9-1-1988

Simplification of Manipulator Dynamic Model for Nonlinear Decoupled Control

J. W. Jeon
Purdue University

C S. G. Lee
Purdue University

Simplification of Manipulator Dynamic Model for Nonlinear Decoupled Control

J. W. Jeon
C. S. G. Lee

TR-EE 88-45
September 1988

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Simplification of Manipulator Dynamic Model for Nonlinear Decoupled Control

J. W. Jeon and C. S. G. Lee

School of Electrical Engineering

Purdue University

West Lafayette, Indiana 47907

TR-EE 88-45

September 1988

ABSTRACT

This paper presents the development of simplified manipulator dynamic models which satisfy the desired steady-state error specification in the joint-variable space or in the Cartesian space under a nonlinear decoupled controller. The formulae which relate the tracking errors of joint variables in the joint-variable space or the manipulator hand in the Cartesian space to the dynamic modeling errors are first developed. Using these formulae, we derive the maximum error tolerance for each dynamic coefficient of the equations of motion. Then each simplified dynamic coefficient of the equations of motion can be expressed as a linear combination of the product terms of sinusoidal and polynomial basis functions. To illustrate the approach, a computer simulation has been carried out to obtain two simplified dynamic models of a Stanford robot arm which satisfy the specified error tolerances in the joint-variable space and in the Cartesian space under respective nonlinear decoupled controllers. Finally, to measure the time complexity of simplified models, the number of mathematical operations in terms of multiplication and addition for computing the joint torques is tabulated and discussed with the parallel computation result of Newton-Euler equations of motion.

This work was supported in part by the National Science Foundation Grant CDR-8803017 and by a grant from the Ford Fund. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the funding agency.

1. Introduction

Robot manipulators are highly nonlinear systems and their dynamic performance is directly dependent on the efficiency of the control laws and the dynamic model of the robot. Past work focused mainly on designing efficient and robust controllers for manipulators [7-9]. This paper focuses on the inverse problem and addresses the issue of how much manipulator dynamics information should be included in the manipulator dynamic model for a nonlinear decoupled controller such that the controlled robotic system satisfies the desired steady-state system performance in the joint-variable space or in the Cartesian space . We further show the efficiency of simplified models by considering the computational complexity of the controllers based on these simplified models.

The analysis and design of robot motion control strategies require the development of efficient closed-form dynamic equations. Two competing approaches [1],[13] have been developed for handling the mathematical complexities involved in the dynamic model of robot manipulators. In the first approach, the emphasis focuses on the formulation 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 [13]. 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. In the second approach, the emphasis is on the formulation of 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 [1]. This is motivated by the growing interest in applying advanced control theory to robot manipulators [7-9]. 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, the development of efficient schemes/algorithms for obtaining a simplified model that reveals the dominant dynamics without introducing significant errors into the dynamic model is essential for the advanced control of robot manipulators.

Various schemes/algorithms have been proposed for simplifying robot arm dynamic model [1],[2],[4],[12]. Their approaches fall into one of the following categories:

(1) The reduction-rule and basis-function techniques which eliminate relatively insignificant inertial, Coriolis and centrifugal, and gravity terms to arrive at an approximate model based on the relative importance of their forces/torques as compared to the complete equations of motion [1],[2].

(2) A projection algorithm which, based on a least-square criterion, minimizes the l_2 norm error between the approximant and the nonlinear robot manipulator model [4]. One of the drawbacks of this method is the requirement to testing all the terms of a specific dynamic coefficient exhaustively for their significance.

(3) The expression of each dynamic coefficient is expressed as a linear combination of the basis functions and a minimax curve-fitting technique is used to provide an approximate model [12].

All the above existing simplification schemes described various methods of obtaining simplified dynamic models as compared to the complete Euler-Lagrange (EL) formulation. But all of them did not analyze the effects of the simplified dynamic model on the manipulator system performance. That is, they did not describe the effect and the relationship between the modeling error in the equations of motion and the controlled system performance. Recently, Chang and Lee [3] developed a multi-layered minimax simplification scheme which obtains a simplified dynamic model in symbolic form based on the desired manipulator steady-state error system performance in the joint-variable space under a proportional-plus-derivative (PD) controller. This minimax simplification scheme constructs the simplified dynamic coefficients using the basis functions. Each simplified dynamic coefficient is expressed as a linear combination of the product terms of sinusoidal and polynomial functions of the manipulator's joint variables. Furthermore, the robot system under a PD controller using the simplified dynamic model satisfies the desired steady-state system performance in the joint-variable space. However, since the manipulator is a serial-chained mechanism, a small error in each joint variable in the joint-variable space control may propagate the error through the chain and cause a moderate error of the manipulator end-effector in the Cartesian space [9]. As a result, the simplified dynamic model derived in the joint-variable space and based on the steady-state error specification in the joint-variable space may cause a moderate steady-state error in the Cartesian space. Since the dynamic behavior of the manipulator end-effector in the Cartesian space is one of the most significant characteristics in evaluating the performance of the manipulator [8],[9], we shall develop methods for deriving simplified dynamic models for satisfying the tracking error in the Cartesian space as well as the joint-variable space under respective nonlinear decoupled controllers.

In this paper, we extend the multi-layered minimax simplification scheme [3] to obtain a simplified dynamic model explicitly expressed in a symbolic form based on the desired manipulator steady-state error specification in the joint-variable space or in the Cartesian space under respective nonlinear decoupled controllers. We first derive the formulae which relate the tracking error of each joint variable in the joint-variable space or the manipulator hand in the Cartesian space to the dynamic modeling error. Using these formulae, the maximum error tolerance for each dynamic coefficient of the equations of motion is derived. These maximum error tolerance specifications are then used in the multi-layered minimax simplification scheme to obtain a simplified dynamic model. Using the simplified dynamic model and under the control of nonlinear decoupled controller, the controlled system will satisfy the desired steady-state error conditions in the joint-variable space or in the Cartesian space. A computer simulation was performed on a Stanford arm to verify the proposed simplification scheme. Finally, to measure the time complexity of simplified models, the number of mathematical operations in terms of multiplication and addition for computing the joint torques is tabulated and discussed with the parallel computation result of Newton-Euler equations of motion.

2. Relation of Steady-State Error of Joint Variables and the Modeling Error

In this section, we shall derive the formulae which relate the tracking error of each joint variable in the joint-variable space to the dynamic modeling error under a nonlinear decoupled controller. This error relationship will be used in the minimax simplification scheme in obtaining a simplified dynamic model to be used for computing the nonlinear decoupled controller. The robot system under the nonlinear decoupled controller using the simplified dynamic model will satisfy the control system performance. The equations of motion of an n-jointed manipulator expressed in the joint-variable space can be written as [1],[6]

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}(t) + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{\Gamma}(t) \tag{1}$$

where $\mathbf{D}(\mathbf{q})$ is the $n \times n$ kinetic energy matrix, $\mathbf{q}(t)$ is the $n \times 1$ vector of the generalized coordinates (or joint variables), $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ is the $n \times 1$ vector of the Coriolis and centrifugal force/torque, $\mathbf{g}(\mathbf{q})$ is the $n \times 1$ vector of the gravitational force/torque, and $\mathbf{\Gamma}(t)$ is the $n \times 1$ applied joint force/torque vector. Expressing the above equation in its components, we have

$$\sum_{j=1}^{n} d_{ij} \ddot{q}_j(t) + \sum_{j=1}^{n} \sum_{k=1}^{n} h_{ijk} \dot{q}_j(t) \dot{q}_k(t) + g_i = \tau_i(t) \ , \ i = 1, 2, \cdots, n \tag{2}$$

where d_{ij}, h_{ijk} and g_i are dynamic coefficients of the manipulator equations of motion. A nonlinear decoupled controller $\mathbf{\Gamma}_c$ to control the joint variables of a manipulator in the joint-variable space has the following form [6] (see Fig. 1)

$$\mathbf{\Gamma}_c = \mathbf{D}_c(\mathbf{q}) \, [\, \ddot{\mathbf{q}}_d(t) + \mathbf{K}_v \, \dot{\mathbf{e}}(t) + \mathbf{K}_p \, \mathbf{e}(t) \,] + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q}) \tag{3}$$

where $\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t)$ and $\dot{\mathbf{e}}(t) = \dot{\mathbf{q}}_d(t) - \dot{\mathbf{q}}(t)$ are, respectively, the tracking errors of joint position and joint velocity in the joint-variable space. $\mathbf{K}_v$ and $\mathbf{K}_p$ are, respectively, the velocity and position feedback diagonal matrix gains. $\mathbf{D}_c(\mathbf{q})$, $\mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{g}_c(\mathbf{q})$ represent the computed values of $\mathbf{D}(\mathbf{q})$, $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$, respectively, in (1). Manipulating equation (3), we obtain

$$\mathbf{\Gamma}_c = \mathbf{D}_c(\mathbf{q}) \, [\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v \, \dot{\mathbf{e}}(t) + \mathbf{K}_p \, \mathbf{e}(t) \,] + \mathbf{D}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q}) \tag{4}$$

where $\ddot{\mathbf{e}}(t) = \ddot{\mathbf{q}}_d(t) - \ddot{\mathbf{q}}(t)$ is the tracking error of joint acceleration in the joint-variable space. To achieve better robustness, a high gain feedback to the nonlinear decoupled controller is added in (4) [15] (see Fig. 2). Then the computed torque $\mathbf{\Gamma}_c$ of the nonlinear decoupled controller becomes

$$\mathbf{\Gamma}_c = k_f \; \mathbf{D}_c(\mathbf{q}) \, [\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v \; \dot{\mathbf{e}}(t) + \mathbf{K}_p \; \mathbf{e}(t) \,] + \mathbf{D}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q}) \tag{5}$$

where k_f is a scalar gain. Substituting the computed controller from (5) into (1), we have

$$\begin{aligned} \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = k_f \; \mathbf{D}_c(\mathbf{q}) \, [\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v \; \dot{\mathbf{e}}(t) + \mathbf{K}_p \; \mathbf{e}(t) \,] \\ + \mathbf{D}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q}) \end{aligned} \tag{6}$$

