

5-1-1992

Robust Robot Control Based on Simplified Dynamic Model

Andrzej P. Bartoszewicz
Purdue University School of Electrical Engineering

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Bartoszewicz, Andrzej P., "Robust Robot Control Based on Simplified Dynamic Model" (1992). *ECE Technical Reports*. Paper 292.
<http://docs.lib.purdue.edu/ecetr/292>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

ROBUST ROBOT CONTROL BASED ON A SIMPLIFIED DYNAMIC MODEL

ANDRZEJ P. BARTOSZEWICZ

TR-EE 92-21
MAY 1992



SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Robust Robot Control Based on Simplified Dynamic Model

Andrzej P. Bartoszewicz[†]

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

In this paper, we propose a manipulator control based on a simplified robot dynamic model. The proposed controller is computationally efficient, and it assures asymptotic trajectory tracking. The simplified robotic model is obtained in a systematic way and is adequate for control purpose. The simplification algorithm is suitable either for general or trajectory specific motion. The algorithm simplifies only the position dependent elements of manipulator equations of motion. Therefore, if position tracking errors alone are small, the algorithm applied off-line for the desired trajectory, generates **simplified** model valid also for an actual trajectory. The algorithm assembles the structure of the simplified model, estimates its parameters and formulates bounds on the approximation errors. The simplified model and the bounds on approximation errors are then used to **construct** a control law based on Lyapunov stability theory. Finally, the control is extended to robust and adaptive cases and computer simulations for a three-link robot arm are presented to verify the performance of the proposed technique.

Key Words

Robot control, Robot dynamics model, Model simplification.

[†] Currently on leave from the Institute of Automatic Control, Department of Electrical Engineering, Technical University of Lodz, 18/22 Stefanowskiego St, 90-924 Lodz, Poland.

1. Introduction

The real-time control of a robot manipulator is a challenging task due to the presence of nonlinearity, coupling and interactions between various links. A great variety of control algorithms have already been proposed, ranging from the conventional PID control [23] to robust and adaptive algorithms based on Lyapunov theory [4,8,11,13,15,26,29,30,31]. Reviews of various adaptive control techniques can be found in [12, 28] and of robust control in [1].

Earliest work in robot adaptive control [10,18,20,21] applied model-referenced and self-tuning adaptive controls to manipulators based on the assumptions of time-invariant, decoupled dynamics, and local linearization and approximation. These assumptions are relaxed after some results were developed in the context of parameter estimation [2,3,17]. These parameter estimation schemes allow one to select a proper set of equivalent parameters such that the manipulator dynamics depends linearly on these parameters. Based on this linear parameterization property of the manipulator dynamics, more efficient and robust adaptive controls were developed.

Craig, et al. [8] proposed an adaptive version of the computed torque method for the control of manipulators with rigid links. They employed a parameter updating rule and rigorously proved the stability of the system in the sense of Lyapunov using the properties of positive real transfer function. However, their method requires the computation of the inverse of the inertial matrix $D(q)$. Hsu, et al. [13] proposed a new scheme very similar to their earlier scheme [8], which does not require the measurement of joint accelerations. Slotine and Li [29-31] proposed an adaptive control algorithm which consists of a proportional-plus-derivative (PD) feedback part and a full dynamics feed-forward compensation part, with an on-line parameter estimation scheme for the unknown manipulator payload parameters. Their adaptive control was computationally simpler because of the effective exploitation of the structure of the manipulator dynamics; that is, they made use of the fact that the matrix $[\dot{D}(q, \dot{q}) - 2C(q, \dot{q})]$ is skew symmetric. They used the variable structure system theory in the parameter adaptation for the robustness of their control algorithm. Although their control algorithm has the global asymptotical property, Johansson [15] pointed out that their V function was not a Lyapunov function because a

Lyapunov function should be a function of all the state vector components. Johansson then proposed algorithms for continuous-time direct adaptive control of robot manipulators by using the Lyapunov function and analyzed the stability property of his control algorithm. Finally, Ham and Lee [11] explicitly used the skew symmetric property of $[\dot{D}(\mathbf{q}, \mathbf{q}) - 2C(\mathbf{q}, \mathbf{q})]$ matrix to propose efficient, robust non-adaptive and adaptive controls similar to those proposed by Johansson.

The analysis and design of the above control algorithms require the development of efficient closed-form dynamic equations. A number of methods can be used to formulate manipulator arm dynamics, however, the two most often used methods are the Lagrangian formulation and the recursive Newton-Euler formulation. The Newton-Euler formulation focuses on the development of the dynamic model in an efficient recursive inverse dynamics form for generating the required generalized **forces/torques** for a given set of generalized coordinates, their time derivatives, and physical and geometric **parameters** of the robot arm [25]. One of the major drawbacks of these recursive dynamic equations is that they do not show the details of dynamic characteristics of robot manipulators in explicit terms for control system analysis, design, and synthesis. On the other hand, the Lagrangian formulation results in explicit state equations for manipulator dynamics, expressing the relationship between the generalized **forces/torques** and the generalized coordinates with the system parameters explicit in the equations [5]. Unfortunately, the generation of these state equations by hand (or even by a computer) for most industrial robots is a lengthy and tedious process. Furthermore, these lengthy state equations may exhibit too many **insignificant** details of dynamic characteristics of the manipulator, resulting in excessive computations in real time. Thus, obtaining simplified dynamic models that reveal the dominant dynamics without introducing significant errors into the dynamic model is essential for applying various control algorithms to control manipulators.

Paul [27], in his earlier experiments on a Stanford arm, discovered that the contributions from the Coriolis and centrifugal **terms** are relatively insignificant. This is true only when the manipulator is moving at slow speeds. Later on, **Bejczy** [5,7], experimenting an extended

Stanford arm, developed an approximated model for inertia and gravity terms based on the relative importance of inertial torques and gravity **torques/forces** as compared to the complete Lagrange-Euler (**L-E**) equations of motion. **Luh** and Lin [24] first presented an automatic computer procedure for generating simplified dynamic coefficients according to a threshold. They utilized the Newton-Euler equations of motion and compared all the terms in a computer for their significance to eliminate various **terms**. They then rearranged the remaining terms to form the equations of motion in symbolic form. **Desrochers** and Seaman [9] developed a projection algorithm based on the least-squares criterion which minimizes the L_2 norm error between the approximant and the nonlinear manipulator dynamic model. **Bejczy** and Lee [6], based on the differential transformation matrix technique, developed the model reduction method which utilizes a matrix numeric analysis to produce simplification for the **Coriolis** and centrifugal terms. Later, Lin and Chang [22] proposed a decomposition scheme which can avoid testing all the **terms** of a specific dynamic coefficient exhaustively for their significance. They expressed each dynamic coefficient as a linear combination of significant basis functions using a minimax curve fitting technique which provides a better approximate model.

The above model **simplification** schemes argue specifically from the viewpoint of obtaining the simplified dynamic model as compared to the complete **Euler-Lagrange** equations of motion. However, none of them analyzed the effects of approximate models on the system performance (e.g., **path/trajectory** tracking error) of the manipulator. The issue of simplifying the dynamic model while keeping desired performance specifications was addressed by Lee and Chang [19] and Jeon and Lee [14]. Lee and Chang [19] extended Lin and Chang's basis function concept and developed a multi-layered minimax decision scheme for automatic generation of a simplified manipulator dynamic model based on the desired manipulator system performance under a PD controller. Jeon and Lee [14] extended Lee and Chang's result to obtain a simplified dynamic model under nonlinear decoupled controllers. They based their modified scheme on steady-state **error** specifications expressed in both the Cartesian and joint variable space.

This paper presents the use of a simplified dynamic model in manipulator control algorithms to assure asymptotic trajectory tracking with boundedness of all internal signals. We first identify a robot simplified dynamic model, then determine its accuracy and propose a control law based on this simplified model. Since the model simplification is a complex and time consuming task, it is best to be performed off-line. Furthermore, to get the most effective simplified model, the simplification should be trajectory specific, that is, it should be done for a particular manipulator trajectory. Therefore it is reasonable to simplify manipulator model for the desired, prespecified trajectory and to set up simplification process so that the simplified model be optimal in a sense of some criterion for the desired trajectory and optimal or very close to optimal for the actual trajectory. Then, if the bounds on model approximation errors are determined, the simplified model together with the bounds can be used to construct control law assuring convergence of trajectory tracking errors.

In Section 2 of this paper we introduce the model simplification algorithm which is based on the Gram-Schmidt orthogonalization process. We propose both general motion and trajectory specific simplification procedures. Our algorithm identifies the simplified model, estimates its parameters, and establishes bounds on the model approximation errors. Only position dependent elements of the manipulator model are simplified and therefore, if position tracking errors are small, the simplification done for the desired trajectory is still valid for the actual trajectory. In fact Kelkar and Alberts [16] have shown that the assumption of small position tracking errors holds very well for PD-controlled manipulators under various payload conditions. In Section 3, we first briefly review control scheme introduced by Ham and Lee [11] and then, using their result, we propose controls based on the simplified model. Tracking error convergence is proven using Lyapunov method and then an extension to robust and adaptive case is proposed. Section 4 is devoted to computer simulations of the proposed approach and technique. Finally, Section 5 summarizes our findings and comprises some concluding remarks.

2. Model Simplification Procedure

As mentioned, a manipulator dynamics model is simplified along a desired trajectory prespecified for any time $t \in [0, t_f]$. Only position dependent elements are simplified. By this, the only assumption necessary about the actual trajectory in order for the model simplification to be valid for this actual trajectory is that the position tracking errors are small enough.

