

Adaptive control of flexible multilink manipulators

MEHMET BODUR† and M. EROL SEZER‡

An adaptive self-tuning control scheme is developed for end-point position control of flexible manipulators. The proposed scheme has three characteristics. First, it is based on a dynamic model of a flexible manipulator described in cartesian coordinates, which eliminates the burden and inaccuracy of translating a desired end-point trajectory to joint coordinates using inverse kinematic relations. Second, the effect of flexibility is included in the dynamic model by approximating flexible links with a number of rigid sublinks connected at fictitious joints. The relatively high stiffness of the fictitious joints is shown to result in a decomposition of the model into two subsystems operating at different rates. This allows for stabilization of the oscillatory modes associated with the flexible links by a fast feedback control in addition to a slower control for trajectory tracking. Third, the control is constructed from measurements of the end-point position and deformations of the flexible links, with the manipulator parameters required to form the control obtained using a recursive least-squares estimation algorithm, which is fast enough for on-line applications. Satisfactory results are obtained from digital simulation of a two-link flexible manipulator.

1. Introduction

Conventional industrial manipulators are usually made stiff and bulky to avoid vibrations and thus to achieve precision in motion control. Several considerations such as lower arm cost, higher motion speeds, better energy efficiency, safer operation, and improved mobility resulted in a new generation of manipulators with lightweight, flexible links (Book 1985). Two major problems in the motion control of flexible manipulators are to avoid oscillations due to flexibility distributed along the links and to achieve accuracy in the positioning of the end-effector.

While several control schemes have been established for rigid manipulators (see, for example, Luh *et al.* 1980, Freund 1982, Balestrino *et al.* 1983, Koivo and Guo 1983, Lee and Lee 1984, Slotine 1985, Koivo 1985, Souissi and Koivo 1987, Craig 1988), a direct application of these schemes to flexible manipulators gives unsatisfactory results owing to neglected elastic modes. Effective control of flexible manipulators requires inclusion of elastic modes in the dynamic model (Book 1984, Judd and Falkenburg 1985, Çetinkut *et al.* 1986), and additional control to stabilize these vibrational modes. The singular perturbations approach (Kokotovic 1984) based on a decomposition of the manipulator dynamics into slow and fast modes associated with the joint variables and elastic deformations has been reported to give satisfactory performance (Siciliano and Book 1988).

Another difficulty associated with the control of flexible manipulators is the

Received 26 March 1992. Revised 16 August 1992.

† Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey.

‡ Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey.

solution of inverse kinematic relations in order to express a desired end-point trajectory in terms of joint variables. Although it is possible to calculate the static deflections of the links due to gravity at every point of a specific trajectory, this is a tedious task as all calculations have to be redone for each new trajectory. Besides, deflections of the links during fast motions do not exactly match the precalculated static deflections. The pseudo-link concept introduced by Nemir *et al.* (1988) allows for a simple description of the end-point trajectory in terms of pseudo-link angles, considerably simplifying the inverse kinematics problem. The pseudo-link angles needed for feedback purposes are obtained from the measurements of the joint angles and of the deflections of the links using strain gauges attached to them. The pseudo-link concept has also been used by King *et al.* (1990) in a two-time-scale approach to control of flexible manipulators, which promotes the application of model reference adaptive control schemes (Craig 1988).

An obvious way of avoiding the inverse kinematics problem is to model the manipulator dynamics and describe the desired trajectory in cartesian base-frame coordinates. Although end-point position control in cartesian and other task-oriented coordinates has been studied by many researchers (e.g. Luh *et al.* 1980, Takegaki and Arimoto 1981, Koivo 1985), the resulting control schemes require extensive computational effort, and are, therefore, impractical for real-time applications.

The objective of this paper is to develop a computationally attractive adaptive scheme for end-point position control of flexible manipulators suitable for on-line applications. For this purpose, flexible links are modelled by a simple bulk approximation using a series of fictitious joints, which represent lightly damped vibration modes. This bulk approximation allows for the derivation of dynamic equations of a flexible manipulator using the systematic formulations such as Lagrangian (Uicker *et al.* 1964), or generalized d'Alembert (Lee *et al.* 1983), developed for rigid manipulators. Using the jacobian of the bulk approximated rigid manipulator, the dynamic equations are expressed in terms of cartesian end-point variables and fictitious-joint angles. The resulting model is discretized into an auto-regressive externally excited system, which is shown to possess a two-time-scale property. This allows for a decomposition of the control law into two parts, one to force the slow system describing the end-point dynamics to track a desired trajectory, and the other to stabilize the fast subsystem describing the vibrational dynamics. The slow control consists of a feedforward computed-torque component and a feedback from cartesian tracking errors, as standard in motion control of rigid manipulators (Koivo 1989). The fast control, which is a feedback from the fast components of the fictitious-joint variables, is approximated by a feedback from the actual (measured) fictitious-joint variables and their quasi-constant components, which can be expressed in terms of the end-point variables.

Implementation of the derived control law requires that the manipulator parameters be known at every point of the trajectory. They can either be calculated from the measured end-point variables and link deformations using the employed formulation, or be estimated using an on-line adaptive scheme. The latter involves much less computational effort, and is, therefore, more suitable for on-line applications.

The developed control scheme is tested by digital simulation of a two-link

manipulator with a flexible first link. The simulations are carried out in two groups. The first group consists of simulations of collocated control schemes without and with compensation of the static torque due to gravity. The second group includes simulations of the developed control scheme with the manipulator parameters computed from the bulk model using the generalized d'Alembert formulation, and with the parameters obtained by a recursive least-squares estimation (RLSE) algorithm (Goodwin and Sin 1984) to obtain a self-tuning regulatory control. Simulation results indicate that the self-tuning approach can be successfully implemented in real-time position control of flexible manipulators.

2. Dynamic modelling

The dynamics of a flexible link can be accurately modelled as a distributed parameter system resulting in a set of partial differential equations and initial/boundary conditions as given by Cannon and Schmitz (1984). Such a model describes an infinite number of modes, and is extremely complicated for practical use. Studies by Cannon and Schmitz on identification of finite-order transfer functions of flexible links showed that the modal gains beyond the third flexible mode are negligible. Thus, the dynamics of a flexible link can adequately be approximated by a second-order bulk model, which consists of two rigid sublinks joined by a fictitious joint as shown in Fig. 1. Note that the angular displacement of the fictitious joint is a measure of the deflection of the link.