Manipulating equation (6), we obtain

$$k_f \; \mathbf{D}_c(\mathbf{q}) \, (\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v \; \dot{\mathbf{e}}(t) + \mathbf{K}_p \; \mathbf{e}(t) \,) = \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}) \tag{7}$$

where

$$\Delta\mathbf{D}(\mathbf{q}) \triangleq \mathbf{D}(\mathbf{q}) - \mathbf{D}_c(\mathbf{q}), \tag{8}$$

$$\Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) \triangleq \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}), \tag{9}$$

$$\Delta\mathbf{g}(\mathbf{q}) \triangleq \mathbf{g}(\mathbf{q}) - \mathbf{g}_c(\mathbf{q}). \tag{10}$$

Since $\mathbf{D}_c(\mathbf{q})$ is invertible, we have

$$\ddot{\mathbf{e}}(t) + \mathbf{K}_v \; \dot{\mathbf{e}}(t) + \mathbf{K}_p \; \mathbf{e}(t) = \frac{1}{k_f} \; \mathbf{D}_c^{-1}(\mathbf{q}) \, [\, \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}) \,] \, . \tag{11}$$

If no modeling error exists, then the right-hand-side of (11) vanishes, resulting in

$$\ddot{\mathbf{e}}(t) + \mathbf{K}_v \; \dot{\mathbf{e}}(t) + \mathbf{K}_p \mathbf{e}(t) = \mathbf{0}, \tag{12}$$

and if the values of $\mathbf{K}_v$ and $\mathbf{K}_p$ are so chosen that the characteristic roots of (12) have negative real parts, then $\mathbf{e}(t)$ approaches zero asymptotically [14]. However, since it is impossible to have an exact dynamic model of a manipulator and the complete Euler-Lagrange equations of motion may also have some modeling errors, it is more cost-effective to use an approximate model which does not introduce significant errors into the dynamic model and the controlled system still satisfies the desired steady-state system performance. Using an approximate model, the joint variable error $\mathbf{e}(t)$ in (11) may approach nonzero value due to the dynamic modeling error. Thus (11) can be used to derive the relation of the maximum error in the dynamic coefficients and the tracking error in the joint-variable space. Considering the i th component of the vector equation in (11), we have

$$\ddot{e}_i(t) + k_{vi} \; \dot{e}_i(t) + k_{pi} \, e_i(t) = \frac{1}{k_f} [\, \mathbf{D}_c^{-1}(\mathbf{q}) \,]_i \; [\, \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}) \,] \tag{13}$$

where

$$k_{vi} = \text{ the } i\text{th diagonal element of } \mathbf{K}_v \ , \tag{14}$$

$$k_{pi} = \text{ the } i\text{th diagonal element of } \mathbf{K}_p \ , \tag{15}$$

$$[\mathbf{D}_c^{-1}(\mathbf{q})]_i = \text{the } i\text{th row of } [\mathbf{D}_c^{-1}(\mathbf{q})], \quad i = 1, 2, \cdots, n \tag{16}$$

and $e_i(t)$, $\dot{e}_i(t)$, and $\ddot{e}_i(t)$ are the ith component of the error vectors $\mathbf{e}(t)$, $\dot{\mathbf{e}}(t)$, and $\ddot{\mathbf{e}}(t)$, respectively.

3. Effect of Simplified Model on Steady-State Error in the Joint-Variable Space

To determine the steady-state error of the joint variables in the joint-variable space, (13) is transformed into its Laplace transform equivalence

$$E_i(s) = \frac{\delta_i(s)}{k_f \ (s^2 + k_{vi} s + k_{pi})} \tag{17}$$

where $E_i(s)$ and $\delta_i(s)$ are the Laplace transforms of $e_i(t)$ and $[\mathbf{D}_c^{-1}(\mathbf{q})]_i [\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q})]$, respectively. If the desired path $\mathbf{q}_d(t)$ to each joint variable is a unit step (i.e. a constant displacement), then the velocity-related and acceleration-related terms of the modeling error will disappear in the steady state because they are functions of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$. Thus, to compute the steady-state error, we need only to consider the following

$$\left| \int_0^\infty [(\mathbf{D}_c^{-1}(\mathbf{q}))_i \ \Delta\mathbf{g}(\mathbf{q})] \, e^{-st} \, dt \right| \le \int_0^\infty \left| [\mathbf{D}_c^{-1}(\mathbf{q})]_i \ \Delta\mathbf{g}(\mathbf{q}) \right| \ e^{-st} \, dt \ . \tag{18}$$

Using the following convenient notations,

$$r_{ij} \triangleq \max_{\mathbf{q}} \left| \ [\mathbf{D}^{-1}(\mathbf{q})]_{ij} \right| = \max_{\mathbf{q}} \left| \text{ the } ij\text{th element of } [\mathbf{D}^{-1}(\mathbf{q})] \right| \ , \tag{19}$$

$$r_{ij}^c \triangleq \max_{\mathbf{q}} \left| \ [\mathbf{D}_c^{-1}(\mathbf{q})]_{ij} \right| = \max_{\mathbf{q}} \left| \text{ the } ij\text{th element of } [\mathbf{D}_c^{-1}(\mathbf{q})] \right| \ , \tag{20}$$

the absolute value inside the integral in (18) becomes

$$\begin{aligned} \left| \ [\mathbf{D}_c^{-1}(\mathbf{q})]_i \ \Delta\mathbf{g}(\mathbf{q}) \right| &\le \sum_{j=1}^{n} r_{ij}^c \left| \ \Delta g_j(\mathbf{q}) \right| \\ &\le \sum_{j=1}^{n} r_{ij}^c \left| \ \Delta g_j \right|_m \end{aligned} \tag{21}$$

where

$$\Delta\mathbf{g}(\mathbf{q}) = [\Delta g_1(\mathbf{q}), \cdots, \Delta g_n(\mathbf{q})]^T \ , \tag{22}$$

$$| \Delta g_j |_m = \max_{\mathbf{q}} | \Delta g_j(\mathbf{q}) | , \tag{23}$$

and the superscript "T" denotes vector or matrix transpose. Thus, (18) reduces to

$$| \int_0^\infty [(\mathbf{D}_c^{-1}(\mathbf{q}))_i \, \Delta \mathbf{g}(\mathbf{q})] e^{-st} \, dt | \le \frac{1}{s} \sum_{j=1}^{n} r_{ij}^c | \Delta g_j |_m . \tag{24}$$

Applying the final value theorem to (17), and using (24), we have

$$\begin{aligned} e_{ssp}^i &= \lim_{t \to \infty} e_i(t) = \lim_{s \to 0} sE_i(s) \\ &\le \frac{1}{k_{fa} k_{fb} \, k_{pi}} \sum_{j=1}^{n} r_{ij}^c | \Delta g_j |_m \end{aligned} \tag{25}$$

where k_{fa} is larger than 1 and $k_{fb} = \frac{k_f}{k_{fa}}$. If we determine k_{fa} such that for all i, j

$$\frac{1}{k_{fa}} r_{ij}^c \le r_{ij} , \tag{26}$$

then the steady-state error is

$$e_{ssp}^i \le \frac{1}{k_{fb} \, k_{pi}} \sum_{j=1}^{n} r_{ij} | \Delta g_j |_m . \tag{27}$$

In the case that the modeling error in the kinetic energy matrix $\mathbf{D}(\mathbf{q})$ is small, k_{fa} is approximately 1. The value of k_{fa} is determined such that the inequality in (26) holds in the computer simulation. If n steady-state error specifications ε_{ssp}^i of the manipulator are given, $1 \le i \le n$, and under the condition that the desired path is a unit step, a sufficient condition for the actual steady-state error to be less than or equal to the error specification (i.e., $| e_{ssp}^i | \le \varepsilon_{ssp}^i$) is

$$\frac{1}{k_{fb} \, k_{pi}} \sum_{j=1}^{n} r_{ij} | \Delta g_j |_m \le \varepsilon_{ssp}^i . \tag{28}$$

Thus we have obtained n linear inequalities for bounding the maximum modeling error in each gravity dynamic coefficient (i.e., g_i in (2)). Equation (28) indicates that the problem of finding the maximum modeling error in each gravity coefficient is classified as a linear programming problem or a nonlinear programming problem depending on whether a linear combination or a nonlinear combination of $| \Delta g_i |_m$, $1 \le i \le n$, is being used in the objective function. To optimize the objective function, the problem can be solved by several methods [11]. It is worth pointing out that some of the $g_j(\mathbf{q})$ may be zero or constant due to the manipulator structure

[1],[2],[16]. For example, in a Stanford arm, $g_1(\mathbf{q})=0, g_6(\mathbf{q})=0$, we need only consider the objective function consisting of a linear or nonlinear combination of $|\Delta g_i|_m$, $2 \le i \le 5$. Furthermore, if we want to allow a larger error bound for some $g_j(\mathbf{q})$, we can use a particular objective function which weights $|\Delta g_j|_m$ heavier.