Based on the Lagrangian formulation and assuming rigid-body motion, the dynamic equations of an n -link manipulator, excluding gear friction and backlash, can be expressed as follows:

$$\sum_{j=1}^n d_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + g_i = \tau_i - \sum_{j=1}^n b_{ij} \dot{q}_j \quad (1)$$

for $i = 1, 2, \dots, n$, where q_i is the i th generalized coordinate, \dot{q}_i and \ddot{q}_i are, respectively, i th joint velocity and acceleration, d_{ij} is ij th moment of inertia, h_{ijk} is ijk th Coriolis and/or centrifugal force coefficient, g_i is the gravitational force acting on i th joint, τ_i is i th applied generalized force/torque and b_{ij} is an ij th frictional coefficient. Each d_{ij} , h_{ijk} , and g_i coefficient in Eq. (1) is a linear combination of some ϕ_l $l = 1, 2, \dots, p$ functions, which in turn are constant function and functions of solely q_i , $i = 1, 2, \dots, n$. Note that usually the number p of ϕ_l functions is big and most of the functions appear in more than one coefficients. For example, for the first three links of a PUMA robot arm $p = 17$, and the total number of terms in Eq. (1) is so big that the model is computationally inefficient. Moreover, some of the terms are insignificant and can be omitted. In the sequel, we propose a procedure to approximate each of d_{ij} , h_{ijk} and g_i coefficients by ζ term expression, where $\zeta < p$. Only these terms d_{ij} , h_{ijk} and g_i which originally have more than ζ terms will be approximated. Our algorithm uses the Gram-Schmidt orthogonalization procedure.

First, we define a r -element ordered set a of coefficients d_{ij} , h_{ijk} , and g_i which will be simplified. We include in the set each coefficient that:

- 1) is a linear combination of more than ζ different functions ϕ_l ,

- 2) has not already been included, which means that it is not identical (e.g., $d_{21} = d_{12}$) to any coefficient already included.

Let α_r , be the m th element of the set \mathbf{a} . Let us also denote by U_{mi} , $i = 1, 2, \dots, \gamma_m$, the variable (e.g., $\ddot{q}_1, q_1 \dot{q}_2$) which is multiplied by α_r . We next define weighting factors

$$\omega_m = \sum_{i=1}^{\gamma_m} \int_0^{t_f} U_{mi}^2 dt \quad (2)$$

which will be used in our algorithm.

With the above definitions, the following algorithm will be used to simplify the manipulator model and to estimate its parameters:

1. Let $j = 1$; $u_e = 0$ for $e = 1, 2, \dots, p$; $\phi_l^{(1)} = \phi_l$ for $l = 1, 2, \dots, p$; $\mathbf{a}_r^{(1)} \equiv \alpha_m$ for $m = 1, 2, \dots, r$.
2. For $l = 1, 2, \dots, p$; $l \neq u_e$; $e = 1, 2, \dots, j-1$, compute

$$Q_{ml}^{(j)} = \int_0^{t_f} (k_{ml}^{(j)} \phi_l)^2 dt = \frac{\left[\int_0^{t_f} \alpha_m^{(j)} \phi_l^{(j)} dt \right]^2}{\int_0^{t_f} [\phi_l^{(j)}]^2 dt} \quad (3)$$

where $Q_{ml}^{(j)}$ is a squared projection of $\alpha_m^{(j)}$ onto the direction of function $\phi_l^{(j)}$, and

$$k_{ml}^{(j)} = \frac{\int_0^{t_f} \alpha_m^{(j)} \phi_l^{(j)} dt}{\int_0^{t_f} [\phi_l^{(j)}]^2 dt} . \quad (4)$$

Define $R_l^{(j)}$ as the sum of weighted-squared projections $Q_{ml}^{(j)}$

$$R_l^{(j)} = \sum_{m=1}^r \omega_m Q_{ml}^{(j)} . \quad (5)$$

Determine $\max_I R_l^{(j)}$, let maximizing $l = u_j$ and include ϕ_{u_j} in the simplified model.

3. For $l = 1, 2, \dots, p$, $l \neq u_e$, $e = 1, 2, \dots, j$, compute

$$\phi_l^{(j+1)} = \phi_l^{(j)} - \frac{\phi_{u_j}^{(j)} \int_0^{t_f} \phi_l^{(j)} \phi_{u_j}^{(j)} dt}{\int_0^{t_f} \left[\phi_{u_j}^{(j)} \right]^2 dt} \quad (6)$$

and for $m = 1, 2, \dots, r$, compute

$$\alpha_m^{(j+1)} = \alpha_m^{(j)} - \frac{\phi_{u_j}^{(j)} \int_0^{t_f} \alpha_m^{(j)} \phi_{u_j}^{(j)} dt}{\int_0^{t_f} \left[\phi_{u_j}^{(j)} \right]^2 dt} \quad (7)$$

4. If $j < \zeta$, then let $j = j + 1$ and go to step 2; otherwise stop. The simplified model contains unchanged all d_{ij} , h_{ijk} and g_i coefficients that contain less than or equal to ζ number of ϕ_l functions, namely those coefficients which were not simplified and linear combinations of ζ selected $\phi_{u_1}, \phi_{u_2}, \dots, \phi_{u_\zeta}$ functions in place of all other d_{ij} , h_{ijk} and g_i coefficients. Only position dependent coefficients are modified while the structure of Eq. (1) remains unchanged. Each of the linear combinations, approximating α_m elements, can be expressed as:

$$\alpha_m - \alpha_m^{(\zeta+1)} = k_{mu_1}^{(1)} \mathcal{S}_{u_1}''' + k_{mu_2}^{(2)} \phi_{u_2}^{(2)} + \dots + k_{mu_\zeta}^{(\zeta)} \phi_{u_\zeta}^{(\zeta)} \quad (8)$$

and the upper bound on the m th coefficient approximation error is given by

$$(\delta\alpha_m)_{\max} = \max_{t \in [0, t_f]} \left| \alpha_m^{(\zeta+1)}(t) \right|. \quad (9)$$

The above algorithm simplifies position dependent elements of Eq. (1). estimates the model parameters (Eq. (4)) and gives clear bounds on how much the simplified coefficients d_{ij} , h_{ijk} and g_i may vary from their actual counterparts (Eq. (9)). We simplify the continuous-time model, thus avoiding problems connected with selecting sampling rate, providing precise bounds on parameter **errors** and eliminating estimation inaccuracy. If in equations (Eqs. (3), (4). (6), and

(7)) the integrals with respect to time are replaced by multiple integrals with respect to q_1, q_2, \dots, q_n over the whole manipulator workspace and if the integrals in Eq. (2) are evaluated over all admissible values of U_{mi} with respect to dU_{mi} rather than dt , then we obtain a simplification procedure suitable for all manipulator motions and not only for a particular trajectory.

Note that as the whole approximation procedure was performed for the desired trajectory the bounds in Eq. (9) are also **determined** under the assumption of zero **tracking** error. Now let us consider $\mathbf{a}_{m,i}$, and its simplified counterpart $\underline{\alpha}_m$. Taking into account imperfect trajectory tracking, the total upper bound on $\mathbf{a}_{m,i}$ approximation error due to both model simplification and approximate tracking becomes

$$(\Delta\alpha_m)_{\max} = \max_{\mathbf{q}_e \in E_q; t \in [0, t_f]} |\alpha_m^{(\zeta+1)}(\mathbf{q}_r) + [\alpha_m(\mathbf{q}_r + \mathbf{q}_e) - \alpha_m(\mathbf{q}_r)] - [\underline{\alpha}_m(\mathbf{q}_r + \mathbf{q}_e) - \underline{\alpha}_m(\mathbf{q}_r)]| \quad (10)$$

where $\mathbf{q}_r = \mathbf{q}_r(t)$ denotes a desired trajectory, and \mathbf{q}_e is the vector of position tracking errors. The maximum operation in Eq. (10) is performed over the whole motion time period and over all possible position errors. The set of **all** possible position errors E_q depends on the system initial conditions and will be determined in the next section for the proposed **control** method. If \mathbf{q}_e is small enough, then Eq. (10) can be reduced to $(\delta\alpha_m)_{\max}$.

Finally, we want to note that some practical **modifications** of the proposed simplification scheme are possible. If values of trigonometric functions can be predetermined and stored in a look-up table, then the number of ϕ_i functions included in the simplified model is not crucial. Then each $\mathbf{a}_{m,i}$ parameter can be simplified separately, reducing approximation errors and making simplification procedure computationally more efficient.

3. Manipulator control based on proposed simplified model

3.1. Preliminaries

We next briefly review the result of the paper [11]. The robot dynamics in Eq. (1) can be expressed as follows:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}(t) + \mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}}(t) + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}(t) - \mathbf{B}\dot{\mathbf{q}}(t) \quad (11)$$

where $\mathbf{q}(t) \in \mathbf{R}^n$ is the vector of generalized coordinates, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$ are, respectively, the vectors of joint velocity and acceleration, $\mathbf{D}(\mathbf{q}) \in \mathbf{R}^{n \times n}$ is a positive definite moment of inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbf{R}^n$ a vector function containing Coriolis and centrifugal forces, $\mathbf{G}(\mathbf{q}) \in \mathbf{R}^n$ is a vector function consisting of gravitational forces, $\boldsymbol{\tau}(t) \in \mathbf{R}^n$ is the vector function consisting of applied generalized forces/torques, and $\mathbf{B} \in \mathbf{R}^{n \times n}$ is a frictional coefficient matrix.