Consider a flexible manipulator consisting of n joints, and let the m flexible links ($m \leq n$) be modelled by second-order bulk approximations. Using the Lagrange-Euler or the generalized d'Alembert formulations, the dynamical model of such a manipulator can be obtained (King *et al.* 1990) as

$$\begin{bmatrix} A_{qq} & A_{q\delta} \\ A_{\delta q} & A_{\delta\delta} \end{bmatrix} \begin{bmatrix} \ddot{q}(t) \\ \ddot{\delta}(t) \end{bmatrix} + \begin{bmatrix} B_{qq} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}(t) \\ \dot{\delta}(t) \end{bmatrix} + \begin{bmatrix} c_q \\ c_\delta \end{bmatrix} = \begin{bmatrix} \tau_q(t) \\ \tau_\delta(t) \end{bmatrix} \quad (2.1)$$

where $q \in R^n$ and $\delta \in R^m$ are generalized displacements of the actual and fictitious joints, τ_q and τ_δ are the corresponding generalized torques, and

$$A(q, \delta) = \begin{bmatrix} A_{qq} & A_{q\delta} \\ A_{\delta q} & A_{\delta\delta} \end{bmatrix}, \quad B = \begin{bmatrix} B_{qq} & 0 \\ 0 & 0 \end{bmatrix}, \quad C(q, \delta, \dot{q}, \dot{\delta}) = \begin{bmatrix} c_q \\ c_\delta \end{bmatrix}$$

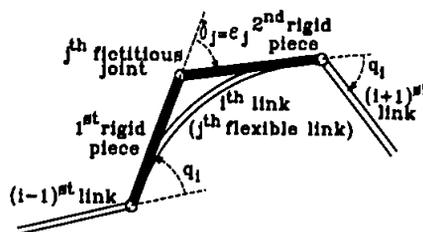


Figure 1. Bulk model of a flexible link.

are, respectively, the non-singular matrix of inertial coefficients, the matrix of the damping coefficients of the actuators at the actual joints, and the vector of coriolis, centrifugal and gravitational torques. The actuator torques, $\tau_q(t)$, in (2.1) are external inputs to the manipulator and can be chosen freely within the limitations of the actuators. The torques of the fictitious joints, however, are determined by the internal dynamics of the flexible links as

$$\tau_\delta(t) = -K_s\delta(t) - K_d\dot{\delta}(t) \quad (2.2)$$

where $K_s = \text{diag}\{K_{s1}, \dots, K_{sm}\}$ and $K_d = \text{diag}\{K_{d1}, \dots, K_{dm}\}$ contain the spring constants and the viscous dampings of the fictitious joints.

Equations (2.1) and (2.2) constitute a dynamic model of a flexible manipulator in the joint coordinates. To obtain a model in cartesian coordinates, we assume that the end-point displacement vector $p(t)$ has the same dimension as $q(t)$, and write

$$\begin{bmatrix} p \\ \varepsilon \end{bmatrix} = \begin{bmatrix} f(q, \delta) \\ \delta \end{bmatrix} \quad (2.3)$$

Accordingly

$$\left. \begin{aligned} \begin{bmatrix} \dot{p} \\ \dot{\varepsilon} \end{bmatrix} &= \begin{bmatrix} J_q & J_\delta \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{\delta} \end{bmatrix} \\ \begin{bmatrix} \ddot{p} \\ \ddot{\varepsilon} \end{bmatrix} &= \begin{bmatrix} J_q & J_\delta \\ 0 & I \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} \dot{J}_q & \dot{J}_\delta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{\delta} \end{bmatrix} \end{aligned} \right\} \quad (2.4)$$

where $J_q = \partial f / \partial q$, $\dot{J}_q = [\sum_{l=1}^n (\partial^2 f_i / \partial q_j \partial q_l) \dot{q}_l + \sum_{l=1}^m (\partial^2 f_i / \partial q_j \partial \delta_l) \dot{\delta}_l]_{n,n}$, and J_δ and \dot{J}_δ are defined similarly. To avoid pathological cases, we assume that the jacobian $J_q(q, \delta)$ is non-singular and well-conditioned along the desired trajectory so that the mapping in (2.3) is invertible. Applying the transformation in (2.3) to the model in (2.1) and (2.2), we get

$$\begin{aligned} \begin{bmatrix} A_{pp} & A_{p\varepsilon} \\ A_{\varepsilon p} & A_{\varepsilon\varepsilon} \end{bmatrix} \begin{bmatrix} \ddot{p}(t) \\ \ddot{\varepsilon}(t) \end{bmatrix} + \begin{bmatrix} B_{pp} & B_{p\varepsilon} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{p}(t) \\ \dot{\varepsilon}(t) \end{bmatrix} + \begin{bmatrix} c_p \\ c_\varepsilon \end{bmatrix} \\ = \begin{bmatrix} \tau_q(t) \\ -K_s\varepsilon(t) - K_d\dot{\varepsilon}(t) \end{bmatrix} \end{aligned} \quad (2.5)$$

where

$$\left. \begin{aligned} \begin{bmatrix} A_{pp} & A_{p\varepsilon} \\ A_{\varepsilon p} & A_{\varepsilon\varepsilon} \end{bmatrix} &= \begin{bmatrix} A_{qq} & A_{q\delta} \\ A_{\delta q} & A_{\delta\delta} \end{bmatrix} \begin{bmatrix} J_q^{-1} & -J_q^{-1}J_\delta \\ 0 & I \end{bmatrix} \\ [B_{pp} \quad B_{p\varepsilon}] &= B_{qq}[J_q^{-1} \quad -J_q^{-1}J_\delta] \end{aligned} \right\} \quad (2.6)$$

where the terms involving \dot{J}_q and \dot{J}_δ are included in c_p and c_ε . Note that the identity $\varepsilon = \delta$ is used only to change the notation so that $A_{\varepsilon\varepsilon}$ in (2.5) is distinguished from $A_{\delta\delta}$ in (2.1). Note also that while $B_{q\delta} = 0$, $B_{p\varepsilon} \neq 0$.

Finally, we discretize (2.5) to get

$$\left. \begin{aligned} T^{-2}A_{pp}\Delta^2 p_k + T^{-2}A_{p\varepsilon}\Delta^2 \varepsilon_k + T^{-1}B_{pp}\Delta p_k + T^{-1}B_{p\varepsilon}\Delta \varepsilon_k + c_p &= \tau_k \\ T^{-2}A_{\varepsilon p}\Delta^2 p_k + T^{-2}A_{\varepsilon\varepsilon}\Delta^2 \varepsilon_k + K_s\varepsilon_k + T^{-1}K_d\Delta \varepsilon_k + c_\varepsilon &= 0 \end{aligned} \right\} \quad (2.7)$$

where $p_k = p(kT)$, $\varepsilon_k = \varepsilon(kT)$, and $\tau_k = \tau_q(kT)$ with T being the discretization step, the difference operator Δ is defined as

$$\Delta f_k = f_k - f_{k-1} \quad (2.8)$$

and the subscript k is omitted from the time-varying coefficients A_{pp} , \dots , $B_{p\varepsilon}$, c_p and c_ε for simplicity in notation.

3. A non-collocated two-time-scale control

It has already been shown by Siciliano and Book (1988) that, for sufficiently large values of the spring constants of the fictitious joints, the joint-space model in (2.1) possesses a two-time-scale property. Although the analysis of Siciliano and Book (1988) can be imitated for the cartesian-space model in (2.5) to obtain a composite control consisting of slow and fast parts, which is then suitably discretized for use in (2.7), we prefer to derive the control law directly using the discrete-time model in (2.7) for the purpose of discrete implementation of the adaptive controller.