If the desired path $\mathbf{q}_d(t)$ to each joint variable is a unit ramp (i.e., tracking a stationary object on a moving conveyor belt), then the acceleration-related terms of the modeling error will disappear in the steady state. Thus, to compute the steady-state error, we need only to consider the following

$$\left| \int_0^\infty [(\mathbf{D}_c^{-1}(\mathbf{q}))_i (\Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}))] e^{-st} dt \right| \tag{29}$$

$$\le \int_0^\infty [| (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})| + | (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{g}(\mathbf{q})|] e^{-st} dt .$$

Taking the similar step as before, the first absolute value term inside the integral becomes

$$| (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}})| \le \sum_{j=1}^n r_{ij}^c \left| \sum_{k=1}^n \sum_{l=1}^n \Delta h_{jkl}(\mathbf{q}) \dot{q}_k \dot{q}_l \right|$$

$$\le \sum_{j=1}^n r_{ij}^c \sum_{k=1}^n \sum_{l=1}^n | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m \tag{30}$$

where

$$| \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m = \max_{\mathbf{q},\dot{\mathbf{q}}} | \Delta h_{jkl}(\mathbf{q}) \dot{q}_k \dot{q}_l | . \tag{31}$$

Using (21) and (30), (29) becomes

$$\left| \int_0^\infty [(\mathbf{D}_c^{-1}(\mathbf{q}))_i (\Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}))] e^{-st} dt \right| \tag{32}$$

$$\le \frac{1}{s} \sum_{j=1}^n r_{ij}^c [\sum_{k=1}^n \sum_{l=1}^n | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m + | \Delta g_j |_m] .$$

Applying the final value theorem to (17), and using (26) and (32), the steady-state error due to a unit ramp input becomes

$$e_{ssv}^i \le \frac{1}{k_{fb}\, k_{pi}} \sum_{j=1}^n r_{ij} [\sum_{k=1}^n \sum_{l=1}^n | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m + | \Delta g_j |_m] . \tag{33}$$

If n steady-state error specifications ε_{ssv}^i of the manipulator are given, $1 \le i \le n$, and under the condition that the desired path is a unit ramp, a sufficient condition for the steady-state error to be less than or equal to the error specification (i.e., $|e_{ssv}^i| \le \varepsilon_{ssv}^i$) is

$$\frac{1}{k_{fb}\,k_{pi}} \sum_{j=1}^{n} r_{ij} \left[\sum_{k=1}^{n} \sum_{l=1}^{n} | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m + | \Delta g_j |_m \right] \le \varepsilon_{ssv}^i . \tag{34}$$

In order to know the relative importance of each term of (34), we assume that the relative ratio of the maximum deviation to the corresponding maximum value of non-zero term in (34) is equal. [3] That is, for a fixed j, $1 \le j \le n$,

$$\frac{|\Delta g_j|_m}{|g_j|_m} = \frac{|\Delta h_{jkl}|_m}{|h_{jkl}|_m} \tag{35}$$

where, for $1 \le k, l \le n$,

$$|g_j|_m = \max_{\mathbf{q}} |g_j(\mathbf{q})| \tag{36}$$

and

$$|h_{jkl}|_m = \max_{\mathbf{q}} |h_{jkl}(\mathbf{q})| . \tag{37}$$

Then

$$\frac{|\Delta g_j|_m}{|g_j|_m} = \frac{|\Delta h_{jkl}|_m}{|h_{jkl}|_m} = \frac{\sum_{k=1}^{n}\sum_{l=1}^{n} |\Delta h_{jkl} \dot{q}_k \dot{q}_l|_m + |\Delta g_j|_m}{\sum_{k=1}^{n}\sum_{l=1}^{n} |h_{jkl} \dot{q}_k \dot{q}_l|_m + |g_j|_m} \tag{38}$$

where $|h_{jkl}\dot{q}_k\dot{q}_l|_m = \max_{\mathbf{q},\dot{\mathbf{q}}} |h_{jkl}(\mathbf{q})\dot{q}_k\dot{q}_l|$. The denominator in (38), $|g_j|_m$ and $|h_{jkl}\dot{q}_k\dot{q}_l|_m$, can be found by a searching strategy in the manipulator workspace. Using (38), (34) can be written as

$$\frac{1}{k_{fb}\,k_{pi}} \sum_{j=1}^{n} r_{ij} \frac{|\Delta g_j|_m}{|g_j|_m} B_1^j \le \varepsilon_{ssv}^i \tag{39}$$

and

$$\frac{1}{k_{fb}\,k_{pi}} \sum_{j=1}^{n} r_{ij} \frac{|\Delta h_{jkl}|_m}{|h_{jkl}|_m} B_1^j \le \varepsilon_{ssv}^i \tag{40}$$

where, for a fixed j,

$$B_1^j = \sum_{k=1}^{n}\sum_{l=1}^{n} |h_{jkl}\dot{q}_k\dot{q}_l|_m + |g_j|_m . \tag{41}$$

As before, we have obtained n linear inequalities for bounding the maximum modeling error of the dynamic coefficients (i.e., h_{jkl} and g_j) in (39) and (40). To solve (39) and (40), we can define an objective function consisting of a combination of $| \Delta g_i |_m$ and $| \Delta h_{jkl} |_m$ in (39) and (40), respectively. Similarly, we can use the structure of $\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}})$ as well as that of $\mathbf{g}(\mathbf{q})$ [1],[2],[16] and if we want to allow a larger modeling error bound for some $h_{jkl}(\mathbf{q})$, we can use a particular objective function which weights $| \Delta h_{jkl} |_m$ heavier.

If the desired path $\mathbf{q}_d(t)$ to each joint variable is a parabolic, $\frac{t^2}{2}$, then we have

$$| \delta_i(s) | = | \int_0^\infty [(\mathbf{D}_c^{-1}(\mathbf{q}))_i (\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q}))] e^{-st} \, dt |$$

$$\leq \int_0^\infty [| (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} | + | (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) | + | (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{g}(\mathbf{q}) |] e^{-st} \, dt \, . \qquad (42)$$

Taking the similar step as before, the first absolute value term inside the integral becomes

$$| (\mathbf{D}_c^{-1}(\mathbf{q}))_i \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} | \leq \sum_{j=1}^{n} r_{ij}^c | \sum_{k=1}^{n} \Delta d_{jk}(\mathbf{q}) \ddot{q}_k |$$
$$\leq \sum_{j=1}^{n} r_{ij}^c \sum_{k=1}^{n} | \Delta d_{jk} \ddot{q}_k |_m \qquad (43)$$

where

$$| \Delta d_{jk} \ddot{q}_k |_m = \max_{\mathbf{q}, \ddot{\mathbf{q}}} | \Delta d_{jk}(\mathbf{q}) \ddot{q}_k | \, . \qquad (44)$$

Using (21), (30), and (43), (42) becomes

$$| \delta_i(s) | \leq \frac{1}{s} \sum_{j=1}^{n} r_{ij}^c [\sum_{k=1}^{n} | \Delta d_{jk} \ddot{q}_k |_m + \sum_{k=1}^{n} \sum_{l=1}^{n} | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m + | \Delta g_j |_m] \, . \qquad (45)$$

Applying the final value theorem to (17), and using (26) and (45), the steady-state error due to a parabolic input becomes

$$e_{ssa}^i \leq \frac{1}{k_{fb} \, k_{pi}} \sum_{j=1}^{n} r_{ij} [\sum_{k=1}^{n} | \Delta d_{jk} \ddot{q}_k |_m + \sum_{k=1}^{n} \sum_{l=1}^{n} | \Delta h_{jkl} \dot{q}_k \dot{q}_l |_m + | \Delta g_j |_m] \, . \qquad (46)$$

If n steady-state error specifications ε_{ssa}^i of the manipulator are given, $1 \leq i \leq n$, and under the condition that the desired path is a parabolic, a sufficient condition for the steady-state error to be less than or equal to the error specification (*i.e.* $| e_{ssa}^i | \leq \varepsilon_{ssa}^i$) is

$$\frac{1}{k_{fb}\,k_{pi}}\sum_{j=1}^{n} r_{ij}\,[\,\sum_{k=1}^{n}|\,\Delta d_{jk}\ddot{q}_k\,|_m + \sum_{k=1}^{n}\sum_{l=1}^{n}|\,\Delta h_{jkl}\dot{q}_k\dot{q}_l\,|_m + |\,\Delta g_j\,|_m\,] \le \varepsilon_{ssa}^{i}\,. \tag{47}$$

As before, if we assume that the relative ratio of the maximum deviation to the corresponding maximum value of the non-zero term in (47) is equal, then we obtain n linear inequalities for bounding the maximum modeling error in each of the dynamic coefficients (i.e., d_{jk} , h_{jkl}, and g_j) in (47).

$$\frac{1}{k_{fb}\,k_{pi}}\sum_{j=1}^{n} r_{ij}\frac{|\,\Delta g_j\,|_m}{|\,g_j\,|_m}B_2^j \le \varepsilon_{ssa}^{i} \tag{48}$$

$$\frac{1}{k_{fb}\,k_{pi}}\sum_{j=1}^{n} r_{ij}\frac{|\,\Delta h_{jkl}\,|_m}{|\,h_{jkl}\,|_m}B_2^j \le \varepsilon_{ssa}^{i} \tag{49}$$

$$\frac{1}{k_{fb}\,k_{pi}}\sum_{j=1}^{n} r_{ij}\frac{|\,\Delta d_{jk}\,|_m}{|\,d_{jk}\,|_m}B_2^j \le \varepsilon_{ssa}^{i} \tag{50}$$

where, for a fixed j,

$$B_2^j = \sum_{k=1}^{n}|\,d_{jk}\ddot{q}_k\,|_m + \sum_{k=1}^{n}\sum_{l=1}^{n}|\,h_{jkl}\dot{q}_k\dot{q}_l\,|_m + |\,g_j\,|_m\,. \tag{51}$$

Applying the similar method as before, we can obtain the maximum error bound in each dynamic coefficient.