To derive a suitable control law, let us consider the following Lyapunov function candidate:

$$V(t) = \frac{1}{2} \begin{bmatrix} \dot{\mathbf{q}}_e^T & \mathbf{q}_e^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{D}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{qq} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_e \\ \mathbf{q}_e \end{bmatrix} \quad (12)$$

where $\mathbf{q}_e(t) = \mathbf{q}(t) - \mathbf{q}_r(t)$, $\mathbf{q}_r(t)$ is the desired trajectory, \mathbf{I} is an $n \times n$ identity matrix, \mathbf{P}_{qq} and \mathbf{P}_{12} are $n \times n$ symmetric positive definite matrices, and \mathbf{P}_{12} is chosen such that $\mathbf{P}_{12} = \mathbf{P}_{qq}^{-1} \boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ is an $n \times n$ symmetric positive definite matrix. Defining the vector $\mathbf{x}_e(t)$ and matrix \mathbf{S} as follows:

$$\mathbf{x}_e(t) = \begin{bmatrix} \dot{\mathbf{q}}_e \\ \mathbf{q}_e \end{bmatrix} \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} \mathbf{I} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (13)$$

$V(t)$ can be expressed as

$$V(t) = \frac{1}{2} (\mathbf{S}\mathbf{x}_e)^T \begin{bmatrix} \mathbf{D}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{qq} \end{bmatrix} (\mathbf{S}\mathbf{x}_e). \quad (14)$$

Differentiating V and using Eq. (11), we obtain

$$\begin{aligned} \dot{V} = (\mathbf{Sx}_e)^T & \begin{bmatrix} \dot{\mathbf{D}}(\mathbf{q})/2 + \mathbf{D}(\mathbf{q})\mathbf{P}_{12} & -\mathbf{D}(\mathbf{q})\mathbf{P}_{12}\mathbf{P}_{12} \\ \mathbf{P}_{qq} & -\mathbf{P}_{qq}\mathbf{P}_{12} \end{bmatrix} (\mathbf{Sx}_e) \\ & + (\mathbf{Sx}_e)^T \begin{bmatrix} \boldsymbol{\tau} - \mathbf{B}\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}_r \\ \mathbf{0} \end{bmatrix}. \end{aligned} \quad (15)$$

Ham and Lee [11] have shown that if \mathbf{P}_{qq} , $\boldsymbol{\Omega}$, \mathbf{W} are $n \times n$ symmetric positive definite matrices and $\mathbf{P}_{12} = \mathbf{P}_{qq}^{-1} \boldsymbol{\Omega}$, then the control law:

$$\begin{aligned} \boldsymbol{\tau} = \mathbf{D}(\mathbf{q}) (\ddot{\mathbf{q}}_r - \mathbf{P}_{12} \dot{\mathbf{q}}_e) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) (\dot{\mathbf{q}}_r - \mathbf{P}_{12} \mathbf{q}_e) + \mathbf{B}\dot{\mathbf{q}} \\ + \mathbf{G}(\mathbf{q}) - (\mathbf{W} + \mathbf{P}_{qq} \boldsymbol{\Omega}^{-1} \mathbf{P}_{qq}) (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) + \mathbf{P}_{qq} \mathbf{q}_e \end{aligned} \quad (16)$$

actually makes $V(t)$ a Lyapunov function with the time derivative

$$\dot{V}(t) = - \begin{bmatrix} \dot{\mathbf{q}}_e^T & \mathbf{q}_e^T \end{bmatrix} \mathbf{Q}_1 \begin{bmatrix} \dot{\mathbf{q}}_e \\ \mathbf{q}_e \end{bmatrix} \leq 0 \quad (17)$$

where

$$\mathbf{Q}_1 = \mathbf{Q}_1^T = \begin{bmatrix} \mathbf{I} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{W} + \mathbf{P}_{qq} \boldsymbol{\Omega}^{-1} \mathbf{P}_{qq} & -\mathbf{P}_{qq} \\ -\mathbf{P}_{qq} & \boldsymbol{\Omega} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{P}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \quad (18)$$

as they have shown that the matrix:

$$\begin{bmatrix} \mathbf{W} + \mathbf{P}_{qq} \boldsymbol{\Omega}^{-1} \mathbf{P}_{qq} & -\mathbf{P}_{qq} \\ -\mathbf{P}_{qq} & \boldsymbol{\Omega} \end{bmatrix} \quad (19)$$

is symmetric positive definite.

Let us now consider the adaptive control case. The control law in Eq. (16) can be expressed in the form which is linear in the system parameters:

$$\boldsymbol{\tau} = \boldsymbol{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \boldsymbol{\theta} + \boldsymbol{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \quad (20)$$

and as the actual system parameters $\boldsymbol{\theta}$ are unknown, the following control law, where $\hat{\boldsymbol{\theta}}$ is an estimate of $\boldsymbol{\theta}$, can be applied:

$$\begin{aligned} \hat{\boldsymbol{\tau}} &= \boldsymbol{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \hat{\boldsymbol{\theta}} + \boldsymbol{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \\ &= \hat{\mathbf{D}}(\mathbf{q}) (\ddot{\mathbf{q}}_r - \mathbf{P}_{12} \dot{\mathbf{q}}_e) + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) (\dot{\mathbf{q}}_r - \mathbf{P}_{12} \mathbf{q}_e) + \hat{\mathbf{B}} \dot{\mathbf{q}} \end{aligned} \quad (21)$$

$$+ \hat{\mathbf{G}}(\mathbf{q}) - (\mathbf{W} + \mathbf{P}_{qq} \mathbf{\Omega}^{-1} \mathbf{P}_{qq}) (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) + \mathbf{P}_{qq} \mathbf{q}_e .$$

Defining $\boldsymbol{\theta}_e$ as

$$\boldsymbol{\theta}_e = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \quad (22)$$

the above control law in Eq. (21) can be expressed as

$$\hat{\boldsymbol{\tau}} = \boldsymbol{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \boldsymbol{\theta} + \boldsymbol{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) + \boldsymbol{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \boldsymbol{\theta}_e . \quad (23)$$

Define a Lyapunov candidate function $V_1(t)$ as

$$V_1(t) = V(t) + \frac{1}{2} \boldsymbol{\theta}_e^T \mathbf{P}_{\theta\theta} \boldsymbol{\theta}_e \quad (24)$$

where $V(t)$ is the same as in Eq. (12), and $\mathbf{P}_{\theta\theta}$ is a positive definite matrix whose size is determined by the vector $\boldsymbol{\theta}_e$. Using the control law expressed in Eq. (21) and the following adaptation law for unknown parameters,

$$\dot{\boldsymbol{\theta}}_e = \dot{\hat{\boldsymbol{\theta}}} = -\mathbf{P}_{\theta\theta}^{-1} \boldsymbol{\Psi}^T (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) , \quad (25)$$

we obtain $\dot{V}_1(t) = \dot{V}(t)$. Hence Eqs. (21) and (25) determine the adaptive control scheme for the system.

3.2. Nonadaptive control based on simplified model

In Section 2, a simplified model of robot arm dynamics was developed, and bounds on d_{ij} , h_{ijk} , g_i coefficient **errors** were also established. Thus, we directly obtain

$$[\mathbf{D}]_{ij} - [\Delta\mathbf{D}]_{ij} < [\underline{\mathbf{D}}]_{ij} < [\mathbf{D}]_{ij} + [\Delta\mathbf{D}]_{ij} \quad (26)$$

$$[\mathbf{G}]_i - [\Delta\mathbf{G}]_i < [\underline{\mathbf{G}}]_i < [\mathbf{G}]_i + [\Delta\mathbf{G}]_i \quad (27)$$

where:

$\underline{\mathbf{D}}$ is a counterpart of \mathbf{D} obtained from the simplification algorithm,

$\Delta\mathbf{D}$ is a matrix of bounds on \mathbf{D} matrix approximation error,

$\underline{\mathbf{G}}$ is a counterpart of \mathbf{G} obtained **from** the simplification algorithm,

\mathbf{AG} is a vector of bounds on \mathbf{G} vector approximation error.

Each element of the matrix

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{11}(\mathbf{q}, \dot{\mathbf{q}}), & \cdots & c_{n1}(\mathbf{q}, \dot{\mathbf{q}}), \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ c_{1n}(\mathbf{q}, \dot{\mathbf{q}}) & \cdots & c_{nn}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \quad (28)$$

has the following form

$$c_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = h_{ij1}(\mathbf{q}) \dot{q}_1 + h_{ij2}(\mathbf{q}) \dot{q}_2 + \dots + h_{ijn}(\mathbf{q}) \dot{q}_n \quad (29)$$

From Section 2 we know the bounds (Δh_{ijk}) , $i, j, k = 1, 2, \dots, n$ on h_{ijk} approximation error. As \dot{q}_k is bounded for $k = 1, 2, \dots, n$ the bound on $c_{ij}(\mathbf{q}, \dot{\mathbf{q}})$ approximation error can be expressed as:

$$(\Delta c_{ij})_{\max} = \sum_{k=1}^n (\Delta h_{ijk})_{\max} \max(|\dot{q}_k|) \quad (30)$$

where $\max(|\dot{q}_k|)$ is a maximum speed of k th joint.

Similarly to Eq. (26) and (27), we can write

$$[\mathbf{C}]_{ij} - [\Delta \mathbf{C}]_{ij} < [\underline{\mathbf{C}}]_{ij} < [\mathbf{C}]_{ij} + [\Delta \mathbf{C}]_{ij} \quad (31)$$

where:

$\underline{\mathbf{C}}$ is a counterpart of \mathbf{C} obtained from the simplification procedure,

$\Delta \mathbf{C}$ is a matrix of bounds on \mathbf{C} matrix approximation error.

Since frictional **coefficients** are usually not known exactly, it is reasonable to assume that

$$[\mathbf{B}]_{ij} - [\Delta \mathbf{B}]_{ij} < [\underline{\mathbf{B}}]_{ij} < [\mathbf{B}]_{ij} + [\Delta \mathbf{B}]_{ij} \quad (32)$$

where:

$\underline{\mathbf{B}}$ is a **matrix** of **assumed** frictional coefficients,

$\Delta \mathbf{B}$ is a matrix of bounds on $(\mathbf{B} - \underline{\mathbf{B}})$.

Note that \mathbf{AB} , \mathbf{AC} , \mathbf{AD} , \mathbf{AG} are all known constant matrices with all nonnegative elements.