We assume that elements of K_s are sufficiently large compared with other parameters in (2.7), and write $K_s = T^{-2}K'_s$. As shown in the Appendix, the system has a two-time-scale property, which allows for a decomposition of the variables into quasi-constant and fast components as

$$p_k \simeq \bar{p}_k, \quad \varepsilon_k \simeq \bar{\varepsilon}_k + \hat{\varepsilon}_k, \quad \tau_k \simeq \bar{\tau}_k + \hat{\tau}_k \quad (3.1)$$

Accordingly, (2.7) is decomposed into two subsystems:

$$T^{-2}A_{pp}\Delta^2\bar{p}_k + T^{-1}B_{pp}\Delta\bar{p}_k + c_p = \bar{\tau}_k \quad (3.2)$$

describing the slow dynamics, and

$$\Delta^2\hat{\varepsilon}_k + E_{\varepsilon\varepsilon}K'_s\hat{\varepsilon}_k = T^2E_{\varepsilon p}\hat{\tau}_k \quad (3.3)$$

describing the fast dynamics, where

$$\begin{bmatrix} E_{pp} & E_{p\varepsilon} \\ E_{\varepsilon p} & E_{\varepsilon\varepsilon} \end{bmatrix} = \begin{bmatrix} A_{pp} & A_{p\varepsilon} \\ A_{\varepsilon p} & A_{\varepsilon\varepsilon} \end{bmatrix}^{-1} \quad (3.4)$$

The quasi-constant component of ε_k is specified in terms of \bar{p}_k and $\bar{\tau}_k$ as

$$\bar{\varepsilon}_k = -K_s^{-1}[A_{\varepsilon p}A_{pp}^{-1}(\bar{\tau}_k - T^{-1}B_{pp}\Delta\bar{p}_k - c_p) + c_\varepsilon] \quad (3.5)$$

At this point it is appropriate to note that, as in all applications of singular perturbations theory, whether a parameter could be considered to be 'large' relative to others depends on the eigenstructure of the system rather than its absolute magnitude. Although our theoretical development above formally shows that the assumption $K_s = T^{-2}K'_s$ leads to a separation of the time scales of the system, the validity of this result should be verified either by a simulation or by actual implementation.

The above decomposition of the manipulator dynamics into slow and fast components is valid only if the fast subsystem in (3.3) is asymptotically stable. To achieve stability of $\hat{\varepsilon}_k$, we apply a fast feedback control of the form

$$\hat{\tau}_k = \hat{K}_1\hat{\varepsilon}_{k-1} + \hat{K}_2\hat{\varepsilon}_{k-2} \quad (3.6)$$

which results in

$$(I + E_{\varepsilon\varepsilon}K'_3)\hat{\varepsilon}_k - (2I + T^2E_{\varepsilon p}\hat{K}_1)\hat{\varepsilon}_{k-1} + (I - T^2E_{\varepsilon p}\hat{K}_2)\hat{\varepsilon}_{k-2} = 0 \quad (3.7)$$

From (3.7) we observe that if \hat{K}_1 and \hat{K}_2 are chosen to satisfy

$$\left. \begin{aligned} E_{\varepsilon p}\hat{K}_1 &= -T^{-2}[(I + E_{\varepsilon\varepsilon}K'_3)\hat{Z}_1 + 2I] \\ E_{\varepsilon p}\hat{K}_2 &= -T^{-2}[(I + E_{\varepsilon\varepsilon}K'_3)\hat{Z}_2 - I] \end{aligned} \right\} \quad (3.8)$$

then the closed-loop fast subsystem is described by

$$\hat{\varepsilon}_k + \hat{Z}_1\hat{\varepsilon}_{k-1} + \hat{Z}_2\hat{\varepsilon}_{k-2} = 0 \quad (3.9)$$

Now \hat{Z}_1 and \hat{Z}_2 can be chosen arbitrarily to stabilize the fast subsystem by placing its $2m$ poles at desired locations. Note that this also allows for decoupling of the fast dynamics, which means the vibrational modes associated with the fictitious joints can be controlled independently. We also note that since the bulk approximations of the flexible links contain only a single fictitious joint, we have $m \leq n$, so that equations in (3.8) are generically solvable for \hat{K}_1 and \hat{K}_2 .

With the fast subsystem stabilized, it now remains to choose the slow component of the control input τ_k to achieve a satisfactory performance of the slow subsystem in (3.2). This, however, is the standard tracking problem for rigid manipulators, for which a solution is available (Koivo 1989): We simply choose

$$\bar{\tau}_k = T^{-2}A_{pp}\Delta^2 p_k^d + T^{-1}B_{pp}\Delta p_k^d + c_p + \bar{K}_1 e_{k-1}^p + \bar{K}_2 e_{k-2}^p \quad (3.10)$$

where p_k^d describes the desired trajectory of the end-point, and $e_k^p = p_k^d - \bar{p}_k$ is the deviation of the actual trajectory from the desired one. Note that the first three terms in (3.10) constitute a feedforward control (computed torque) which decouples the manipulator's slow dynamics, and the last two terms form a feedback control to achieve a desired performance of the decoupled dynamics. Applying the control in (3.10) to the slow subsystem in (3.2), and rearranging the terms, we obtain

$$(A_{pp} + TB_{pp})e_k^p + (-2A_{pp} - TB_{pp} + T^2\bar{K}_1)e_{k-1}^p + (A_{pp} + T^2\bar{K}_2)e_{k-2}^p = 0 \quad (3.11)$$

Letting

$$\left. \begin{aligned} \bar{K}_1 &= T^{-2}[A_{pp} + (A_{pp} + TB_{pp})(\bar{Z}_1 + I)] \\ \bar{K}_2 &= T^{-1}B_{pp} + T^{-2}(A_{pp} + TB_{pp})(\bar{Z}_2 - I) \end{aligned} \right\} \quad (3.12)$$

(3.11) is reduced to

$$e_k^p + \bar{Z}_1 e_{k-1}^p + \bar{Z}_2 e_{k-2}^p = 0 \quad (3.13)$$

Clearly, the error dynamics described by (3.13) can be controlled as desired by choosing \bar{Z}_1 and \bar{Z}_2 properly to place the $2n$ poles arbitrarily.

The composite control is the sum of the fast and slow controls in (3.6) and (3.10). Since $\hat{\varepsilon}_k$ is not directly measurable, we approximate $\hat{\tau}_k$ in (3.6), by using $\hat{\varepsilon}_k \approx \varepsilon_k - \bar{\varepsilon}_k$ and letting $\bar{\varepsilon}_{k-1} \approx \bar{\varepsilon}_{k-2} \approx \bar{\varepsilon}_k$, as

$$\hat{\tau}_k = \hat{K}_1 \varepsilon_{k-1} + \hat{K}_2 \varepsilon_{k-2} - (\hat{K}_1 + \hat{K}_2)\bar{\varepsilon}_k \quad (3.14)$$

Now the expression for $\bar{\varepsilon}_k$ in (3.5) can be used in (3.14) to get

$$\begin{aligned} \hat{\tau}_k &= \hat{K}_1 \varepsilon_{k-1} + \hat{K}_2 \varepsilon_{k-2} \\ &+ (\hat{K}_1 + \hat{K}_2) K_s^{-1} [A_{\varepsilon p} A_{pp}^{-1} (\bar{\tau}_k - T^{-1} B_{pp} \Delta \bar{p}_k - c_p) + c_\varepsilon] \end{aligned} \quad (3.15)$$

which, like the slow control $\bar{\tau}_k$, involves only measurable variables.