4. Effect of Simplified Model on Steady-State Error in the Cartesian Space

In this section, we shall derive the formulae which relate the tracking error of the manipulator hand in the Cartesian space to the dynamic modeling error under a nonlinear decoupled controller. For ease of discussion, we shall assume that the manipulator under control is nonredundant ($n = 6$) and is always at a nonsingular configuration. The equations of motion of the manipulator end-effector can be expressed in the Cartesian space as [8],

$$\mathbf{\Lambda}\,(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}\,(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}\,(\mathbf{x}) = \mathbf{F}(t) \tag{52}$$

$$\dot{\mathbf{x}}(t) = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}(t) \tag{53}$$

where $\mathbf{x}(t)$ is the 6×1 vector of the Cartesian variables describing the position and orientation of the manipulator hand, $\mathbf{J}(\mathbf{q})$ is the 6×6 Jacobian matrix, $\mathbf{\Lambda}(\mathbf{x})$ is the 6×6 kinetic energy matrix of the manipulator,

$$\mathbf{\Lambda}\,(\mathbf{x}) = \mathbf{J}^{-T}(\mathbf{q})\,\mathbf{D}(\mathbf{q})\,\mathbf{J}^{-1}(\mathbf{q}), \tag{54}$$

$\mu(\mathbf{x}, \dot{\mathbf{x}})$ is the 6×1 vector of the Coriolis and centrifugal force,

$$\mu(\mathbf{x}, \dot{\mathbf{x}}) = \mathbf{J}^{-T}(\mathbf{q})\, \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}^{-T}(\mathbf{q})\, \mathbf{D}(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})\, \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \tag{55}$$

$\mathbf{p}(\mathbf{x})$ is the 6×1 vector of the gravitational force,

$$\mathbf{p}(\mathbf{x}) = \mathbf{J}^{-T}(\mathbf{q})\, \mathbf{g}(\mathbf{q}), \tag{56}$$

$\mathbf{F}(t)$ is the 6×1 vector of the generalized force and is related to the joint torques through the manipulator Jacobian,

$$\Gamma(t) = \mathbf{J}^{T}(\mathbf{q})\mathbf{F}(t), \tag{57}$$

$\dot{\mathbf{x}}(t)$ and $\ddot{\mathbf{x}}(t)$ are the first and second time derivatives of $\mathbf{x}(t)$, respectively.

To control the manipulator end-effector to track a desired motion trajectory, we use a nonlinear decoupled controller in the Cartesian space, which is similar to (5) (See Fig. 3). The computed generalized force of the nonlinear decoupled controller is

$$\mathbf{F}_c(t) = k_f\, \Lambda_c(\mathbf{x})\, (\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v\, \dot{\mathbf{e}}(t) + \mathbf{K}_p\, \mathbf{e}(t)\,) + \Lambda_c(\mathbf{x})\ddot{\mathbf{x}} + \mu_c(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{p}_c(\mathbf{x}) \tag{58}$$

where $\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t)$ is the 6×1 error vector of the position and orientation of the manipulator hand in the Cartesian space. $\mathbf{K}_v$ and $\mathbf{K}_p$ are, respectively, the velocity and position feedback diagonal matrix gains, k_f is a scalar gain. $\Lambda_c(\mathbf{x})$, $\mu_c(\mathbf{x})$, and $\mathbf{p}_c(\mathbf{x})$ represent, respectively, the computed values of the $\Lambda(\mathbf{x})$, $\mu(\mathbf{x}, \dot{\mathbf{x}})$, and $\mathbf{p}(\mathbf{x})$ in (52). $\Lambda_c(\mathbf{x})$, $\mu_c(\mathbf{x})$, and $\mathbf{p}_c(\mathbf{x})$ have similar relations to $\mathbf{D}_c(\mathbf{q})$, $\mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{g}_c(\mathbf{q})$, respectively, as in (54)-(56). From (54)-(58), the computed torque Γ_c of the nonlinear decoupled controller is

$$\begin{aligned}\Gamma_c &= k_f\, \mathbf{D}_c(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})(\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v\, \dot{\mathbf{e}}(t) + \mathbf{K}_p\, \mathbf{e}(t)\,) \\ &\quad + \mathbf{D}_c(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})(\, \ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\,) + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q})\end{aligned} \tag{59}$$

Substituting the computed torque from (59) into (1), we have

$$\begin{aligned}\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) &= k_f\, \mathbf{D}_c(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})(\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v\, \dot{\mathbf{e}}(t) + \mathbf{K}_p\, \mathbf{e}(t)\,) \\ &\quad + \mathbf{D}_c(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})(\, \ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}\,) + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q})\end{aligned} \tag{60}$$

Taking the time derivative of (53) and manipulating the resulted equation, we have

$$\ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\, \dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}}, \tag{61}$$

and substituting it into (60) and manipulating the equation, we obtain

$$k_f\, \mathbf{D}_c(\mathbf{q})\, \mathbf{J}^{-1}(\mathbf{q})\, (\, \ddot{\mathbf{e}}(t) + \mathbf{K}_v\, \dot{\mathbf{e}}(t) + \mathbf{K}_p\, \mathbf{e}(t)\,) = \Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q}, \dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q})\,. \tag{62}$$

Since $\mathbf{D}_c(\mathbf{q})$ is invertible, we have

$$\ddot{\mathbf{e}}(t) + \mathbf{K}_v\,\dot{\mathbf{e}}(t) + \mathbf{K}_p\,\mathbf{e}(t) = \frac{1}{k_f}\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,[\,\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q})\,]\,. \quad (63)$$

In deriving (63), we assume that the manipulator is nonredundant and always at a nonsingular location. In fact, in the case of a redundant manipulator with a full rank of the Jacobian matrix, the same relation holds (See the Appendix). However, in the singular case or non-full rank of the Jacobian matrix, further investigation is necessary. It is interesting to note that (11) and (63) are in the same form except the absence of the Jacobian matrix in (11).

If no modeling error exists, then the right-hand-side of (63) vanishes, resulting in

$$\ddot{\mathbf{e}}(t) + \mathbf{K}_v\,\dot{\mathbf{e}}(t) + \mathbf{K}_p\mathbf{e}(t) = \mathbf{0}\,, \quad (64)$$

and if the values of $\mathbf{K}_v$ and $\mathbf{K}_p$ are so chosen that the characteristic roots of (64) have negative real parts, then $\mathbf{e}(t)$ approaches zero asymptotically. Using an approximate model, the Cartesian error $\mathbf{e}(t)$ in (63) may approach nonzero value due to the dynamic modeling error. Thus (63) can be used to derive the relation of the maximum error in the dynamic coefficients and the steady-state error of the manipulator hand in the Cartesian space. Considering the ith component of the error vector in (63), we have

$$\ddot{e}_i(t) + k_{vi}\,\dot{e}_i(t) + k_{pi}\,e_i(t) = \frac{1}{k_f}\,[\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,]_i\,[\,\Delta\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q})\,] \quad (65)$$

where

$$[\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,]_i = \text{the } i\text{th row of } [\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,],\quad i = 1, 2, \cdots, n \quad (66)$$

and $e_i(t)$, $\dot{e}_i(t)$, and $\ddot{e}_i(t)$ are the ith component of the error vectors $\mathbf{e}(t)$, $\dot{\mathbf{e}}(t)$, and $\ddot{\mathbf{e}}(t)$, respectively.

Comparing (13) with (65), the same approach and method in Section 3 can be applied. We only need to change (19) and (20) to determine the maximum modeling error of each dynamic coefficient,

$$r_{ij} \triangleq \max_{\mathbf{q}} |\,[\,\mathbf{J}(\mathbf{q})\,\mathbf{D}^{-1}(\mathbf{q})\,]_{ij}\,| = \max_{\mathbf{q}} |\,\text{the } ij\text{th element of } [\,\mathbf{J}(\mathbf{q})\,\mathbf{D}^{-1}(\mathbf{q})\,]\,|\,, \quad (67)$$

$$r_{ij}^c \triangleq \max_{\mathbf{q}} |\,[\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,]_{ij}\,| = \max_{\mathbf{q}} |\,\text{the } ij\text{th element of } [\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,]\,|\,. \quad (68)$$

Then if we are given steady-state error specifications ε_{ssp}^i, ε_{ssv}^i, and ε_{ssa}^i of the manipulator,

k_f , $\mathbf{K}_v$, and $\mathbf{K}_p$ in the Cartesian space, the same method can be applied as in the joint-variable space.

5. Determination of Maximum Modeling Errors based on System Performance

In the above derivation of steady-state errors in the joint-variable space or the Cartesian space for unit step, unit ramp, and parabolic inputs, we can determine the maximum error tolerances (or *maximum modeling errors*) of the dynamic coefficients so that the manipulator control system using a simplified model under a nonlinear decoupled controller can still achieve the desired steady-state error specifications. Since most inputs consist of a linear combination of these three standard test signals, the maximum error tolerances for the dynamic coefficients of the equations of motion must be selected according to

$$\varepsilon_{g_i} \triangleq \min(\varepsilon_{ssp}(g_i), \varepsilon_{ssv}(g_i), \varepsilon_{ssa}(g_i)) \tag{69}$$

$$\varepsilon_{h_{ijk}} \triangleq \min(\varepsilon_{ssv}(h_{ijk}), \varepsilon_{ssa}(h_{ijk})) \tag{70}$$

$$\varepsilon_{d_{ij}} \triangleq \varepsilon_{ssa}(d_{ij}) \tag{71}$$

where $\varepsilon_{ssp}(\cdot)$, $\varepsilon_{ssv}(\cdot)$, and $\varepsilon_{ssa}(\cdot)$ are, respectively, the maximum modeling errors of $(\cdot)$ due to unit step, unit ramp, and parabolic inputs. The derivation and analytical expression in previous sections relating the maximum error tolerances of the dynamic coefficients to the steady-state error specifications of the manipulator control system leads us to an interesting question: Given the desired manipulator steady-state error specifications under a nonlinear decoupled controller, how can we determine the complexity of the manipulator dynamic model such that the manipulator control system can still achieve the desired performance? Thus, the complexity of the simplified dynamic model depends on the steady-state error specification of the manipulator system. An efficient minimax simplification scheme for reducing the cost of obtaining the dynamic coefficients of the simplified dynamic model to satisfy the desired steady-state error specifications has been proposed [3]. Following their approach and using the above derived maximum error tolerances of the dynamic coefficients, a simplified dynamic model can be obtained that satisfies the desired manipulator steady-state error specifications under a nonlinear decoupled control.