Now we introduce the following proposition:

Proposition 1:-

Let \mathbf{P}_{qq} , \mathbf{R} , $\mathbf{W} \in \mathbf{R}^{n \times n}$ be symmetric positive definite matrices such that all the elements of $\mathbf{P}_{12} = \mathbf{P}_{qq}^{-1} \mathbf{\Omega}$ are nonnegative. Let $\mathbf{\Delta C}_1$ be any symmetric matrix such that all successive principal minors of the determinant of $\mathbf{\Delta C}_1 + \mathbf{\delta C}_{sym}$, where $\mathbf{\delta C}_{sym}$ is a symmetrized $\mathbf{\delta C}$ matrix, be positive for all $\mathbf{\delta C}$ matrices such that

$$- [\mathbf{\Delta C}]_{ij} < [\mathbf{\delta C}]_{ij} < [\mathbf{\Delta C}]_{ij} \quad \text{for } i, j = 1, 2, \dots, n. \quad (33)$$

Then for any choice of \mathbf{P}_{qq} , $\mathbf{\Omega}$ and \mathbf{W} , the control law

$$\begin{aligned} \underline{\boldsymbol{\tau}} = & \underline{\mathbf{D}}(\mathbf{q}) (\ddot{\mathbf{q}}_r - \mathbf{P}_{12} \dot{\mathbf{q}}_e) + \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) (\dot{\mathbf{q}}_r - \mathbf{P}_{12} \mathbf{q}_e) + \underline{\mathbf{B}}\dot{\mathbf{q}} + \underline{\mathbf{G}}(\mathbf{q}) \\ & - (\mathbf{W} + \mathbf{P}_{qq} \mathbf{\Omega}^{-1} \mathbf{P}_{qq} + \mathbf{\Delta C}_1) (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) + \mathbf{P}_{qq} \mathbf{q}_e \\ & - \mathbf{SG}(\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) \left\{ (\mathbf{\Delta B} + \mathbf{\Delta C}) |\dot{\mathbf{q}}| + \mathbf{\Delta G} + \mathbf{\Delta D} [|\ddot{\mathbf{q}}_r| + \mathbf{P}_{12} |\dot{\mathbf{q}}_e|] \right\} \end{aligned} \quad (34)$$

makes the function $V(t)$ in Eq. (12) an actual Lyapunov function with the time derivative negative definite, where by definition,

$$\mathbf{SG}(\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) = \begin{bmatrix} \text{sgn}[\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e]_1 & 0 & \dots & 0 \\ 0 & \text{sgn}[\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e]_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \text{sgn}[\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e]_n \end{bmatrix} \quad (35)$$

is an $n \times n$ diagonal matrix and $\text{sgn}[\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e]_i$ is a signum function of i th element of $\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e$ vector.

Proof:

$V(t)$ is a positive definite quadratic form of \mathbf{x}_e . Substituting Eq.(34) into Eq.(15), we obtain:

$$\dot{V} = (\mathbf{Sx}_e)^T \begin{bmatrix} \dot{\mathbf{D}}/2 + \mathbf{D}\mathbf{P}_{12} & -\mathbf{D}\mathbf{P}_{12}\mathbf{P}_{12} \\ \mathbf{P}_{qq} & -\mathbf{P}_{qq}\mathbf{P}_{12} \end{bmatrix} \mathbf{Sx}_e + (\mathbf{Sx}_e)^T \begin{bmatrix} \underline{\boldsymbol{\tau}} - \mathbf{B}\dot{\mathbf{q}} - \mathbf{C}\dot{\mathbf{q}} - \mathbf{G} - \mathbf{D}\ddot{\mathbf{q}}_r \\ \mathbf{0} \end{bmatrix}$$

$$\begin{aligned}
 &= (\mathbf{Sx}_e)^T \begin{bmatrix} \dot{\mathbf{D}}/2 - \underline{\mathbf{C}} - \Delta\mathbf{C}_1 - (\mathbf{W} + \mathbf{P}_{qq}\mathbf{\Omega}^{-1}\mathbf{P}_{qq}) & , \mathbf{P}_{qq} \\ \mathbf{P}_{qq} & , -\mathbf{P}_{qq}\mathbf{P}_{12} \end{bmatrix} \mathbf{Sx}_e \\
 &+ (\mathbf{Sx}_e)^T \begin{bmatrix} -\underline{\mathbf{D}}\mathbf{P}_{12}\dot{\mathbf{q}}_e + \mathbf{D}\mathbf{P}_{12}\dot{\mathbf{q}}_e - \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \Delta\mathbf{D}\mathbf{P}_{12} | \dot{\mathbf{q}}_e | \\ \mathbf{0} \end{bmatrix} \\
 &+ (\mathbf{Sx}_e)^T \begin{bmatrix} (\underline{\mathbf{D}} - \mathbf{D})\ddot{\mathbf{q}}_r + (\underline{\mathbf{C}} - \mathbf{C})\dot{\mathbf{q}} + (\underline{\mathbf{B}} - \mathbf{B})\dot{\mathbf{q}} + \underline{\mathbf{G}} - \mathbf{G} - \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) [(\Delta\mathbf{B} + \Delta\mathbf{C}) | \dot{\mathbf{q}} | + \Delta\mathbf{G} + \Delta\mathbf{D} | \ddot{\mathbf{q}}_r | \\ \mathbf{0} \end{bmatrix}
 \end{aligned} \tag{36}$$

Denoting

$$\underline{\mathbf{B}} - \mathbf{B} \triangleq \delta\mathbf{B}$$

$$\underline{\mathbf{C}}(t) - \mathbf{C}(t) \triangleq \delta\mathbf{C}(t)$$

$$\underline{\mathbf{D}}(t) - \mathbf{D}(t) \triangleq \delta\mathbf{D}(t)$$

$$\underline{\mathbf{G}}(t) - \mathbf{G}(t) \triangleq \delta\mathbf{G}(t)$$

and dropping argument t for simplicity, we obtain

$$\begin{aligned}
 \dot{V} &= (\mathbf{Sx}_e)^T \begin{bmatrix} \dot{\mathbf{D}}/2 - \mathbf{C} - \delta\mathbf{C} - \Delta\mathbf{C}_1 - (\mathbf{W} + \mathbf{P}_{qq}\mathbf{\Omega}^{-1}\mathbf{P}_{qq}) & , \mathbf{P}_{qq} \\ \mathbf{P}_{qq} & , -\mathbf{P}_{qq}\mathbf{P}_{12} \end{bmatrix} \mathbf{Sx}_e \\
 &+ (\mathbf{Sx}_e)^T \begin{bmatrix} \delta\mathbf{D}\mathbf{P}_{12}\dot{\mathbf{q}}_e - \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \Delta\mathbf{D}\mathbf{P}_{12} | \dot{\mathbf{q}}_e | \\ \mathbf{0} \end{bmatrix} \\
 &+ (\mathbf{Sx}_e)^T \begin{bmatrix} \delta\mathbf{D}\ddot{\mathbf{q}}_r + \delta\mathbf{C}\dot{\mathbf{q}} + \delta\mathbf{B}\dot{\mathbf{q}} + \delta\mathbf{G} - \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) [(\Delta\mathbf{B} + \Delta\mathbf{C}) | \dot{\mathbf{q}} | + \Delta\mathbf{G} + \Delta\mathbf{D} | \ddot{\mathbf{q}}_r | \\ \mathbf{0} \end{bmatrix}.
 \end{aligned} \tag{37}$$

Taking into account that matrix $\dot{\mathbf{D}}/2 - \mathbf{C}$ is skew symmetric, and after some calculations, we get

$$\begin{aligned}
 \dot{V} &= (\mathbf{Sx}_e)^T \begin{bmatrix} -(\mathbf{W} + \mathbf{P}_{qq}\mathbf{\Omega}^{-1}\mathbf{P}_{qq}) & , \mathbf{P}_{qq} \\ \mathbf{P}_{qq} & , -\mathbf{P}_{qq}\mathbf{P}_{12} \end{bmatrix} \mathbf{Sx}_e + \\
 &- (\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e)^T (\delta\mathbf{C} + \Delta\mathbf{C}_1) (\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \\
 &+ (\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e)^T \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) [\mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \delta\mathbf{D}\mathbf{P}_{12}\dot{\mathbf{q}}_e - \Delta\mathbf{D}\mathbf{P}_{12} | \dot{\mathbf{q}}_e |] \\
 &+ (\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e)^T \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \\
 &\left\{ \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) [\delta\mathbf{D}\ddot{\mathbf{q}}_r + (\delta\mathbf{B} + \delta\mathbf{C})\dot{\mathbf{q}} + \delta\mathbf{G}] - [\Delta\mathbf{D} | \ddot{\mathbf{q}}_r | + (\Delta\mathbf{B} + \Delta\mathbf{C}) | \dot{\mathbf{q}} | + \Delta\mathbf{G}] \right\}
 \end{aligned} \tag{38}$$

$$\begin{aligned}
&= (\mathbf{Sx}_e)^T \left[\begin{array}{cc} -(\mathbf{W} + \mathbf{P}_{qq} \boldsymbol{\Omega}^{-1} \mathbf{P}_{qq}) & , \mathbf{P}_{qq} \\ \mathbf{P}_{qq} & , -\mathbf{P}_{qq} \mathbf{P}_{12} \end{array} \right] \mathbf{Sx}_e \\
&\quad - (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e)^T (\boldsymbol{\delta C} + \Delta \mathbf{C}_1) (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) \\
&\quad + (|\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e|)^T [\boldsymbol{\delta D}^* \mathbf{P}_{12} \dot{\mathbf{q}}_e - \Delta \mathbf{D} \mathbf{P}_{12} |\dot{\mathbf{q}}_e|] \\
&\quad + (|\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e|)^T \left\{ \boldsymbol{\delta D}^* \ddot{\mathbf{q}}_e + (\boldsymbol{\delta C}^* + \boldsymbol{\delta B}^*) \dot{\mathbf{q}}_e + \boldsymbol{\delta G}^* - [\Delta \mathbf{D} |\ddot{\mathbf{q}}_e| + (\Delta \mathbf{B} + \Delta \mathbf{C}) |\dot{\mathbf{q}}_e| + \Delta \mathbf{G}] \right\}.
\end{aligned} \tag{39}$$

Notice that matrices $\boldsymbol{\delta D}^*$, $\boldsymbol{\delta C}^*$, $\boldsymbol{\delta B}^*$, and $\boldsymbol{\delta G}^*$ are essentially the same as $\boldsymbol{\delta D}$, $\boldsymbol{\delta C}$, $\boldsymbol{\delta B}$, $\boldsymbol{\delta G}$, but the only difference is that **all** the elements in some rows have opposite signs.

First term on the right hand side of Eq. (39) is negative definite as proven by Ham and Lee [11]. Since the Sylvester's criterion for positive **definiteness** of the quadratic form, $(\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e)^T (\boldsymbol{\delta C} + \Delta \mathbf{C}_1) (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e)$ is satisfied by the assumptions of proposition 1, the second term of Eq. (39) is also negative definite. **Taking** into account the relation in Eq. (26) and the fact that **all** the elements of \mathbf{P}_{12} are nonnegative, the third **term** in **V** expansion becomes **nonpositive**. Finally, relations in **Eqs.(26), (27), (31), and (32)** assure that the last term is nonpositive and therefore the V is negative definite.