4. Adaptive self-tuning non-collocated control

Implementation of the control law derived in the previous section requires that the manipulator parameters A_{pp} , \dots , $B_{p\varepsilon}$, c_p and c_ε be known. If the desired trajectory is known beforehand, then these parameters can be calculated and stored off-line at several points along the trajectory: under the assumption of perfect tracking without any oscillations, we set $p_k = \bar{p}_k = p_k^d$ and $\hat{\varepsilon}_k = 0$, so that $e_k^d = 0$ and

$$\varepsilon_k = \bar{\varepsilon}_k = -K_s^{-1} (T^{-2} A_{\varepsilon p} \Delta^2 p_k + c_\varepsilon)$$

which is obtained by substituting (3.10) with $e_k^d = 0$ into (3.5). Using inverse kinematic relations, actual and fictitious joint displacements q_k and δ_k are calculated from p_k and ε_k , which, in turn, are used to calculate the parameters A_{qq} , \dots , B_{qq} in (2.1), and finally A_{pp} , \dots , $B_{p\varepsilon}$ in (2.7) via (2.6). Obviously, this method involves extensive computations which are not suitable for on-line applications.

An alternative to calculating the manipulator parameters is to estimate them from input/output data using the RLSE algorithm (Goodwin and Sin 1984). To reduce the computational effort, we make a simplifying assumption that $B_{pp} = 0$ and $B_{p\varepsilon} = 0$. Then, defining

$$v_k = T^{-1} \Delta p_k, \quad w_k = T^{-1} \Delta \varepsilon_k \quad (4.1)$$

which represent the approximate cartesian velocity of the end-point, and the angular velocities of the fictitious joints, respectively, (2.7) becomes

$$\begin{bmatrix} A_{pp} & A_{p\varepsilon} \\ A_{\varepsilon p} & A_{\varepsilon\varepsilon} \end{bmatrix} \begin{bmatrix} \Delta v_k \\ \Delta w_k \end{bmatrix} = \begin{bmatrix} T\tau_k \\ -TK_s \varepsilon_k - K_d \Delta \varepsilon_k \end{bmatrix} + \begin{bmatrix} -Tc_p \\ -Tc_\varepsilon \end{bmatrix} \quad (4.2)$$

Letting

$$y_k = \begin{bmatrix} \Delta v_k \\ \Delta w_k \end{bmatrix}, \quad u_k = \begin{bmatrix} T\tau_k \\ -TK_s \varepsilon_k - K_d \Delta \varepsilon_k \end{bmatrix}, \quad c_k = \begin{bmatrix} c_p \\ c_\varepsilon \end{bmatrix} \quad (4.3)$$

Equations (4.2) can be written in the standard form

$$y_k = \Theta_k \phi_k \quad (4.4)$$

where y_k is the observation vector

$$\phi_k = [u_k^T \ 1]^T$$

is the input, and

$$\Theta_k = [E_k - TE_k c_k]$$

is the parameter matrix, with E_k denoting the matrix in (3.4).

Assuming slow variations of the parameters, a least-squares estimate of Θ_k which minimizes the weighted sum

$$J_k = \sum_{l=0}^k \gamma^{k-l} (y_l - \Theta_l \phi_l)^T (y_l - \Theta_l \phi_l) \quad (4.5)$$

is obtained (Goodwin and Sin 1984) as

$$\left. \begin{aligned} \hat{\Theta}_k &= \hat{\Theta}_{k-1} + \alpha_k (y_k - \hat{\Theta}_{k-1} \phi_k) \phi_k^T P_{k-1} \\ P_k &= \gamma^{-1} (P_{k-1} - \alpha_k P_{k-1} \phi_k \phi_k^T P_{k-1}) \end{aligned} \right\} \quad (4.6)$$

where $\alpha_k = (1 + \phi_k^T P_{k-1} \phi_k)^{-1}$, and $0 < \gamma \leq 1$ is the forget factor.

In (4.6), the input vector ϕ_k is constructed from the calculated value of the input torques, τ_k , and the measurement of the deflections, ε_k , of the fictitious joints. Once Θ_k is calculated, estimates of the parameters A_{pp} , \dots , $A_{\varepsilon\varepsilon}$, c_p and c_ε are obtained by inverting the $(n+m) \times (n+m)$ matrix E_k .

In implementing the RLSE algorithm, the initial value of Θ_k can either be calculated using inverse kinematic relations at the start of the trajectory assuming static deflections, or obtained by a $(n+m+1)$ -step non-recursive least-squares estimation scheme. It may also be necessary to add suitable continuous persistent oscillations to the input τ_k in order to excite all the modes of the manipulator (Goodwin and Sin 1984).

With the manipulator parameters estimated, the slow control is given by

$$\bar{\tau}_k = T^{-2} A_{pp}^c [\Delta^2 p_k^d + (\bar{Z}_1 + 2I) e_{k-1}^p + (\bar{Z}_2 - I) e_{k-2}^p] + c_p^c \quad (4.7)$$

which is obtained from (3.10) and (3.12) with $B_{pp} = 0$, and superscript e denoting the estimated value of the indicated parameter. Substituting (4.7) into (3.15), the fast control becomes

$$\begin{aligned} \hat{\tau}_k &= \hat{K}_1 \varepsilon_{k-1} + \hat{K}_2 \varepsilon_{k-2} + (\hat{K}_1 + \hat{K}_2) K_s^{-1} c_\varepsilon^e \\ &+ T^{-2} (\hat{K}_1 + \hat{K}_2) K_s^{-1} A_{\varepsilon p}^c [\Delta^2 p_k^d + (\bar{Z}_1 + 2I) e_{k-1}^p + (\bar{Z}_2 - I) e_{k-2}^p] \end{aligned} \quad (4.8)$$

The expressions for $\bar{\tau}_k$ and $\hat{\tau}_k$ above show a further advantage of assuming $B_{pp} = 0$; that is, the inverse of A_{pp} , required in (3.15), is avoided.

5. Simulation results

The proposed two-time-scale control is tested by digital simulation of a planar two-link manipulator shown in Fig. 2. Performances of non-collocated control based on calculated and estimated parameters are compared with each other and with a simple collocated control based on the rigidity assumption.

5.1. Manipulator and trajectory descriptions

The manipulator shown in Fig. 2 consists of two links, the first one being flexible. The links have lengths 0.5 m and masses 10 kg each. The effective inertias and damping coefficients of the actuators are 6.35 kg m² and 2.12 N m s for the first joint, and 1.30 kg m² and 0.43 N m s for the second joint.