6. Computer Simulation

The multi-layered minimax simplification procedure in the joint-variable space has been implemented in a "C" program [3] and can be used to generate the simplified dynamic coefficients for any manipulator with prismatic and/or rotary joints. Here we used the software package for generating the simplified dynamic coefficients of the equations of motion based on the steady-state error specifications in the joint-variable space or the Cartesian space to satisfy the system performance under a nonlinear decoupled controller. The Stanford arm which consists of rotational and translational joints is used as an example to verify the simplification algorithm. Some of the parameters of the Stanford arm used in the computer simulation are listed in Table 1. In order not to excite the resonant frequency of the manipulator under a nonlinear decoupled controller in the joint-variable space [6], the position feedback diagonal matrix gain $\mathbf{K}_p$ is set such that the undamped natural frequency of each joint is less than one-half of the structural resonant frequency, and the velocity feedback diagonal matrix gain $\mathbf{K}_v$ is set to have a critically damped or an overdamped system. In a Cartesian nonlinear decoupled controller, $\mathbf{K}_p$ is set such that the undamped natural frequency of each decoupled Cartesian subsystem is equal to $2Hz$ [17], and the velocity feedback diagonal matrix gain $\mathbf{K}_v$ is similarly set as in the joint-variable space. We selected the same steady-state error specifications in the joint-variable space as in [3]. The steady-state error specifications in the Cartesian space can be generated from the steady-state error specifications in the joint-variable space through the Jacobian matrix (see Table 2). k_{fa} is set to 1.2 to validate the inequality in (26) and k_f is set to 30 for both nonlinear decoupled controllers. In a linear or a nonlinear programming problem under constraints (28), (39), (40), and (48)-(50), we can define some objective functions and solve the problem to optimize each objective function. As an example, an objective function consisting of the products of the maximum modeling errors can be defined to find $\varepsilon_{ssp}(g_j)$, $1 \le j \le n$,

$$f(|\Delta g_1|_m, |\Delta g_2|_m, \cdots, |\Delta g_n|_m) \triangleq \prod_{i=1}^{n} |\Delta g_i|_m . \tag{72}$$

Under the constraint of (28) for $i = 1$, the maximum value of the objective function is achieved by

$$| \Delta g_j |_m = \frac{1}{n r_{1j}} k_{fb} k_{p1} \varepsilon_{ssp}^1 \tag{73}$$

for all j, $1 \le j \le n$. Repeating for the other i, $2 \le i \le n$, $\varepsilon_{ssp}(g_1)$, is set to the minimum among all the values $| \Delta g_1 |_m$ that are determined by the same process as (73) for all $i, 1 \le i \le n$. After that, each $\varepsilon_{ssp}(g_j)$, for $2 \le j \le n$ is set to the minimum among values $| \Delta g_j |_m$ that are determined as following for all $i, 1 \le i \le n$,

$$| \Delta g_j |_m = \frac{1}{(n-j+1) r_{ij}} (k_{fb} k_{pi} \varepsilon_{ssp}^i - \sum_{l=1}^{(j-1)} r_{il} \varepsilon_{ssp}(g_l)) . \tag{74}$$

In fact, since the above solution was derived by considering each constraint successively rather than simultaneously, it is not optimal but suboptimal. If some of the $g_j(\mathbf{q})$ are zero or constant due to the manipulator structure, then we exclude those terms in (72) and apply the similar method. Similarly, we found the other maximum modeling error bounds for the remaining dynamic coefficients.

The ratio between the maximum force/torque contributed by a specific dynamic coefficient (such as d_{ij}, g_i or h_{ijk}) and the total maximum force/torque (such as B_1^i or B_2^i) is a criterion which can determine the relative significance of that dynamic coefficient. From our computer simulations, it was discovered that there are many dynamic coefficients that are insignificant. Table 3 and Table 4 list the significant dynamic coefficients of the simplified dynamic models which satisfy the steady-state error specifications in the joint-variable space and in the Cartesian space, respectively, under each respective nonlinear decoupled controller. As discussed in the previous sections, the maximum modeling error in each dynamic coefficient depends on the position gain matrix $\mathbf{K}_p$ which relates to undamped natural frequencies, the high feedback gain k_f and the steady-state error specification. Thus under respective nonlinear decoupled controllers, the time complexity of the simplified model depends on the steady-state error specification. A larger steady-state error specification will result in simpler dynamic coefficients with less number of basis function terms. Similarly, by adjusting the position gain matrix and the high feedback gain, various complexity of the dynamic coefficients can be obtained. Although a large magnitude of the high feedback gain is desirable to achieve simpler dynamic coefficients, however, in practice we cannot increase the high feedback gain without bound since a large magnitude of the high feedback gain will reduce the manipulator bandwidth. The trade-off must be considered [15]. A major bottleneck in computing respective nonlinear decoupled controllers (i.e., (5) and (59)) is to compute the dynamic terms

$$\mathbf{D}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_c(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q}) . \tag{75}$$

This computation is equivalent to the robot inverse dynamics computation [6]. Since various parallel algorithms have been developed to compute the robot inverse dynamics based on the Newton-Euler equations of motion [10], the efficiency of the simplified dynamic model(s) can be gauged by comparing the required number of mathematical operations in terms of multiplication and addition with those stated in [10]. Table 5 compares the time complexity of calculating (75) on a uniprocessor computer using simplified models with the parallel computation of Newton-Euler equations of motion on a multiprocessor system [10]. Table 5 shows that the computation of simplified dynamic models on a uniprocessor has about the same amount of computation as the parallel algorithms on a multiprocessor system with six microprocessors [10].

7. Conclusion

This paper presents the derivation of the formulae which relate the steady-state error in the joint-variable space or the manipulator end-effector steady-state error to the modeling error under respective nonlinear decoupled controllers. From the formulae, we could obtain the maximum admissible modeling errors in the dynamic coefficients while satisfying the desired steady-state error performance. Using the multi-layered minimax simplification algorithm, we obtained the significant dynamic coefficients of the simplified models of a Stanford arm. The complexity of computing a simplified dynamic model on a uniprocessor is quite comparable to those parallel algorithms on a multiprocessor system with six microprocessors. Furthermore, simplified dynamic models obtained from the minimax simplification scheme also satisfies the desired steady-state error specification under a nonlinear decoupled controller.

Acknowledgment. The authors would like to thank Y. Tang for his assistance in the computer

simulation.

8. References

[1] A. K. Bejczy, "Robot Arm Dynamics and Control," *JPL Technical Memorandum*, 33-669, Pasadena, CA, Feb. 1974.

[2] A. K. Bejczy and S. Lee, "Robot Arm Dynamic Model Reduction for Control," *Proc. of the 22nd IEEE Conf. of Decision and Control*, pp. 1466-1474, San Antonio, TX. Dec. 1983

[3] P. R. Chang and C. S. G. Lee, "Simplification of Robot Dynamic Model Based on System Performance," *Proc. of 26th IEEE Conference on Decision and Control*, Los Angeles, CA, pp. 1017-1023, December 9-11, 1987.

[4] A. A. Desrochers and C. M. Seaman, "A Projection Method for Simplifying Robot Manipulator Models," *Proc. 1986 IEEE Int'l. Conf. on Robotics and Automation*, pp. 504-509, San Francisco, CA, April 1986

[5] E. Freund, "The Structure of Decoupled Nonlinear Systems," *Int. J. Control*, vol. 21, no. 3, pp. 443-450, 1975.

[6] K. S. Fu, R. C. Gonzalez, and C.S.G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*, McGraw-Hill, 1987.

[7] E. G. Gilbert and I.J. Ha, "An approach to Nonlinear Feedback Control with Applications to Robotics," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-14, no. 6, pp 879-884, Nov./Dec. 1984.

[8] O. Khatib, "Unified Approach for Motion and Force Control of Robot Manipulators : The Operational Space Formulation," *IEEE J. of Robotics and Automation*, vol. RA-3, no. 1, pp 43-53, Feb. 1987.

[9] C. S. G. Lee and B. H. Lee, "Resolved Motion Adaptive Control for Mechanical Manipulators," *ASME J. of Dynam. Syst., Meas. and Contrl*, vol. 106, pp. 134-142, June 1984.

[10] C. S. G. Lee and P. R. Chang "Efficient Parallel Algorithm For Robot Inverse Dynamics Computation," *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-16, no. 4, pp 532-542, July/Aug. 1986.

[11] D. G. Luenberger *Linear and Nonlinear Programming*, Addison-Wesley, 1984.

[12] J. Y. S. Luh and C. S. Lin, "Automatic Generation of Dynamic Equations for Mechanical Manipulators," *Proc. 1981 of Joint Auto. Control Conf.*, TA 2-D, Charlottesville, VA, June 1981

[13] J. Y. S Luh, M. W. Walker and R. P. Paul, "On Line Computational Scheme for Mechanical Manipulators," *Trans. of ASME Dynamics, Measurement and Control*, vol. 102, pp. 69-76, June 1980.

[14] J. Y. S. Luh, M. W. Walker and R. P. Paul, "Resolved-Acceleration Control," *IEEE Trans. on Automatic Control*, vol. AC-25, no. 3, pp 468-474, June 1980.

[15] John Studenny and Pierre R. Belanger "Robot Manipulator Control By Acceleration Feedback: Stability, Design and Performance Issues," *Proc. of 25th IEEE Conference on Decision and Control*, Athens, Greece, pp 80-85, Dec. 1986.

[16] V. D. Tourassis and C. P. Neuman, ''Properties and Structure of Dynamic Robot Models for Control Engineering Applications,'' *Mechanisms and Machine Theory*, vol. 20 no. 1, pp. 27-40, 1985.