One possible and computationally efficient choice of $\max \Delta \mathbf{C}_1$ is $\Delta \mathbf{C}_1 = k \mathbf{I}$ where I is an identity matrix, with k big enough; for example, $k > \sum_{i,j=1}^n [\Delta \mathbf{C}]_{ij}$. Notice that the term $-\Delta \mathbf{C}_1 (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e)$ in the control law in Eq. (34) could be replaced by $-\mathbf{S} \mathbf{G} (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) \mathbf{A} \mathbf{C} |\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e|$. Then the second term of V function would be replaced by

$$- [(\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e)^T \boldsymbol{\delta C} (\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e) + |\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e| \Delta \mathbf{C} |\dot{\mathbf{q}}_e + \mathbf{P}_{12} \mathbf{q}_e|] \tag{40}$$

which is also nonpositive by the assumption in Eq. (31), and therefore V would remain negative definite.

As AB, AC, $\Delta \mathbf{C}_1$, AD, AG are constant matrices, a priori known and matrices $\underline{\mathbf{D}}$, $\underline{\mathbf{C}}$, $\underline{\mathbf{G}}$ can, with a proper selection of number of functions in an approximate model, be essentially simpler than their exact counterparts, the **control** law in Eq. (34) is clearly computationally more

efficient than the control law in Eq. (16), while it still assures **tracking** error convergence.

Finally, we address the issue of **determining** the set \mathcal{E}_q of possible tracking errors that should be considered in Eq. (10). Suppose that initial position and velocity error vectors are $|\mathbf{q}_e(0)| \leq \mathbf{q}_{emax}(0)$ and $|\dot{\mathbf{q}}_e(0)| \leq \dot{\mathbf{q}}_{emax}(0)$, respectively. Then $\mathbf{x}_e(0) = [\dot{\mathbf{q}}_e^T(0), \mathbf{q}_e^T(0)]^T$ and maximum value of $V(t)$ can be expressed as

$$V_{\max} = \frac{1}{2} [\mathbf{S}\mathbf{x}_e(0)]^T \begin{bmatrix} \mathbf{D}[\mathbf{q}(0)] & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{qq} \end{bmatrix} [\mathbf{S}\mathbf{x}_e(0)] \quad (41)$$

Consequently, the set of all possible position **errors** is determined by the following inequality

$$V_{\max} \geq \frac{1}{2} \mathbf{q}_e^T \mathbf{P}_{qq} \mathbf{q}_e . \quad (42)$$

In order to keep the position errors and consequently simplified model parameter errors reasonably small, matrix \mathbf{P}_{qq} should be selected so that its minimum eigenvalue is big enough to make

$$\frac{1}{2} \mathbf{q}_{emax}^T(0) \mathbf{P}_{qq} \mathbf{q}_{emax}(0) \gg V_{\max} - \frac{1}{2} \mathbf{q}_e^T(0) \mathbf{P}_{qq} \mathbf{q}_e(0) \quad (43)$$

for any vector $\mathbf{x}_e(0)$.

3.3. Extension to simplified-model-based robust and adaptive control

Suppose that due to payload changes or degradation of components in a robot arm, simplified dynamics model parameters are not known exactly. This is equivalent to the assumption that only simplified model structure is known while its parameters must be estimated and bounds on the parameter errors have to be determined. The new bounds on parameter errors can be expressed as follows:

$$\mathbf{A}\mathbf{B}' = \mathbf{A}\mathbf{B}, \quad \mathbf{A}\mathbf{C}' = \max_{\mathbf{A}}(\Delta\mathbf{C}), \quad \mathbf{A}\mathbf{D}' = \max_{\mathbf{A}}(\Delta\mathbf{D}), \quad \mathbf{A}\mathbf{G}' = \max_{\mathbf{A}}(\Delta\mathbf{G}), \quad (44)$$

where maximum is determined over the whole admissible set of payloads and manipulator parameters.

Now we introduce two modifications of the proposed control algorithm to make it robust or adaptive to manipulator and payload dynamics uncertainties. In the robust **control** algorithm, the system model is simplified and its parameters **are** estimated for the desired trajectory and nominal manipulator with payload dynamics, that is, approximation and estimation are performed exactly as described in the previous section. After the model structure has been determined and its parameters have been estimated, we search for maximum error of each simplified model element α_m over the considered motion time $[0, t_f]$ and all admissible **manipulator** and payload dynamics conditions. The **determined** maximum errors are then used as new bounds in Eq. (44), while the control law in Eq. (34) remains unchanged. The main advantage of the robust control is that no additional calculations **are** performed on-line. Therefore the computational complexity of the control scheme is not increased, while various payload and manipulator dynamics conditions are accounted for. However, as bounds on model element **errors** are increased, chattering in control signals becomes more significant. To diminish this **undesirable** effect, adaptive control can be applied. Similarly to the robust control, first simplified model structure is assembled for the adaptive algorithm. However, to establish bounds on model element **errors**, model parameters would be estimated separately for all, rather than only for nominal, manipulator and load conditions. The bounds should be determined as a maximum difference between actual value of each model element and its simplified value with parameters estimated for each actual payload and manipulator dynamics conditions. The maximum operation would be again performed over the considered motion time $[0, t_f]$ and all admissible dynamics conditions. As the above bounds **determination task** formulated as a maximum search process would require repetitive estimation and thus would be very inefficient, we propose another approach. Suppose that i th inertial parameter of a robot arm, that is, i th element of vector δ in Eq. (20) can vary by no more than a factor k_i ($k_i \geq 0$) from its nominal value. Taking into account that each element α_m of a robot arm model is linear in the elements of δ vector, each bound in Eq. (44) cannot differ from its counterpart calculated for nominal dynamics of the system by more than factor $k_m = 1 + \max_i |1 - k_i|$, where maximum operation is **performed** over these numbers i , where i

denotes $k_i = th$ factors multiplied by a function ϕ_i included in a particular exact α_m element and not included in the simplified model. Now, when the simplified model structure is assembled and bounds on the model parameter **errors** are determined, we propose a robot arm adaptive control based on the simplified model. The control law in Eq. (34) can be expressed in the following, linear in parameters, form

$$\begin{aligned}
 \underline{\tau} &= \underline{\mathbf{D}}(\mathbf{q})(\ddot{\mathbf{q}}_r - \mathbf{P}_{12}\dot{\mathbf{q}}_e) + \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_r - \mathbf{P}_{12}\mathbf{q}_e) + \underline{\mathbf{B}}\dot{\mathbf{q}} + \underline{\mathbf{G}}(\mathbf{q}) \\
 &\quad - (\mathbf{W} + \mathbf{P}_{qq}\mathbf{\Omega}^{-1}\mathbf{P}_{qq} + \Delta\mathbf{C}_1)(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) + \mathbf{P}_{qq}\mathbf{q}_e \\
 &\quad - \mathbf{S}\mathbf{G}(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \left\{ (\Delta\mathbf{B} + \Delta\mathbf{C})|\dot{\mathbf{q}}| + \Delta\mathbf{G} + \Delta\mathbf{D}[|\ddot{\mathbf{q}}_r| + \mathbf{P}_{12}|\dot{\mathbf{q}}_e|] \right\} \\
 &= \underline{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r)\underline{\theta} + \underline{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r)
 \end{aligned} \tag{45}$$

where $\underline{\theta}$ is a vector of simplified model parameters.

Similar to Section 3.1, we replace $\underline{\theta}$ with its estimate $\hat{\underline{\theta}}$

$$\underline{\hat{\tau}} = \underline{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r)\hat{\underline{\theta}} + \underline{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) \tag{46}$$

and defining $\underline{\theta}_e = \hat{\underline{\theta}} - \underline{\theta}$, we express Eq. (46) as

$$\underline{\hat{\tau}} = \underline{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r)\underline{\theta} + \underline{\Psi}_0(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r) + \underline{\Psi}(\ddot{\mathbf{q}}_r, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \mathbf{q}, \mathbf{q}_r)\underline{\theta}_e. \tag{47}$$

Defining a Lyapunov function candidate, we have

$$\underline{V}_1(t) = V(t) + \frac{1}{2} \underline{\theta}_e^T \underline{\mathbf{P}}_{\theta\theta} \underline{\theta}_e \tag{48}$$

where $V(t)$ is the same as in Eq. (12) and $\underline{\mathbf{P}}_{\theta\theta}$ is a positive definite matrix whose size is determined by the vector $\underline{\mathbf{g}}_e$. Again similarly to Section 3.1, if we use the control law in Eq. (46) and the following adaptation scheme

$$\dot{\underline{\theta}}_e = \dot{\hat{\underline{\theta}}} = -\underline{\mathbf{P}}_{\theta\theta}^{-1} \underline{\Psi}^T(\dot{\mathbf{q}}_e + \mathbf{P}_{12}\mathbf{q}_e) \tag{49}$$

we will obtain $\dot{\underline{V}}_1(r) = \dot{V}(t)$. Hence Eqs. (46) and (49) constitute our adaptive control scheme, for the system in Eq. (1), based on simplified model.

In the adaptive control case, the set E_q of possible tracking errors that should be considered in Eq. (10) can be determined similarly to the non-adaptive case with the only difference that Eq. (41) must contain an **extra** term $0.5\theta_{emax}^T P_{\theta\theta} \theta_{emax}$, where θ_{emax} is the maximum of θ_e .

Each element of matrix $\underline{\Psi}$ appearing in Eqs. (46) and (49) can be expressed as

$$[\underline{\Psi}]_{ij} = \phi_k f_k \quad i=1, \dots, n; \quad j=1, \dots, dim\theta \quad (50)$$

that is, as one product of the function ϕ_k in the simplified model and some function of the desired trajectory and measured values of position and velocity. On the other hand, each element of matrix Ψ in Eqs. (21) and (25) is a sum of products of the form:

$$[\Psi]_{ij} = \sum_{k=1}^p \phi_k f_k \quad i=1, \dots, n \quad j=1, \dots, dim\theta \quad (51)$$

Not only the **form** of $[\Psi]_{ij}$ is simpler but the computation of $[\underline{\Psi}]_{ij}$ requires only a limited number of functions ϕ_k . The ϕ_k functions usually contain trigonometric functions, the evaluation of which is a time consuming process. Thus, if the number of functions in the simplified model is reasonable, the simplification **makes** the model essentially more efficient. As the dimension of θ vector depends explicitly on the number of functions included in the simplified model, it suggests that the exact model should be approximated with a **predetermined** number of functions rather than simplified to some arbitrary accuracy.