The flexible link is modelled by a second-order bulk approximation consisting of two sublinks having length 0.25 m each. Assuming that an actuator is placed at the second joint, the mass of the flexible link is distributed between the sublinks as 2 kg and 8 kg respectively. The spring constant and the damping

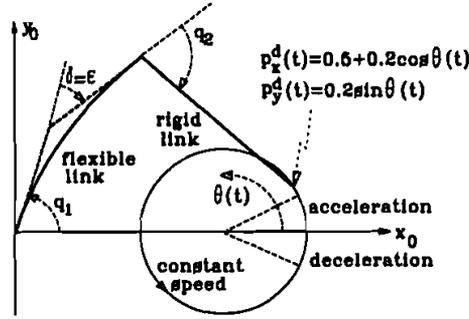


Figure 2. Simulated flexible manipulator and the desired trajectory.

coefficient of the fictitious joint are assumed to be $K_s = 400 \text{ Nm}$ and $K_d = 0 \text{ Nms}$ respectively. Note that the numerical value of K_s is large enough to justify a singularly perturbed model.

The trajectory to be followed by the end-point is a circular path in the plane of the manipulator with radius 0.2 m and centred at $(0.5 \text{ m}, 0)$ as also shown in Fig. 2. It is to be traced in 4 s starting from the point $(0.7 \text{ m}, 0)$, with an angular velocity and acceleration as specified in Fig. 3(a). Corresponding displacements of the end-point in cartesian coordinates are shown in Fig. 3(b). This trajectory is chosen particularly to demonstrate the effectiveness of the proposed control scheme under large inertial variations along the path.

The simulations are performed for a discrete model of the manipulator corresponding to a sampling period of $T = 0.005 \text{ s}$. That is, the manipulator parameters are updated (either by calculation or by estimation) once every 0.005 s , as well as the value of the control input. The Runge-Kutta method with a step size of $h = 0.001 \text{ s}$ is used for numerical integration of the manipulator equations during each sampling period.

5.2. Collocated control

The collocated control is designed for a rigid model given as

$$A_{qq}\ddot{q}(t) + B_{qq}\dot{q}(t) + c_q = \tau_q(t) \quad (5.1)$$

which is obtained from (2.1) by neglecting the flexible modes. Discretization of (5.1) yields

$$T^{-2}A_{qq}\Delta^2 q_k + T^{-1}B_{qq}\Delta q_k + c_q = \tau_k \quad (5.2)$$

which is of the form of (3.2). The control required to track a desired trajectory q_k^d specified in joint coordinates has the same structure as $\bar{\tau}_k$ in (3.10), and is given by

$$\tau_k = T^{-2}A_{qq}\Delta^2 q_k^d + T^{-1}B_{qq}\Delta q_k^d + c_q + K_{q1}e_{k-1}^q + K_{q2}e_{k-2}^q \quad (5.3)$$

where $e_k^q = q_k^d - q_k$. Hence, the choice

$$\left. \begin{aligned} K_{q1} &= T^{-2}[A_{qq} + (A_{qq} + TB_{qq})(Z_{q1} + I)] \\ K_{q2} &= T^{-1}B_{qq} + T^{-2}(A_{qq} + TB_{qq})(Z_{q2} - I) \end{aligned} \right\} \quad (5.4)$$

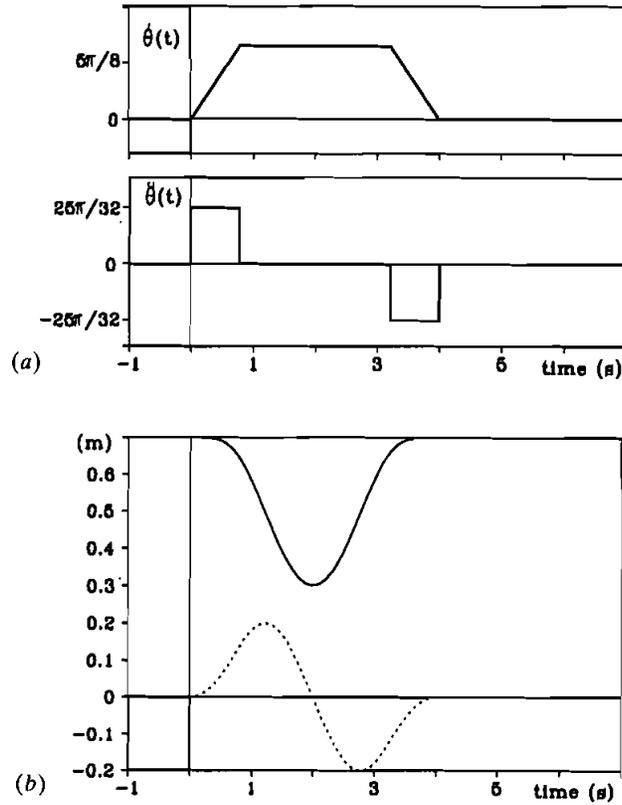


Figure 3. Description of the desired trajectory: (a) angular speed and acceleration along the desired trajectory specified in Fig. 2; (b) desired end-point displacements (p_x^d : solid, p_y^d : dotted).

results in an error equation

$$e_k^q + Z_{q1}e_{k-1}^q + Z_{q2}e_{k-2}^q = 0 \quad (5.5)$$

whose poles can be assigned arbitrarily.

Note that the control law in (5.3) involves only the measurement of the angular displacements of the actual joints, and is, therefore, much simpler to implement than the non-collocated control based on a two-time-scale model.

In the simulations, K_{q1} and K_{q2} in (5.4) are selected to place the poles of the error equation in (5.5) at $z = (1 + 20T)^{-1} = 0.8333$, which corresponds to continuous-time poles at $s = -20$. In the first simulation, parameters A_{qq} , B_{qq} and c_q are calculated at every discretization step using the generalized d'Alembert formulation along the actual trajectory q_k assuming rigid links. The desired trajectory q_k^d is obtained from the specified cartesian trajectory p_k^d using inverse kinematic relations with $\delta_k = \Delta\delta_k = 0$. The simulation results shown in Fig. 4 indicate approximately 15% maximum error in the x -direction and 29% maximum error in the y -direction relative to the radius of the circular path. The static error in the y -direction is attributed to the deflection of the flexible link due to gravity. A compensation of this static torque is expected to improve the performance of the collocated control. To verify this, a second simulation is

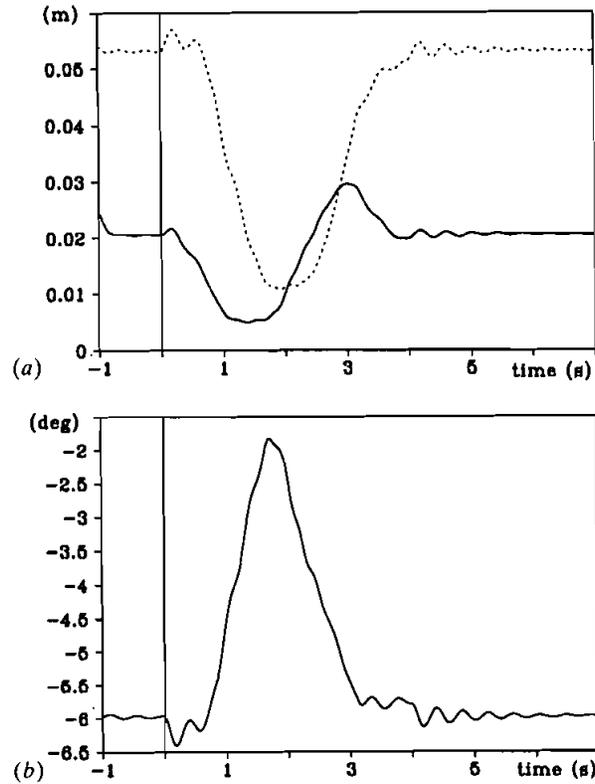


Figure 4. Response to collocated control: (a) tracking error (e_x^p : solid, e_y^p : dotted); (b) fictitious joint displacement.