[17] Chi-Haur Wu and R. P. Paul, "Resolved Motion Force Control of Robot Manipulator,'' *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-12, no. 3, pp. 266-275, June 1982.

Appendix

Derivation of Relation of Cartesian Space Error and Modeling Error for Redundant Manipulators

The equations of motions of an n-jointed redundant manipulator in the Cartesian space can be written as [8]

$$\mathbf{\Lambda}_r(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}_r(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}_r(\mathbf{x}) = \mathbf{F} \tag{A.1}$$

where

$$\mathbf{x} \in R^{m\times 1} \tag{A.2}$$
$$\mathbf{\Lambda}_r(\mathbf{x}) = [\,\mathbf{J}(\mathbf{q})\,\mathbf{D}^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,]^{-1} \in R^{m\times m} \tag{A.3}$$
$$\boldsymbol{\mu}_r(\mathbf{x},\dot{\mathbf{x}}) = \bar{\mathbf{J}}^T(\mathbf{q})\,\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{\Lambda}_r(\mathbf{x})\,\dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \in R^{m\times 1} \tag{A.4}$$
$$\mathbf{p}_r(\mathbf{x}) = \bar{\mathbf{J}}^T(\mathbf{q})\,\mathbf{g}(\mathbf{q}) \in R^{m\times 1} \tag{A.5}$$
$$\bar{\mathbf{J}}(\mathbf{q}) = \mathbf{D}^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\mathbf{\Lambda}_r(\mathbf{x}) \in R^{n\times m} \tag{A.6}$$
$$\text{rank}\,[\,\mathbf{J}(\mathbf{q})\,] = m \tag{A.7}$$
$$\mathbf{J}(\mathbf{q}) \in R^{m\times n} \quad (\, n > m \,) \tag{A.8}$$

($\bar{\mathbf{J}}(\mathbf{q})$ is actually a generalized inverse of the Jacobian matrix.)

In the same way as the nonredundant, select

$$\mathbf{F}_c(t) = k_f\ \mathbf{\Lambda}_{rc}(\mathbf{x})\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\ \dot{\mathbf{e}}(t) + \mathbf{K}_p\ \mathbf{e}(t)\,) + \mathbf{\Lambda}_{rc}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}_{rc}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}_{rc}(\mathbf{x}) \tag{A.9}$$

where $\mathbf{\Lambda}_{rc}(\mathbf{x})$, $\boldsymbol{\mu}_{rc}(\mathbf{x})$, and $\mathbf{p}_{rc}(\mathbf{x})$ represent the calculated value of $\mathbf{\Lambda}_r(\mathbf{x})$, $\boldsymbol{\mu}_r(\mathbf{x},\dot{\mathbf{x}})$ and $\mathbf{p}_r(\mathbf{x})$ respectively. Then a torque which generates F_c is

$$\begin{aligned}\mathbf{\Gamma}_c = \mathbf{J}^T(\mathbf{q})\,\mathbf{F}_c &= k_f\ \mathbf{J}^T(\mathbf{q})\,\mathbf{\Lambda}_{rc}(\mathbf{x})\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\ \dot{\mathbf{e}}(t) + \mathbf{K}_p\ \mathbf{e}(t)\,)\\ &\quad + \mathbf{J}^T(\mathbf{q})\,(\,\mathbf{\Lambda}_{rc}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}_{rc}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}_{rc}(\mathbf{x})\,)\end{aligned} \tag{A.10}$$

Then substituting the computed values from (A.10) into (1), we have

$$\begin{aligned}\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) &= k_f\ \mathbf{J}^T(\mathbf{q})\,\mathbf{\Lambda}_{rc}(\mathbf{x})\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\ \dot{\mathbf{e}}(t) + \mathbf{K}_p\ \mathbf{e}(t)\,)\\ &\quad + \mathbf{J}^T(\mathbf{q})\,(\,\mathbf{\Lambda}_{rc}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}_{rc}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}_{rc}(\mathbf{x})\,)\end{aligned} \tag{A.11}$$

Multiplying each side by $\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})$,

$$\begin{aligned}&\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,[\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})]\\ &= k_f\ \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\ \mathbf{J}^T(\mathbf{q})\,\mathbf{\Lambda}_{rc}(\mathbf{x})\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\ \dot{\mathbf{e}}(t) + \mathbf{K}_p\ \mathbf{e}(t)\,)\\ &\quad + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,(\,\mathbf{\Lambda}_{rc}(\mathbf{x})\ddot{\mathbf{x}} + \boldsymbol{\mu}_{rc}(\mathbf{x},\dot{\mathbf{x}}) + \mathbf{p}_{rc}(\mathbf{x})\,)\end{aligned} \tag{A.12}$$

From (A.1)-(A.6),

$$\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\Lambda_{rc}(\mathbf{x}) = \mathbf{J}(\mathbf{q})\bar{\mathbf{J}}(\mathbf{q}) = \mathbf{I}_{m\times m} \qquad \text{(A.13)}$$

and

$$\begin{aligned}
&\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\mu_{rc}(\mathbf{x},\dot{\mathbf{x}}) \\
&= \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\bar{\mathbf{J}}^T(\mathbf{q})\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\Lambda_{rc}(\mathbf{x})\,\dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\
&= \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,[\,\Lambda_{rc}(\mathbf{x})\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,]\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\
&= [\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\Lambda_{rc}(\mathbf{x})\,]\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \\
&= \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \qquad \text{(A.14)}
\end{aligned}$$

Similarly,

$$\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{J}^T(\mathbf{q})\,\mathbf{p}_c(\mathbf{x}) = \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{g}(\mathbf{q}) \qquad \text{(A.15)}$$

Using (61), (A.13), (A.14) and (A.15), (A.12) becomes

$$\begin{aligned}
&\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,[\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})] \\
&= k_f\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\,\dot{\mathbf{e}}(t) + \mathbf{K}_p\,\mathbf{e}(t)\,) + \ddot{\mathbf{x}} - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{g}_c(\mathbf{q}) \\
&= k_f\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\,\dot{\mathbf{e}}(t) + \mathbf{K}_p\,\mathbf{e}(t)\,) + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,\mathbf{g}_c(\mathbf{q}) \\
&= k_f\,(\,\ddot{\mathbf{e}}(t) + \mathbf{K}_v\,\dot{\mathbf{e}}(t) + \mathbf{K}_p\,\mathbf{e}(t)\,) \\
&\quad + \mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,(\,\mathbf{D}_c(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}_c(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}_c(\mathbf{q})\,) \qquad \text{(A.16)}
\end{aligned}$$

Finally, we obtain the same formula as the nonredundant manipulator, that is,

$$\ddot{\mathbf{e}} + \mathbf{K}_v\,\dot{\mathbf{e}} + \mathbf{K}_p\,\mathbf{e} = \frac{1}{k_f}\,\mathbf{J}(\mathbf{q})\,\mathbf{D}_c^{-1}(\mathbf{q})\,(\,\Delta\mathbf{D}(\mathbf{q})\,\ddot{\mathbf{q}} + \Delta\mathbf{H}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\mathbf{g}(\mathbf{q})\,) \qquad \text{(A.17)}$$

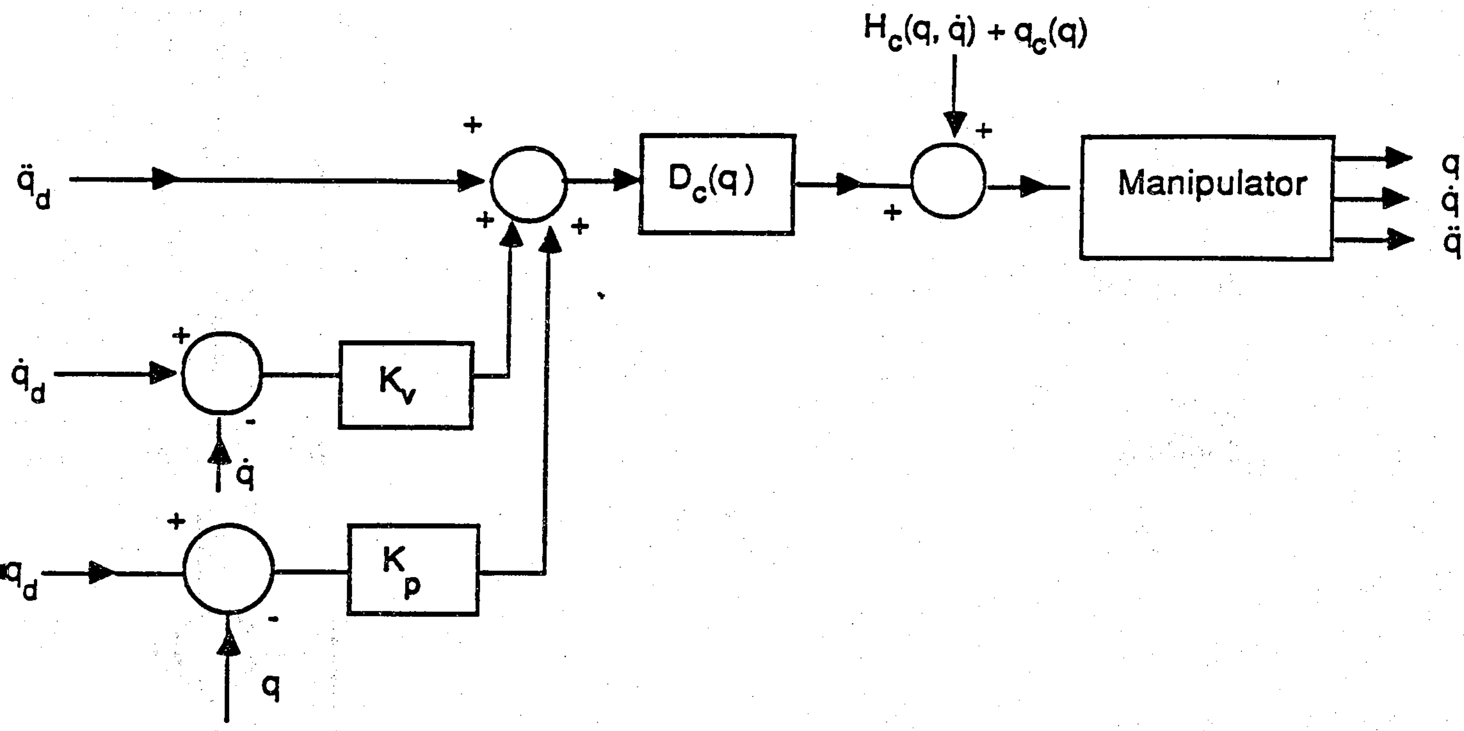

Figure 1. Nonlinear decoupled controller in joint-variable space.