4. Computer simulations

In order to verify the proposed control algorithms, computer simulations for a 3-degree-of-freedom (**3-DOF**) manipulator arm shown in Figure 1 were performed. For the simulations, the lengths of the manipulator links were assumed to be: $l_1=0.5m$, $l_2=0.5m$ and $l_3=0.6m$. All nonzero elements of the manipulator **link pseudo-inertia** matrices are shown in Table 1. In all simulation examples, we assumed the initial position error of each joint to be 0.2 radians and we

set matrices W , P_{qq} , Ω , P_{12} , and $P_{\theta\theta}$ to diagonal matrices 401, 41, I, 0.251 and I, respectively. The **Runge-Kutta** fourth-order method was applied for integrating the manipulator dynamics and a control sampling period of 10ms was used.

First we considered nominal manipulator and payload dynamics. We simulated the system based on the simplified manipulator model and we compared it with the system controlled by the algorithm based on the exact model. Figures 2, 3 and 4 show the simulation results. The dashed line on Figure 2 shows the desired trajectory while the continuous line depicts actual trajectory of the manipulator **arm** controlled by the algorithm based on the simplified model. This actual trajectory was compared to one obtained in case of control based on the exact model, however the differences were so insignificant that the two trajectories cannot be distinguished at the picture. Figures 3 and 4 show respectively manipulator joint torques and trajectory tracking errors for both types of control. It can be seen on Figure 3 that chattering in control signals due to the application of simplified model is relatively small, while Figure 4 shows slightly better tracking performance of the simplified model based control. On each of the Figures 4a-4c, the continuous line depicts a joint **tracking** error for the control based on the simplified model and the dashed line shows the same error in case of control using exact model. Better tracking performance of the simplified model based control is a **consequence** of the fact that our simplified control scheme is constructed so that the Lyapunov derivative for this scheme is always less than the same derivative for the exact model based control. Therefore, for the same initial joint position and velocity **errors** the Lyapunov function itself decreases faster in case of simplified model based control than in case of the exact model based control. It can also be seen both at Figures 2 and 4 that in practical applications tracking error convergence time can be assumed as 1.5 seconds.

As joint dynamics of manipulator and payload is usually unknown we also simulated robust and adaptive control schemes based on the simplified model. We assumed that the manipulator can carry an unknown payload placed at the end of the third link and that the total mass of the third link and the payload can vary between 0.8 and **1.2** of its nominal value. Figure 5 shows

manipulator joint torques for robust and adaptive control schemes, while Figure 6 shows manipulator tracking errors for both types of control. Tracking errors converge faster in case of robust control, however as we expected chattering in the robust scheme is significant. On the contrary in case of adaptive control chattering is essentially smaller but error convergence is slowed down.

5. Conclusions

In this paper, we used robot arm simplified model to derive control laws that would assure asymptotic trajectory tracking for the arm. First we proposed an algorithm to identify arm simplified model and to evaluate the model accuracy. Only position dependent elements of the exact model are approximated and the simplified model obtained for the desired trajectory can also be applied for the actual trajectory. Therefore the time consuming simplification process is performed off-line, without increasing an on-line computational complexity. Non trajectory-specific simplification option is also presented.

After we had identified the arm simplified model, we used Lyapunov stability theory to construct control law assuring asymptotic trajectory tracking. The proposed control law is computationally efficient, assures slightly better tracking performance than the exact model based control and causes relatively small chattering in control signals. Furthermore the discontinuous action of (34) can be easily approximated by the so-called boundary layer controller which is continuous and thus eliminates chattering at all. Finally we extended our result to robust and adaptive control and we simulated all presented algorithms for the 3-DOF robot arm.

6. Acknowledgment

The author gratefully acknowledges constructive help of Professor C.S.George Lee.

7. References

- [1] C. **Abdallah** et. al., "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24-30, Feb 1991.
- [2] C. H. An, C. G. Atkeson, and J. M. Hollerbach, "Estimation of inertial parameters of rigid body links of manipulators," *Proc. of IEEE Conf. on Decision and Control*, **Dec**, 1985, pp. 990-995.
- [3] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Rigid body load identification for manipulators," *Proc. of IEEE Conf. on Decision and Control*, **Dec**, 1985, pp. 996-1002.
- [4] M. **Amestegui**, R. **Ortega**, and J. M. **Ibarra** "Adaptive linearizing-decoupling robot control: a comparative study of different parametrizations," *Proc. 5th Yale Workshop on Applications of Adaptive Systems Theory*, New Haven, Conn., 1987.
- [5] A. K. **Bejczy**, "Robot arm dynamics and control," *JPL Technical Memorandum 33-609*, Pasadena, CA. Feb 1974.
- [6] A. K. **Bejczy** and S. Lee, "Robot arm dynamic model reduction for control," *Proc. of the 22nd IEEE Conf. on Decision and Control*, pp. 1466-1474, San Antonio. TX , **Dec**. 1983.
- [7] A. K. **Bejczy** and R. P. Paul, "Simplified robot arm dynamics for control," *Proc. of the 20th IEEE Conf. on Decision and Control*, San **Diego**, CA, pp. 261-262, **Dec**. 1981.
- [8] J. J. Craig, P. Hsu, and S. S. **Sastry**, "Adaptive control of mechanical manipulators," *Proc. of 1986 IEEE Int'l Conf. on Robotics and Automation*, San Francisco, CA, pp. 190-195, April 1986.
- [9] A. **A. Desrocher** and C. M. Seaman, "A projection method for simplifying robot manipulator models," *Proc. of 1986 IEEE Int'l Conf. on Robotics and Automation*, San Francisco, CA., pp. **504-509**, April 1986.
- [10] S. **Dubowsky** and D. T. **DesForges**, "The application of model referenced adaptive control to robotic manipulators," *Trans. of ASME, J. of Dynamic Systems, Measurement and Control*, vol. 101, pp. 193-200, 1979.

- [11] W. Ham and C. S. G. Lee, "Model reference robot adaptive control based on linear parametrization model," *Int'l J. of Robotics and Automation* (to appear).
- [12] T. C. Hsia, "Adaptive control of robot manipulators - a review," *Proc. of 1986 IEEE Int'l Conf. on Decision and Control*, San Francisco, CA, pp. 183-189, 1986.
- [13] P. Hsu, M. Bodson, S. Sastry, and B. Paden, "Adaptive identification and control for manipulators without using joint accelerations," *Proc. of 1987 IEEE Int'l Conf. on Robotics and Automation*, Raleigh, N.C., pp. 1210-1215, 1987.
- [14] J. W. Jeon and C. S. G. Lee, "Simplification of manipulator dynamic model for nonlinear decoupled control," *Proc. of the 27th Conf. on Decision and Control*, Austin, Texas, Dec. 1988.
- [15] R. Johansson, "Adaptive control of robot manipulator motion," *IEEE Trans. Robotics and Automation*, vol. 6, no. 4, Aug, 1990, pp. 483-490.
- [16] A. G. Kelkar and T. E. Alberts, "Computing maximum tracking error due to payload dynamics," *Proc. of 1990 IEEE Int'l Conf. on Robotics and Automation*, pp. 851-855, 1990.
- [17] P. K. Khosla and T. Kanade, "Parameter identification of robot dynamics," *Proc. of IEEE Conf. on Decision and Control*, Dec, 1985, pp. 1754-1760.
- [18] A. J. Koivo and T. H. Guo, "Adaptive linear controller for robotic manipulators," *IEEE Trans. on Automatic Control*, vol. AC-28, no. 1, pp. 162-171, 1983.
- [19] C. S. G. Lee and P. R. Chang, "Simplification of robot dynamic model based on system performance," *Proc. of 26th IEEE Conf. on Decision and Control*, Los Angeles, CA, pp. 1017-1023, Dec. 1987.
- [20] C. S. G. Lee and M. J. Chung, "An adaptive control strategy for mechanical manipulators," *IEEE Trans. on Automatic Control*, vol. AC-29, no. 9, pp. 837-840, Sept. 1984.
- [21] C. S. G. Lee and B. H. Lee, "Resolved motion adaptive control for mechanical manipulators," *Trans. of ASME, J. Dynamic Syst., Meas., Contr.*, vol. 106, no. 2, pp. 134-142, June

1984.

- [22] C. S. Lin and P. R. Chang, "Automatic dynamics simplification for robot manipulators," *Proc. of the 23rd Conf. on Decision and Control*, pp. 752-759, Las Vegas, NV, Dec. 1985.
- [23] J. Y. S. Luh, "Conventional controller design for industrial robots - a tutorial," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, May/June 1983, pp. 298-316.
- [24] J. Y. S. Luh and C. S. Lin, "Automatic generation of dynamic equations for mechanical manipulators," *Proc. 1981 Joint Auto. Control Conf.*, Charlottesville, VA, June 1981.
- [25] J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "On-line computational scheme for mechanical manipulator," *Trans. of ASME, J. of Dynamic Systems, Measurement and Control*, vol. 102, pp. 69-76, June 1980.
- [26] R. Middleton and G. Goodwin, "Adaptive computed torque control for rigid link manipulators," *Proc. of IEEE Conf. on Decision and Control*, Athens, Greece, 1986.
- [27] R. P. Paul, "Modeling, Trajectory Calculation, and Servoing of Computer Controlled Arm," Stanford University, A. I. Lab., AIM 177, Nov. 1972.
- [28] R. Ortega and M. W. Spong, "Adaptive motion control for rigid robots: a tutorial," *Automatica*, Dec. 1989.
- [29] J. J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int'l J. of Robotics Research*, vol. 6 no. 3, Fall, 1987, pp. 49-59.
- [30] J. J. E. Slotine and W. Li, "Adaptive strategies in constrained manipulation," *Proc. of 1987 IEEE Int'l Conf. on Robotics and Automation*, Raleigh, N.C., pp. 595-601, 1987.
- [31] J. J. E. Slotine and W. Li, "Adaptive manipulator control : a case study," *Proc. of 1987 IEEE Int'l Conf. on Robotics and Automation*, Raleigh, N.C., pp. 1392-1400, 1987.