carried out, in which the desired trajectory q_k^d is generated from p_k^d assuming quasi-constant deflections $\bar{\delta}_k = K_s^{-1}c_\delta$. The parameters A_{qq} , B_{qq} , c_q and c_δ are calculated from the measured values of q_k and δ_k assuming $\Delta\delta_k = 0$. The simulation results shown in Fig. 5 indicate that the static error is eliminated, and the maximum tracking errors in the x - and y -directions are reduced to approximately 1% and 4% respectively.

5.3. Non-collocated two-time-scale control

A second group of simulations are performed to test the performance of the composite control law derived in § 3. The feedback gains \bar{K}_1 and \bar{K}_2 of the slow control are chosen to place the eigenvalues of the slow subsystem at $z = (1 + 20T)^{-1} = 0.8333$, the same value used in simulation of the collocated control. The feedback gains \hat{K}_1 and \hat{K}_2 of the fast control are chosen to place the fast poles at $z = (1 + 60T)^{-1} = 0.6250$, corresponding to continuous-time poles at $s = -60$, three times faster than the slow poles, to provide rapid decay of oscillations.

Simulation results with the manipulator parameters calculated along the desired trajectory are shown in Fig. 6. The results indicate that while the tracking errors are comparable with those in the case of collocated control with

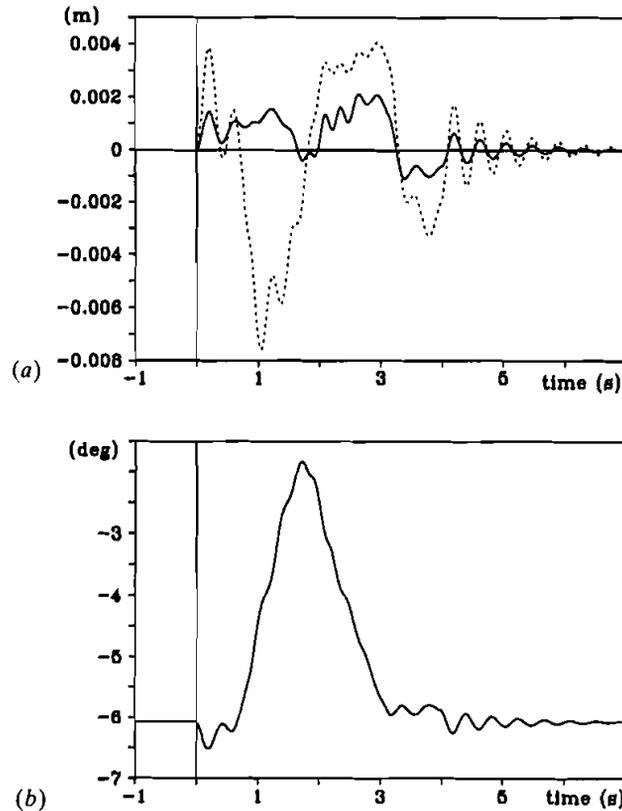


Figure 5. Response to collocated control with static compensation: (a) tracking error (e_x^p : solid, e_y^p : dotted); (b) fictitious joint displacement.

static compensation (approximately 2% maximum error in both directions), a major advantage of the composite control is observed to be the elimination of the long-lasting oscillations at the completion of the trajectory.

Finally, the simulations are repeated with the parameters estimated as described in § 4. In the estimation algorithm, zero-mean triangular signals with amplitudes $\tau_0 = 50$ Nm and periods $3T$ and $5T$ are added to the input torques to excite all modes of the manipulator. The forget factor is taken to be $\gamma = 0.8$, and the initial value of the matrix P_k in (4.6) to be $P_0 = 20I$. During estimation, P_k is reset to P_0 whenever the diagonal elements exceeded 50. The results shown in Fig. 7 indicate that the self-tuning control is as good as the one based on calculated parameters, with the added advantage of being implementable on-line.

We conclude our discussion with a few comments on the computational complexity of the proposed control scheme. Table 1 provides expressions for the total number of additions and multiplications involved at each discrete time step in terms of the number of actual and fictitious joints. Table 2 shows the corresponding numerical values as well as the CPU times (including data transfer between memory and registers) for two different processors. As can be observed from Table 2, the proposed adaptive non-collocated control scheme

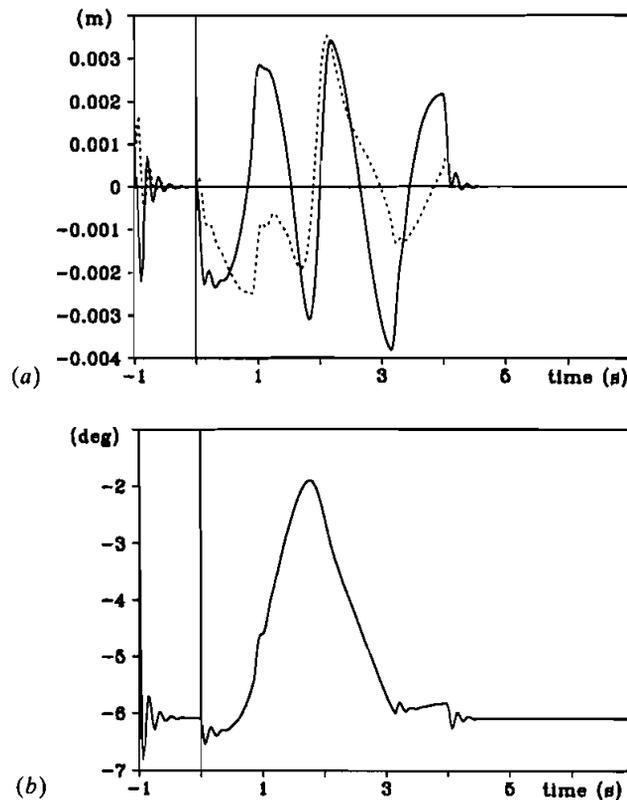


Figure 6. Response to non-collocated control based on calculated parameters: (a) tracking error (e_x^p : solid, e_y^p : dotted); (b) fictitious joint displacement.

based on a two-time-scale model is suitable for on-line control of even relatively sophisticated manipulators with up to six flexible links.

6. Conclusions

A two-time-scale adaptive control scheme is proposed for motion control of flexible manipulators. The self-tuning controller is based on a cartesian-space dynamic model of the manipulator obtained by approximating the flexible links with rigid sublinks connected at fictitious joints, and an estimation algorithm used to obtain the parameters of this model.