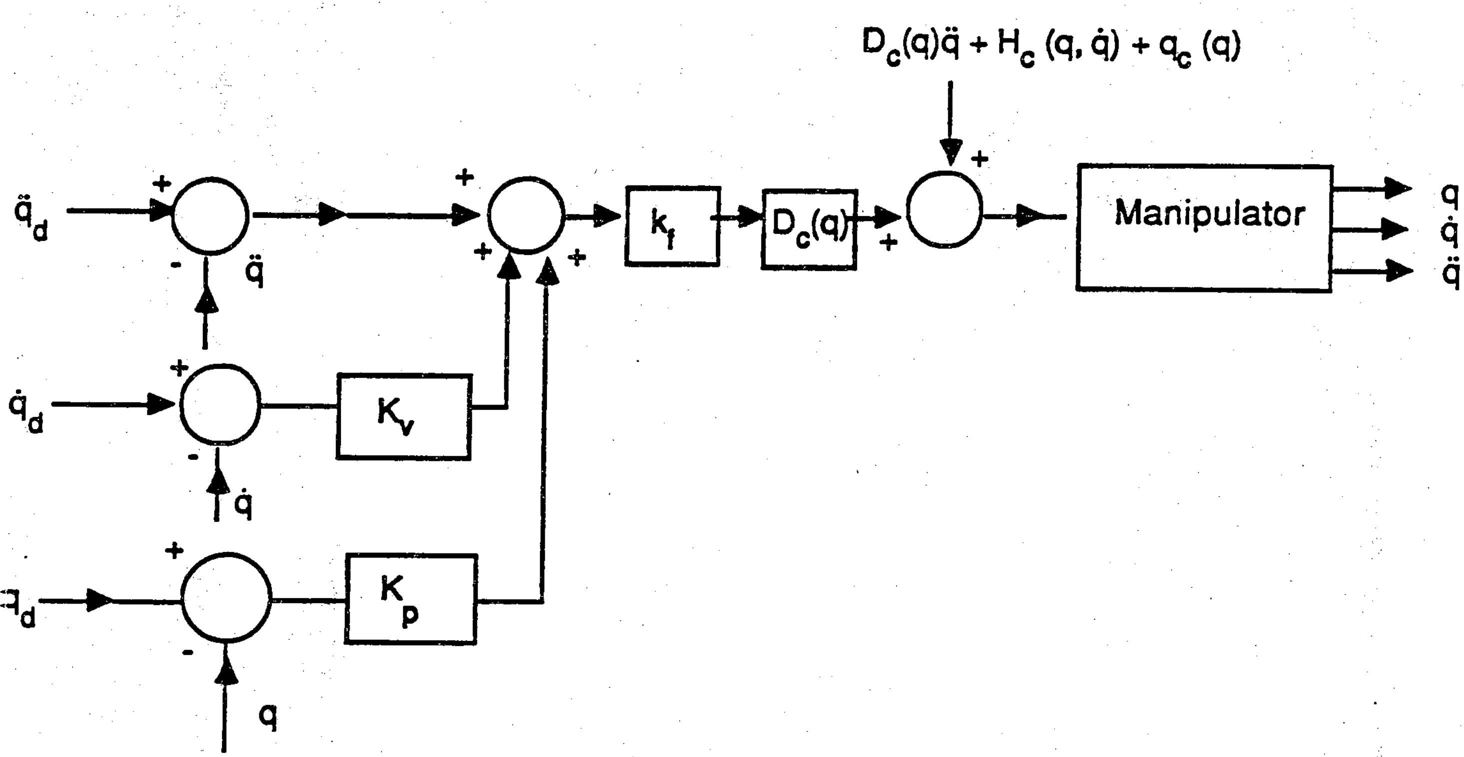

Figure 2. Nonlinear decoupled controller with a high feedback gain.

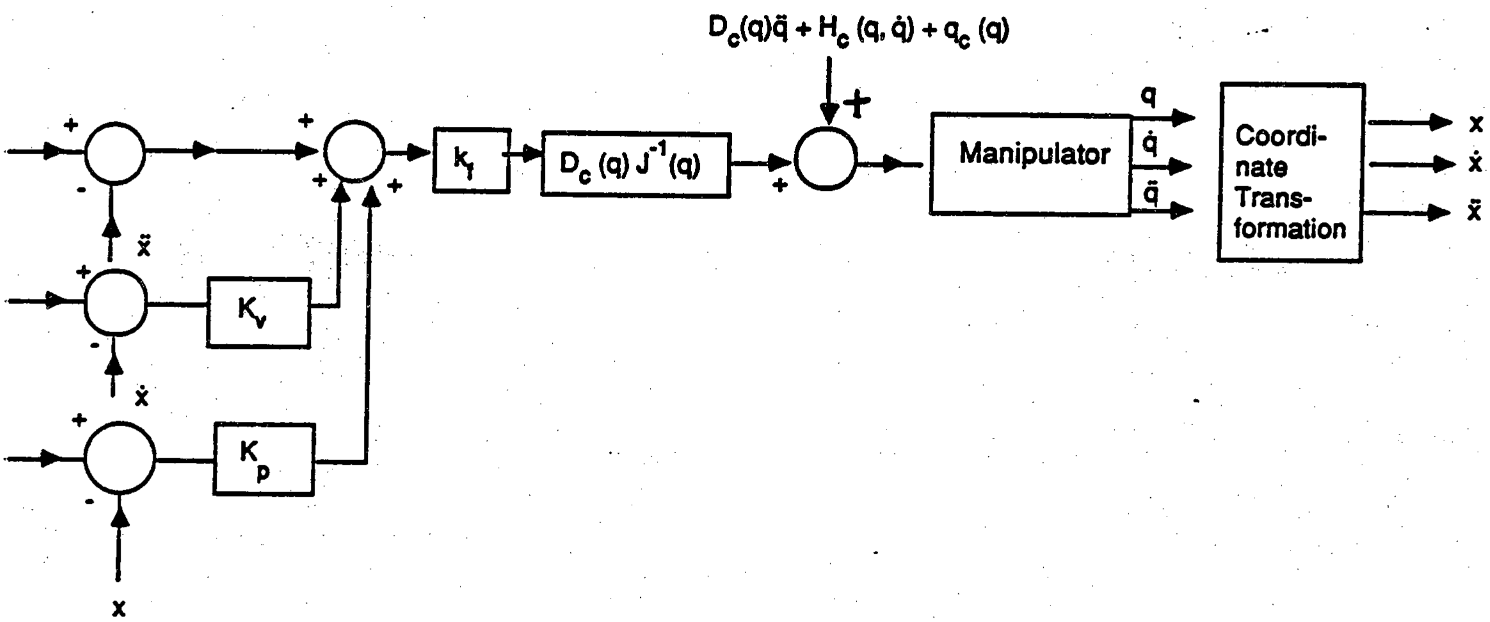

Figure 3. Nonlinear decoupled controller in the Cartesian space with a high feedback gain.

Table 1. Some Parameters and Values for

Evaluating Maximum Error Tolerance

Joint (number)	J_0^i $(kg \cdot m)$	ω_0^i (rad/sec)	$\lvert \dot{q}_i \rvert_{max}$ (rad/sec)	$\lvert \ddot{q}_i \rvert_{max}$ (rad/sec^2)
1	5	25.133	4	4.9
2	5	37.699	2	12.16
3	7	125.664	1‡	5.63‡
4	0.1	94.248	5	37.07
5	0.1	94.248	5	30.61
6	0.04	125.664	8	64.5

‡ The joint velocity and acceleration for joint 3 are in m/sec and m/sec^2, respectively.