Table 1. Elements of link pseudo-inertia matrices

link	mass m (kg)	first order moments			second order moments		
		M_x (kgm)	M_y (kgm)	M_z (kgm)	I_{xx} (gm^2)	I_{yy} (gm^2)	I_{zz} (gm^2)
1	5.5	0	-0.625	0	0.391	208	0.391
2	4	-0.375	0	0	125	0.115	0.115
3	3.5	-0.3	0	0	120	0.0563	0.0563

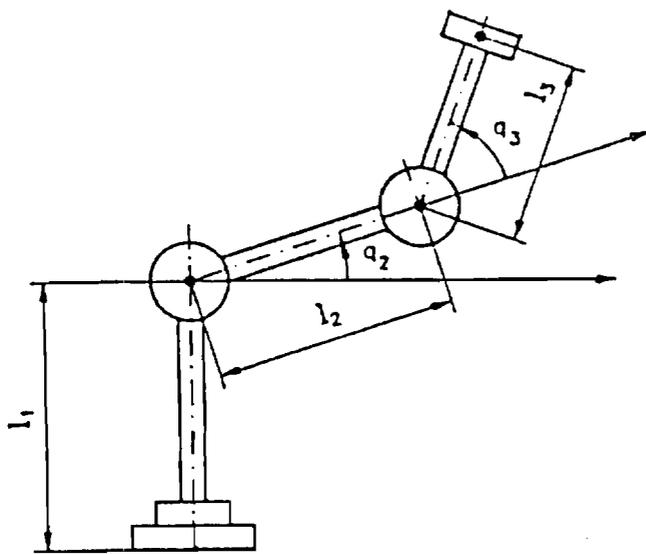


Figure 1 An example of 3-DOF robot arm

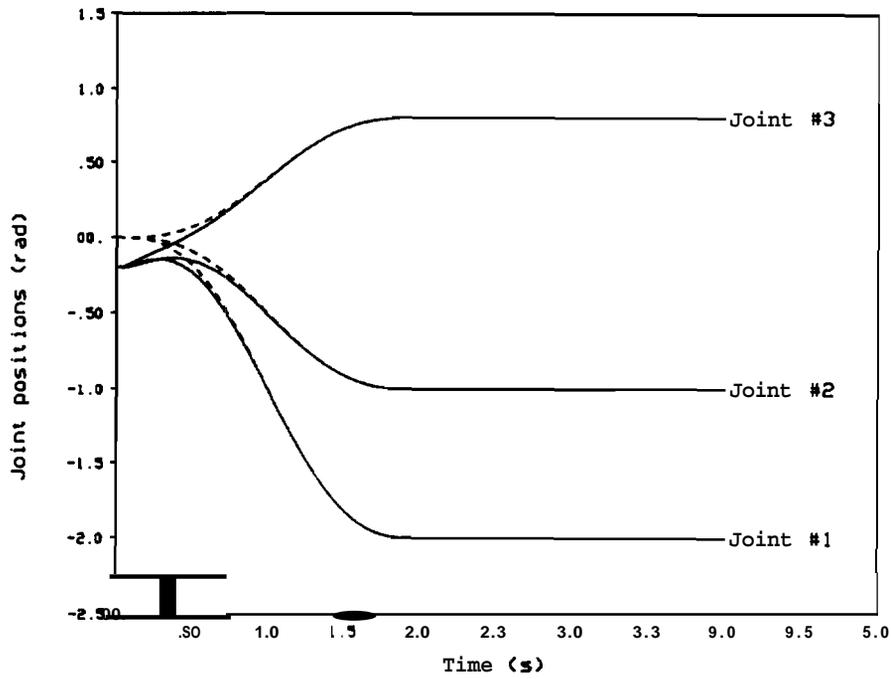


fig.2 Desired and actual manipulator trajectory

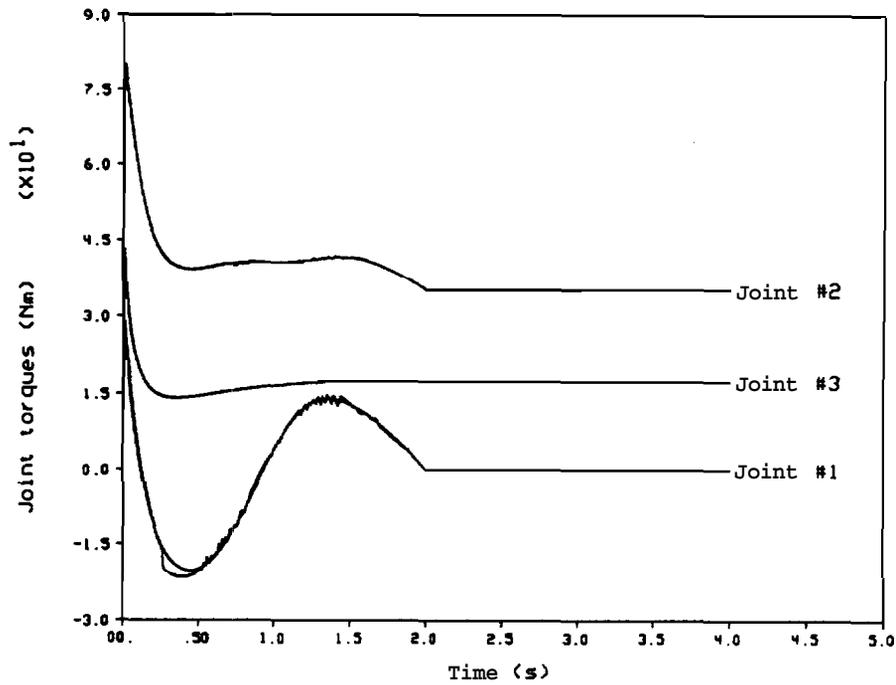


Fig.3 Exact and simplified model based joint torques

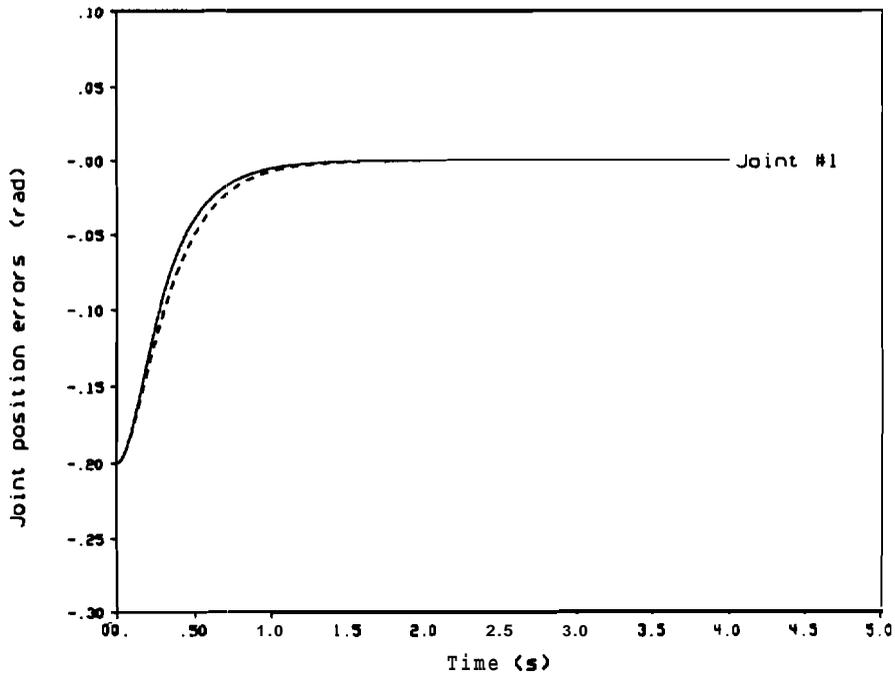