Bulk approximation of flexible links yields a model which is not much more complicated to control than a rigid one, but which includes a reasonable representation of the vibrational modes. Although modelling accuracy can be improved by a finer division of the flexible link into sublinks, this increases the order of the dynamic equations, and hence, the amount of computations involved in generating the control. Besides, not all the vibrational modes may be controlled by a limited number of actuators. This corresponds to a case where $m > n$, so that (3.8) may not be solvable for arbitrary \hat{Z}_1 and \hat{Z}_2 .

Working with a dynamic model in cartesian coordinates has the advantage

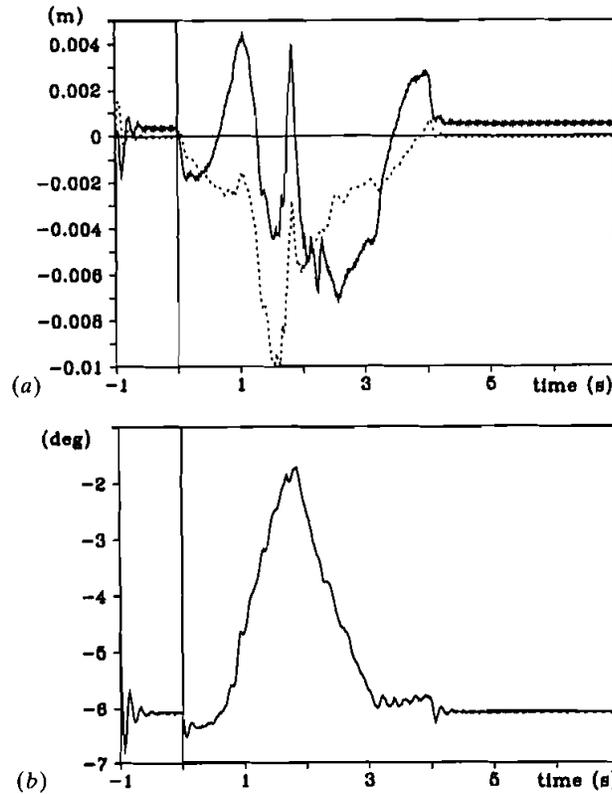


Figure 7. Response to adaptive non-collocated control: (a) tracking error (e_x^p : solid, e_y^p : dotted); (b) fictitious joint displacement.

Computation	Additions	Multiplications
Kinematics	$n^2 + 30n + 31m + nm$	$n^2 + 39n + 39m + nm$
RLSE	$3.5n^2 + 3.5m^2 + 7.5n + 8.5m + 7nm + 2$	$4n^2 + 4m^2 + 11n + 12m + 8nm + 6$
Inversion	$n^3 + n^2 + m^3 + m^2 + 3n^2m + 3m^2n + 2nm$	$n^3 + n + m^3 + m + 3n^2m + 3m^2n$
Control law	$n^2 + 13n + 2m + nm$	$n^2 + 6n + 4m + nm$
Total	$\left\{ \begin{array}{l} n^3 + 6.5n^2 + 50.5n + m^3 + 4.5m^2 + \\ 41.5m + 3n^2m + 3m^2n + 11nm + 2 \end{array} \right.$	$\left\{ \begin{array}{l} n^3 + 6n^2 + 57n + m^3 + 4m^2 + 56m + \\ 3n^3m + 3m^2n + 10nm + 6 \end{array} \right.$

Table 1. Operations count for adaptive non-collocated control.

that a desired trajectory can be specified directly in the task space. This eliminates the need for calculating the desired joint displacements using inverse kinematic relations, which is not only computationally tedious but also inaccurate during fast motions of the manipulator.

Implementing the control using an on-line estimation scheme makes the control structure independent of the coordinate system in which the manipulator is modelled. However, it requires that the end-point position and the deflections of the flexible links be measurable for feedback purposes and that the computations involved be done in reasonable time. A simplified model like the one in

n	m	Kinematics		RLSE		Inversion		Control		TOTAL		Time (ms)	
		Add	Mul	Add	Mul	Add	Mul	Add	Mul	Add	Mul	386†	386-87‡
2	1	97	123	57	76	36	30	34	22	224	251	0.63	1.17
3	3	201	252	176	219	252	222	63	48	692	741	1.88	3.52
5	3	283	352	289	353	576	520	111	82	1259	1307	3.35	6.30
6	6	438	540	602	720	1872	1740	162	132	3074	3132	8.09	15.2

† 80386 (30 MHz) processor with 32 bit signed integer arithmetic.

‡ 80386-87 (30 MHz) processor with 64 bit floating point arithmetic.

Table 2. Number of operations and CPU times.

§ 4 helps reduce the computational effort without causing any significant degradation of the controller performance.

Appendix

Derivation of the two-time-scale model

Defining the state variables

$$x_{1k} = p_k, \quad x_{2k} = T^{-1}\Delta p_k, \quad z_{1k} = T^{-2}\varepsilon_k, \quad z_{2k} = T^{-2}\Delta\varepsilon_k \quad (\text{A } 1)$$

Equations (2.7) can be written in state space form

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & A_{pp} & 0 & TA_{p\varepsilon} \\ 0 & 0 & I & 0 \\ 0 & A_{\varepsilon p} & 0 & TA_{\varepsilon\varepsilon} \end{bmatrix} \begin{bmatrix} \Delta x_{1k} \\ \Delta x_{2k} \\ \Delta z_{1k} \\ \Delta z_{2k} \end{bmatrix} = \begin{bmatrix} 0 & T & 0 & 0 \\ 0 & -TB_{pp} & 0 & -T^2B_{p\varepsilon} \\ 0 & 0 & 0 & I \\ 0 & 0 & -TK'_s & -T^2K_d \end{bmatrix} \begin{bmatrix} x_{1k} \\ x_{2k} \\ z_{1k} \\ z_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ T(\tau_k - c_p) \\ 0 \\ -Tc_\varepsilon \end{bmatrix} \quad (\text{A } 2)$$

where $K'_s = T^2K_s$. Using (3.4), (A 2) can be manipulated into

$$\begin{bmatrix} \Delta x_{1k} \\ \Delta x_{2k} \\ \Delta z_{1k} \\ \Delta z_{2k} \end{bmatrix} = \begin{bmatrix} 0 & T & 0 & 0 \\ 0 & -TE_{pp}B_{pp} & -TE_{p\varepsilon}K'_s & -T^2(E_{pp}B_{p\varepsilon} + E_{p\varepsilon}K_d) \\ 0 & 0 & 0 & I \\ 0 & -E_{\varepsilon p}B_{pp} & -E_{\varepsilon\varepsilon}K'_s & -T(E_{\varepsilon p}B_{p\varepsilon} + E_{\varepsilon\varepsilon}K_d) \end{bmatrix} \begin{bmatrix} x_{1k} \\ x_{2k} \\ z_{1k} \\ z_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ TE_{pp}(\tau_k - c_p) - TE_{p\varepsilon}c_\varepsilon \\ 0 \\ E_{\varepsilon p}(\tau_k - c_p) - E_{\varepsilon\varepsilon}c_\varepsilon \end{bmatrix} \quad (\text{A } 3)$$