Table 2. Maximum Steady-State Errors in the Joint and Cartesian Spaces

Maximum steady-state error in the joint-variable space
ε_{ssp} in rotational joint = 0.01 (degree)
ε_{ssp} in translational joint = 1.0×10^{-5} (meter)
ε_{ssv} in rotational joint = 1 (degree per second)
ε_{ssv} in translational joint = 1.0×10^{-3} (meter/sec)
ε_{ssa} in rotational joint = 1 (degree/s^2)
ε_{ssa} in translational joint = 1.0×10^{-3} (meter/s^2)

Corresponding Maximum Steady-State Errors in the Cartesian Space
ε_{ssp} along x-axis = 3.61800×10^{-4} (meter)
ε_{ssp} along y-axis = 3.51176×10^{-4} (meter)
ε_{ssp} along z-axis = 3.58713×10^{-4} (meter)
ε_{ssp} about x-axis = 4.89395×10^{-4} (rad)
ε_{ssp} about y-axis = 4.89401×10^{-4} (rad)
ε_{ssp} about z-axis = 5.63476×10^{-4} (rad)
ε_{ssv} along x-axis = 3.61800×10^{-2} (meter/sec)
ε_{ssv} along y-axis = 3.51176×10^{-2} (meter/sec)
ε_{ssv} along z-axis = 3.58713×10^{-2} (meter/sec)
ε_{ssv} about x-axis = 4.89395×10^{-2} (rad/sec)
ε_{ssv} about y-axis = 4.89401×10^{-2} (rad/sec)
ε_{ssv} about z-axis = 5.63476×10^{-2} (rad/sec)
ε_{ssa} along x-axis = 3.61800×10^{-2} (meter/s^2)
ε_{ssa} along y-axis = 3.51176×10^{-2} (meter/s^2)
ε_{ssa} along z-axis = 3.58713×10^{-2} (meter/s^2)
ε_{ssa} about x-axis = 4.89395×10^{-2} (rad/s^2)
ε_{ssa} about y-axis = 4.89401×10^{-2} (rad/s^2)
ε_{ssa} about z-axis = 5.63476×10^{-2} (rad/s^2)

Table 3. **Significant Dynamic Coefficients of the Simplified Dynamic Model of the Stanford Arm under a Nonlinear Decoupled Control in the Joint-Variable Space**

Simplified expression	Minimax error	Maximum error tolerance
$d_{11} = 1.451 + 2.357S_2^2 - 5.076S_2^2q_3 + 6.233S_2^2q_3^2$	0.0477†	0.0981
$d_{22} = 4.671$	0.324†	0.430
$d_{33} = 7.252$	0.000†	0.000
$d_{44} = 0.115$	0.0663†	0.990
$d_{55} = 0.113$	0.000†	0.000
$d_{66} = 0.0203$	0.000†	0.000
$d_{12} = 0.449C_2 - 1.050q_3C_2 + 0.119C_2q_3C_4S_5$	0.0240	0.0840
$d_{13} = -1.0475S_2$	0.113	0.190
$d_{14} = 0.1107S_2q_3S_4S_5 + 0.017C_2S_5^2$	0.0189	0.0207
$d_{15} = -0.1160S_2q_3C_4C_5 + 0.0157S_2S_5$	0.0185	0.0233
$d_{23} = -0.1150S_4S_5$	0.00583	0.098
$d_{24} = 0.1167q_3C_4S_5$	0.00873	0.110
$d_{25} = 0.1150q_3S_4C_5$	0.0160	0.0123
$d_{35} = -0.1150S_5$	0.000	0.182
$g_2 = -27.016S_2 + 63.452S_2q_3 + 1.233C_2S_4S_5$	0.748	0.968
$g_3 = -63.446C_2$	0.000	2.110
$g_4 = 1.127S_2C_4S_5$	0.0572	0.420
$g_5 = 1.127S_2S_4C_5 + 1.127C_2S_5$	4.0×10^{-6}	0.918
$h_{112} = 2.763C_2S_2 - 6.205S_2q_3 + 6.920C_2S_2q_3^2$	0.124	0.258
$h_{113} = -2.726S_2^2 + 6.470S_2^2q_3$	0.090	0.451
$h_{115} = -0.017C_4C_5 + 0.115C_2S_2q_3S_4C_5 - 0.115S_2^2q_3S_5 - 0.0548S_2q_3C_4S_5$	0.0148	0.0170
$h_{122} = -0.449S_2 + 1.052S_2q_3 - 0.108S_2q_3C_4S_5$	0.0242	0.105
$h_{123} = -0.984C_2 + 0.331S_2$	6.0×10^{-6}	0.116
$h_{144} = 0.116S_2q_3C_4S_5$	0.0186	0.021
$h_{145} = 0.00764C_2 + 0.00671S_2 - 0.0132C_2C_4 + 0.0847S_2q_3C_4 + 0.00719S_2S_4 + 0.854S_2q_3S_4$	0.0131	0.0133
$h_{155} = 0.1156S_2q_3C_4S_5$	0.0186	0.0210
$h_{211} = -2.929C_2S_2$	0.793	1.025
$h_{214} = -1.582C_2S_2$	0.434	0.546
$h_{215} = -1.539C_2S_2$	0.448	0.549
$h_{223} = -2.712 + 6.47q_3$	0.0771	1.974
$h_{225} = -0.1139q_3S_5$	0.0597	0.0828
$h_{244} = -0.1167q_3S_4S_5$	0.0088	0.110
$h_{245} = 0.0982q_3C_4 - 0.0884S_4$	0.0086	0.0616
$h_{255} = -0.1149q_3S_4S_5$	0.000	0.110
$h_{311} = -1.623S_2^2q_3$	2.676	4.390

$h_{314} = -0.947S_2^2q_3$	1.245	2.171
$h_{315} = -0.902S_2^2q_3$	1.333	2.264
$h_{322} = -1.664q_3$	2.724	4.367
$h_{324} = -0.980q_3$	1.247	2.172
$h_{325} = -0.941q_3$	1.329	2.265
$h_{355} = -0.1150C_5$	0.000	0.0115
$h_{411} = -0.1161C_2S_2q_3C_4S_5$	0.0227	0.0688
$h_{412} = -1.582C_2S_2$	0.434	1.27
$h_{413} = 0.531S_2^2$	1.761	2.367
$h_{511} = 0.131S_2^2q_3S_5$	0.0647	0.1232
$h_{512} = 1.539C_2S_2$	0.448	1.277
$h_{513} = 0.902S_2^2q_3$	1.333	2.468
$h_{522} = 0.118q_3S_5$	0.00756	0.129
$h_{523} = 0.532$	1.841	2.470

† These minimax errors for d_{ii} are in the sense of relative error.

Table 4. Significant Dynamic Coefficients of the Simplified Dynamic Model of the Stanford Arm under a Nonlinear Decoupled Control in the Cartesian Space

Simplified expression	Minimax error	Maximum error tolerance
$d_{11} = 1.416 + 1.634S_2^2 + 1.332S_2^2q_3^2$	0.222†	0.347
$d_{22} = 4.672$	0.324†	0.544
$d_{33} = 7.252$	0.000†	0.000
$d_{44} = 0.115$	0.0663†	0.874
$d_{55} = 0.113$	0.000†	0.000
$d_{66} = 0.0203$	0.000†	0.000
$d_{12} = 0.507C_2 - 1.105q_3C_2$	0.115	0.297
$d_{13} = -1.0475S_2$	0.113	0.672
$d_{14} = 0.130S_2q_3S_4S_5$	0.0354	0.0627
$d_{15} = -0.1170S_2q_3C_4C_5$	0.0158	0.0170
$d_{23} = -0.1150S_4S_5$	0.00583	0.0781
$d_{24} = 0.1167q_3C_4S_5$	0.00873	0.0640
$d_{25} = 0.1150q_3S_4C_5$	0.0160	0.0674
$d_{35} = -0.1150S_5$	0.000	0.0473
$g_2 = -27.392S_2 + 63.453S_2q_3$	1.125	1.310
$g_3 = -63.446C_2$	0.000	1.660
$g_4 = 1.129S_2C_4S_5$	0.0572	0.0886
$g_5 = 1.127S_2S_4C_5 + 1.127C_2S_5$	4.0×10^{-6}	0.0901
$h_{112} = 1.677C_2S_2 + 1.914C_2S_2q_3^2$	0.534	0.913
$h_{113} = -2.726S_2^2 + 6.470S_2^2q_3$	0.0903	1.593
$h_{115} = 0.0797C_2S_2q_3S_4C_5 - 0.112S_2^2q_3S_5 + 0.0288S_2^2q_3C_4S_5$	0.0417	0.0600
$h_{122} = -0.551S_2 + 1.149S_2q_3$	0.122	0.371
$h_{123} = -0.984C_2$	0.331	0.410
$h_{144} = 0.116S_2q_3C_4S_5$	0.0188	0.0740
$h_{145} = 0.0747S_2q_3C_4 + 0.105S_2q_3S_4$	0.0283	0.0468
$h_{155} = 0.1156S_2q_3C_4S_5$	0.0186	0.0743
$h_{211} = -5.503C_2S_2q_3^2$	1.230	1.382
$h_{214} = -2.845C_2S_2q_3^2$	0.615	0.691
$h_{215} = -2.735C_2S_2q_3^2$	0.618	0.695
$h_{223} = -2.712 + 6.47q_3$	0.0771	2.500
$h_{225} = -0.1139q_3S_5$	0.0596	0.105
$h_{244} = -0.1167q_3S_4S_5$	0.0088	0.131
$h_{245} = 0.0982q_3C_4 - 0.0884q_3S_4$	0.00864	0.0780
$h_{255} = -0.1149q_3S_4S_5$	0.000	0.0131
$h_{311} = -1.623S_2^2q_3$	2.676	3.4520
$h_{314} = -0.947S_2^2q_3$	1.2454	1.607
$h_{315} = -0.902S_2^2q_3$	1.333	1.676

$h_{322} = -1.664q_3$	2.724	4.057
$h_{324} = -0.980q_3$	1.247	1.610
$h_{325} = -0.941q_3$	1.329	1.677
$h_{355} = -0.1150C_5$	0.000	0.146
$h_{411} = -0.1161C_2S_2q_3C_4S_5$	0.0227	0.0450
$h_{412} = 1.582C_2S_2$	0.434	0.508
$h_{413} = -1.318S_2^2 + 3.235S_2^2q_3$	0.0454	0.967
$h_{511} = +0.131S_2^2q_3S_5$	0.0647	0.076
$h_{512} = 1.539C_2S_2$	0.448	0.482
$h_{513} = -1.292S_2^2 + 3.235S_2^2q_3$	0.0827	0.910
$h_{522} = 0.118q_3S_5$	0.00756	0.0643
$h_{523} = -1.294 + 3.235q_3$	0.0813	0.910

† These minimax errors for d_{ii} are in the sense of relative error.

Table 5. Computational Complexity of Computing Joint Torques in a Stanford Arm.

	Multiplication	Addition
Parallel Computation ([10])	213	200
Simplified Model 1 (Table 3)	254	99
Simplified Model 2 (Table 4)	238	89