dot{q}_k, \qquad \text{(A1)}$$

where the Christoffel coefficients h_{ijk} verify

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial H_{jk}}{\partial q_i}.$$

Thus, (A1) can be written

$$\sum_{j=1}^{n} C_{ij}\dot{q}_j = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial H_{ij}}{\partial q_k}\dot{q}_j\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j,$$

where we used reindexing to obtain the second term on the right side. Now take

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{n}\frac{\partial H_{ij}}{\partial q_k}\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k$$
$$= \frac{1}{2}\dot{H}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k,$$

and let $\mathbf{W} = \dot{\mathbf{H}} - 2\mathbf{C}$. Then

$$W_{ij} = \sum_{k=1}^{n}\left(\frac{\partial H_{jk}}{\partial q_i} - \frac{\partial H_{ik}}{\partial q_j}\right)\dot{q}_k.$$

Thus for all i, j

$$W_{ij} = -W_{ji},$$

which shows the skew-symmetry of $\dot{\mathbf{H}} - 2\mathbf{C}$. Although other choices of C_{ij} could satisfy (A1), they usually do not possess this skew-symmetry property.

References

An, C. G., Atkeson, C. G., and Hollerbach, J. M. 1986 (San Francisco, Calif.). Experimental determination of the effect of feedforward control on trajectory tracking errors. *IEEE Int. Conf. Robotics and Automation.*

Arimoto, S., and Miyazaki, F. 1984 (Bretton Woods, N.H.). On the stability of P.I.D. feedback with sensory information. *Robotics Research,* eds. M. Brady and R. P. Paul. Cambridge: MIT Press.

Arimoto, S., Kawamura, S., Miyazaki, F., and Tamaki, S. 1985 (Fort Lauderdale, Fla.). Learning control theory for dynamical systems. *IEEE Conf. Decision and Control.*

Asada, H., and Slotine, J-J. E. 1986. *Robot Analysis and Control.* New York: John Wiley and Sons.

Åström, K. J. 1983. Theory and applications of adaptive control: A survey. *Automatica* 19:471.

Åström, K. J. 1984 (Las Vegas, Nev.). Interaction between excitation and unmodeled dynamics in adaptive control. *IEEE Conf. Decision and Control.*

Atkeson, C. G., An, C. G., and Hollerbach, J. M. 1985 (Gouvieux, France). Estimation of inertial parameters of manipulator loads and links. *Robotics Research*, eds. O. Faugeras and G. Giralt. Cambridge: MIT Press.

Craig, J. J., Hsu, P., and Sastry, S. 1986 (San Francisco, Calif.). Adaptive control of mechanical manipulators. *IEEE Int. Conf. Robotics and Automation.*

Dubowsky, S., and DesForges, D. T. 1979. The application of model-reference adaptive control to robotic manipulators. *J. Dyn. Syst. Meas. Contr.* 101:193–200.

Hsia, T. C. 1986 (San Francisco, Calif.). Adaptive control of robot manipulators—A review. *IEEE Int. Conf. Robotics and Automation.*

Khatib, O. 1986 (Osaka, Japan). *U.S.–Japan Symp. Flexible Automation.*

Khosla, P., and Kanade, T. 1985 (Fort Lauderdale, Fla.). Parameter identification of robot dynamics. *IEEE Conf. Decision and Control.*

Koditschek, D. 1984 (Las Vegas, Nev.). Natural motion of robot arms. *IEEE Conf. Decision and Control.*

Koivo, A. J. 1986 (San Francisco, Calif.). Force-position-velocity control with self-tuning for robotic manipulators. *IEEE Int. Conf. Robotics and Automation.*

Morgan, A. P., and Narendra, K. S. 1977. On the uniform asymptotic stability of certain linear nonautonomous differential equations. *SIAM J. Control Optim.* 15.

Slotine, J. J. E. 1984. Sliding controller design for nonlinear systems. *Int. J. Control* 40(2).

Slotine, J. J. E. 1985. The robust control of robot manipulators. *Int. J. Robotics Research* 4(2).

Slotine, J. J. E. 1986 (San Francisco, Calif.). On robustness and adaptation in robot control. *IEEE Int. Conf. Robotics and Automation.*

Slotine, J. J. E., and Coetsee, J. A. 1986. Adaptive sliding controller synthesis for nonlinear systems. *Int. J. Control* 42:6.

Slotine, J. J. E., and Li, W. 1987 (Raleigh, N.C.). Adaptive manipulator control: A case study. *IEEE Int. Conf. Robotics and Automation.*

Slotine, J. J. E., and Sastry, S. S. 1983. Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators. *Int. J. Control* 38(2).