Fig.4a Exact and simplified model based position errors

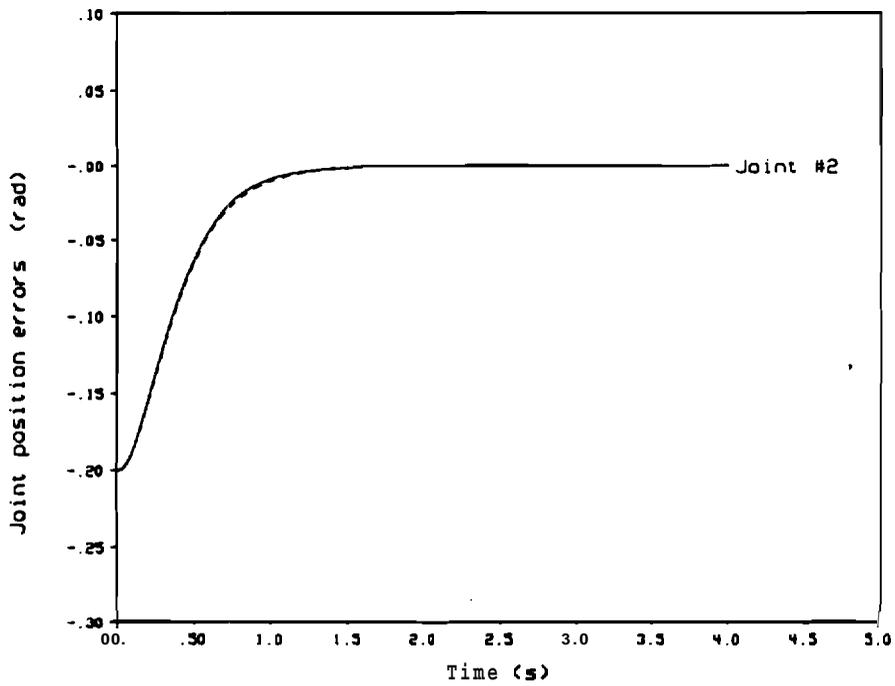


Fig.4b Exact and simplified model based position errors

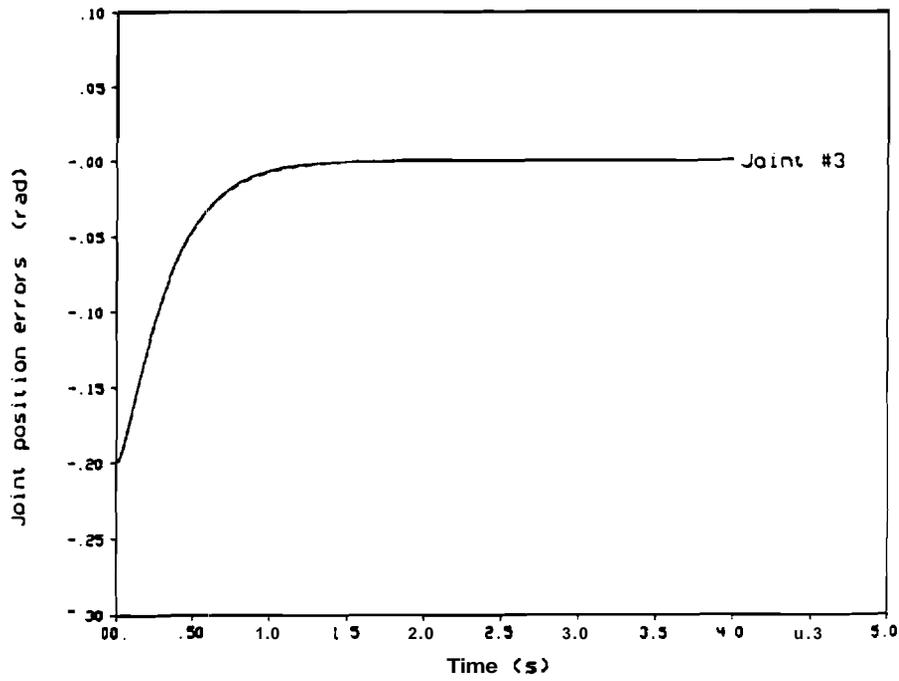


Fig.4c Exact and simplified model based position errors

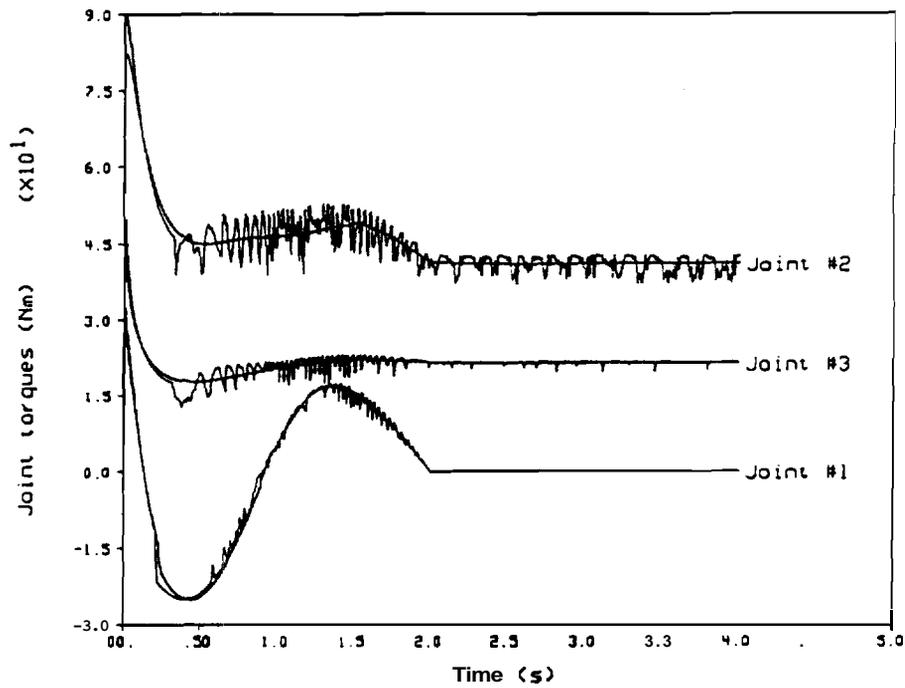


Fig.5 Joint torques For robust and adaptive control

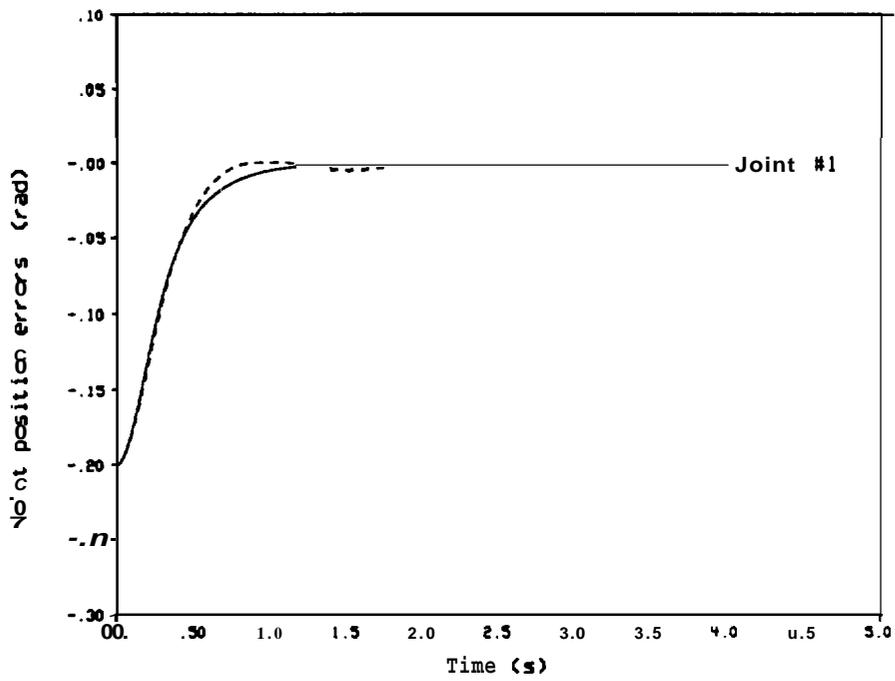


Fig.6a Position errors for robust and adaptive control

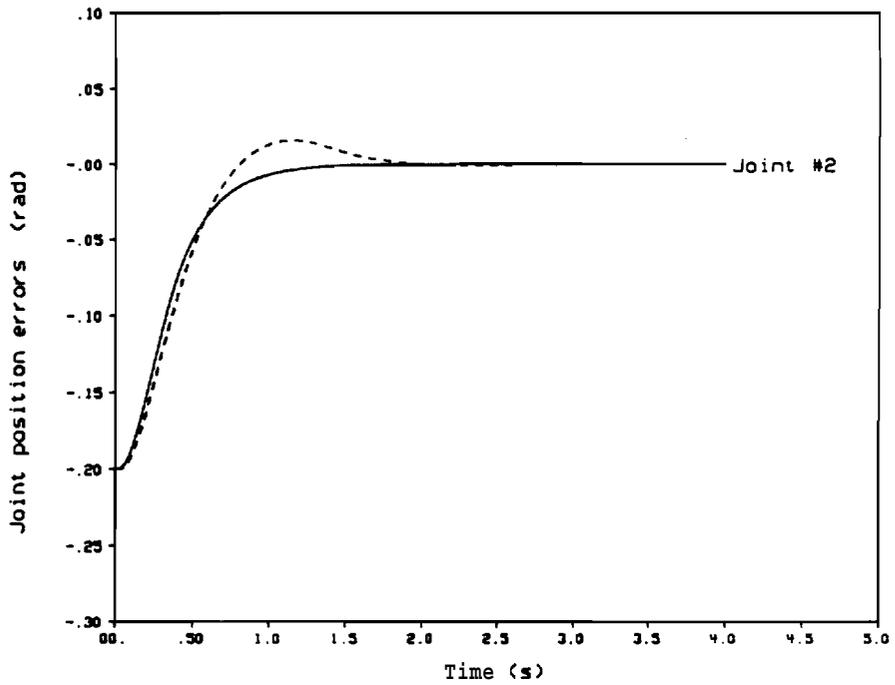


Fig.6b Position errors for robust and adaptive control

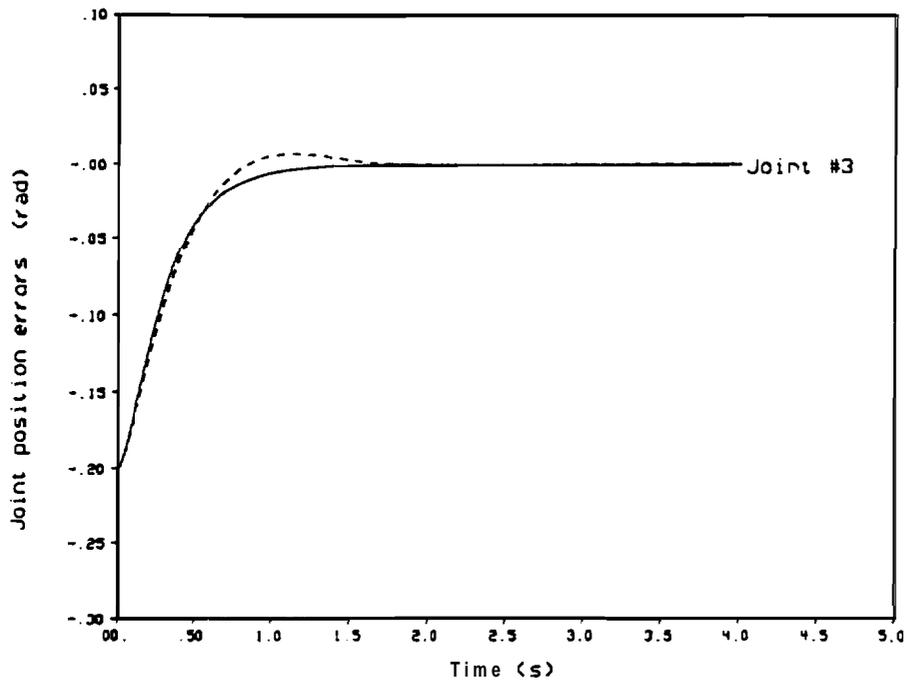


Fig.6c Position errors for robust and adaptive control