Equations (A 3) reveal that incremental changes Δx_{1k} and Δx_{2k} are of the order of T , indicating a slow (quasi-constant) behaviour of x_{1k} and x_{2k} , while z_{1k} and z_{2k} consist of both slow and fast components. Decomposing the state variables into slow and fast components as

$$x_{ik} \approx \bar{x}_{ik}, \quad z_{ik} \approx \bar{z}_{ik} + \hat{z}_{ik}, \quad i = 1, 2, \quad \tau_k \approx \bar{\tau}_k + \hat{\tau}_k$$

and attributing the incremental changes Δz_{ik} to their fast components, the slow components of z_{ik} can be solved from the last two equations in (A 3) with $T = 0$ as

$$\begin{bmatrix} \bar{z}_{1k} \\ \bar{z}_{2k} \end{bmatrix} = \begin{bmatrix} 0 & -(K'_s)^{-1}E_{\varepsilon\varepsilon}^{-1}E_{\varepsilon p}B_{pp} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_{1k} \\ \bar{x}_{2k} \end{bmatrix} + \begin{bmatrix} (K'_s)^{-1}E_{\varepsilon\varepsilon}^{-1}E_{\varepsilon p}(\bar{\tau}_k - c_p) - (K'_s)^{-1}c_\varepsilon \\ 0 \end{bmatrix} \quad (\text{A } 4)$$

Replacing z_{1k} and z_{2k} in the first two equations in (A 3) with \bar{z}_{1k} and \bar{z}_{2k} above and using the relations in (3.4), we get

$$\begin{bmatrix} \Delta \bar{x}_{1k} \\ \Delta \bar{x}_{2k} \end{bmatrix} = T \begin{bmatrix} 0 & I \\ 0 & -A_{pp}^{-1} B_{pp} \end{bmatrix} \begin{bmatrix} \bar{x}_{1k} \\ \bar{x}_{2k} \end{bmatrix} + T \begin{bmatrix} 0 \\ A_{pp}^{-1}(\bar{v}_k - c_p) \end{bmatrix} \quad (\text{A } 5)$$

which, together with (A 4), describes the slow subsystem. The fast dynamics are readily obtained from the last two equations in (A 3), by setting $T = 0$, as

$$\begin{bmatrix} \Delta \hat{z}_{1k} \\ \Delta \hat{z}_{2k} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -E_{\varepsilon\varepsilon} K'_s & 0 \end{bmatrix} \begin{bmatrix} \hat{z}_{1k} \\ \hat{z}_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ E_{\varepsilon p} \hat{v}_k \end{bmatrix} \quad (\text{A } 6)$$

Finally, rewriting (A 4)–(A 6) in terms of the slow and fast components of the original variables p_k and ε_k , we get (3.2), (3.3) and (3.5).

REFERENCES

- BALESTRINO, A., DEMARIA, G. and SCIavicco, L., 1983, An adaptive model following control for robot manipulators. *Transactions of ASME Journal of Dynamic Systems and Measurement Control*, **105**, 143–151.
- BOOK, W. J., 1984, Recursive Lagrangian dynamics of flexible manipulator arms. *International Journal of Robotics Research*, **3**, 87–101; 1985, New concepts in lightweight arms. *Proceedings of the 2nd International Symposium on Robotics Research*, (Cambridge, Mass: MIT Press), pp 203–205.
- CANNON, R. H., JR, and SCHMITZ, E., 1984, Precise control of flexible manipulators. *Proceedings of the 1st International Symposium on Robotics Research*, (Cambridge, Mass: MIT Press), pp. 841–861.
- CRAIG, J. J., 1988, *Adaptive Control of Mechanical Manipulators*, (New York: Addison-Wesley).
- ÇETINKUT, S., SICILIANO, B., and BOOK, W. J., 1986, Symbolic modeling and dynamic analysis of flexible manipulators. *Proceedings of the IEEE International Conference Systems Man Cybernetics*, Atlanta, pp. 798–803.
- FREUND, E., 1982, Fast nonlinear control with arbitrary pole placement for industrial robots and manipulators. *International Journal of Robotics Research*, **1**, 65–78.
- GOODWIN, G. C., and SIN, K. S. 1984, *Adaptive Filtering Prediction and Control* (Englewood Cliffs, NJ: Prentice Hall).
- JUDD, P. R., and FALKENBURG, D. R., 1985, Dynamics of nonrigid articulated robot linkages. *IEEE Transactions on Automatic Control*, **30**, 499–302.
- KING, J. O., GOURISHANKAR, V. G., and RINK, R. E., 1990, Composite pseudolink end-point control of flexible manipulators. *IEEE Transactions on Systems Man Cybernetics*, **20**, 969–977.
- KOIVO, A. J., 1985, Self tuning manipulator control in Cartesian base coordinate system. *Transactions of ASME Journal of Dynamic Systems Measurement and Control*, **107**, 316–323; 1989, *Fundamentals for Control of Robot Manipulators*, (New York: Wiley).
- KOIVO, A. J., and GUO, T., 1983, Adaptive linear controller for robotic manipulator. *IEEE Transactions on Automatic Control*, **28**, 162–170.
- KOKOTOVIC, P. V., 1984, Applications of singular perturbation techniques to control problems. *SIAM Review*, **26**, 501–550.
- LEE, C. S. G., and LEE, B. H., 1984, Resolved motion adaptive control for mechanical manipulators. *Transactions of ASME Journal of Dynamic Systems Measurement and Control*, **106**, 134–142.
- LEE, C. S. G., LEE, B. H., and NIGAM, R., 1983, Development of the generalized d'Alembert equations of motion for mechanical manipulators. *Proceedings of the 22nd Conference on Decision and Control*, San Antonio, Texas, pp. 1205–1210.
- LUH, J. S., WALKER, M. W., and PAUL, R. P. C. 1980, Resolved acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, **25**, 468–474.

- NEMIR, D. C., KOIVO, A. J., and KASHYAP R. L., 1988, Pseudolinks and the self-tuning control of a nonrigid link mechanism. *IEEE Transactions on Systems, Man and Cybernetics*, **18**, 40-48.
- SICILIANO, B., and BOOK, W. J., 1988, A singular perturbation approach to control of lightweight flexible manipulators. *International Journal of Robotics Research*, **7**, 79-90.
- SLOTINE, J. J. E., 1985, The robust control of robot manipulators. *International Journal of Robotics Research*, **4**, 49-64.
- SOUISSI, R., and KOIVO, A. J., 1987, Design of self-tuning controllers for high-speed manipulators. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Alexandria, Virginia, pp. 897-902.
- TAKEGAKI, M., and ARIMOTO, S., 1981, A new feedback method for dynamic control of manipulation. *Transactions on the ASME Journal of Dynamics Systems Measurement and Control*, **102**, 119-125.
- UICKER, J. J., DENAVIT, J., and HARTENBERG, R. S., 1964, An iterative method for the displacement analysis of spatial mechanisms. *Journal of Applied Mechanics*, **26**, 309-314.