Proceedings of the 26th Conference
on Decision and Control
Los Angeles, CA • December 1987

SIMPLIFICATION OF ROBOT DYNAMIC MODEL BASED ON SYSTEM PERFORMANCE

P. R. Chang and C. S. G. Lee

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

The dynamic performance of a robot manipulator is directly dependent on the efficiency of the controller and the dynamic model of the robot. This paper addresses the fundamental issue of how much manipulator dynamics information should be included in the manipulator dynamic model for control such that the manipulator will achieve the desired system performance under a proportional-plus-derivative control scheme. An efficient minimax simplification scheme has been developed which automatically generates simplified closed-form manipulator motion equations in symbolic form while maintaining the desired manipulator system performance under a proportional-plus-derivative controller. The scheme involves the identification and selection of basis functions that represent the dynamic coefficients in the dynamic model. These basis functions consist of a linear combination of the product terms of sinusoidal and polynomial functions of the generalized coordinates and form a Chebyshev set on the workspace of the manipulator. A multi-layered decision scheme is developed for selecting significant basis terms in each layer for each dynamic coefficient. The linear combination of these significant basis terms is then utilized to construct each simplified dynamic coefficient based on the minimax approximation technique. A verification of the proposed scheme on a Stanford robot arm is included for discussion.

I. INTRODUCTION

Robot manipulators are highly nonlinear systems and the analysis and design of their motion control strategies require the development of efficient closed-form dynamic equations. Two competing approaches 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 [1]. 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 [2-4]. This is motivated by the growing interest in applying advanced control theory to robot manipulators [5-7]. 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 schemes/algorithms for obtaining simplified dynamic models that reveal the dominant dynamics without introducing significant errors into the dynamic model is essential for the advanced control of manipulator systems.

Previous work argue specifically from the viewpoint of obtaining the simplified dynamic model as compared to the complete Lagrange-Euler (LE) equations of motion [2,8-12]. Although their analysis, to a certain extent, is correct, a careful examination of the previous work reveals a major deficiency in their approaches. All of them did not analyze the effects of simplified models on the manipulator system performance (i.e. path/trajectory tracking of the controlled system [4]). A five-percent force/torque error from a simplified manipulator dynamic model usually does not translate to mean that the linear/angular position error of the manipulator will be confined to five percent because of the nonlinear function relating the generalized forces/torques to the generalized coordinates. Thus, simplification schemes for manipulator dynamics must utilize the desired manipulator system performance information to determine the significance of each term in the dynamic model.

This paper presents an efficient minimax decision scheme for automatic generation of a simplified manipulator dynamic model based on the desired manipulator system performance under a proportional-plus-derivative controller (PD controller). In other words, we want to study how much manipulator dynamic model information is needed in a PD controller in order to achieve the desired manipulator system performance. The automatic simplification procedure is a multi-layered decision scheme for fitting a specified dynamic coefficient with a selected set of candidate basis functions involved in the dynamic coefficient. The basis functions consist of a linear combination of the product terms of sinusoidal and polynomial functions of the generalized coordinates and form a Chebyshev set on the workspace of the manipulator. The linear combination of these significant basis terms are then utilized to construct each simplified dynamic coefficient based on the minimax approximation technique. This layer-structured algorithm is very efficient and can be implemented in a "C" program applicable to various manipulators given their kinematics and dynamics data.

II. REDUCTION RULES AND BASIS FUNCTIONS

Based on the Lagrangian formulation and assuming rigid-body motion, the complete dynamic equations of an N-link manipulator, excluding the gear friction and backlash, are a set of second-order coupled nonlinear differential equations. In general, the necessary generalized torque $\tau_i(t)$ for driving the ith link of the robot arm can be written as

$$\sum_{j=1}^{N} J_{ij}\ddot{q}_j + \sum_{j=1}^{N}\sum_{k=1}^{N} H_{ijk}\dot{q}_j\dot{q}_k + G_i = \tau_i(t) \quad ; \; i = 1, 2, \cdots, N \tag{1}$$

where q_i is a generalized coordinate, $q_i = \theta_i$, if joint i is rotary; $q_i = d_i$, if joint i is prismatic. J_{ij}, H_{ijk}, and G_i are the dynamic coefficients of the equations of motion and can be expressed as [2-4]

$$J_{ij} = \sum_{k=max(i,j)}^{N} Tr\,(\,\mathbf{U}_{kj}\mathbf{I}_k\mathbf{U}_{ki}^T\,) = \sum_{k=\max(i,j)}^{N} J_{ij}(k) \tag{2}$$

where $J_{ij}(k) = Tr(\,\mathbf{U}_{kj}\mathbf{I}_k\mathbf{U}_{ki}^T\,)$, $max(i,j) \le k \le N$

$$H_{ijk} = \sum_{l=max(i,j,k)}^{N} Tr(\,\mathbf{U}_{ljk}\mathbf{I}_l\mathbf{U}_{li}^T\,) = \sum_{l=max(i,j,k)}^{N} H_{ijk}(l) \tag{3}$$

where $H_{ijk}(l) = Tr(\,\mathbf{U}_{ljk}\mathbf{I}_l\mathbf{U}_{li}^T\,)$, $max(i,j,k) \le l < N$,

$$G_i = \sum_{j=i}^{N}(-\,m_j\mathbf{g}^T\mathbf{U}_{ji}\bar{\mathbf{r}}_j) = \sum_{j=i}^{N}(G_i(j)) \tag{4}$$

where $G_i(j) = -\,m_j\mathbf{g}^T\mathbf{U}_{ji}\bar{\mathbf{r}}_j$, and

$$\mathbf{U}_{pq} = \begin{cases} {}^0\mathbf{A}_{q-1}\mathbf{Q}_q\,{}^{q-1}\mathbf{A}_p & \text{for } \; p \ge q \\ \mathbf{0} & \text{for } \; p < q \end{cases} \tag{5a}$$

$$\mathbf{U}_{pqr} = \begin{cases} {}^0\mathbf{A}_{q-1}\mathbf{Q}_q\,{}^{q-1}\mathbf{A}_{r-1}\mathbf{Q}_r\,{}^{r-1}\mathbf{A}_p & \text{for } \; p \ge r \ge q \\ {}^0\mathbf{A}_{r-1}\mathbf{Q}_r\,{}^{r-1}\mathbf{A}_{q-1}\mathbf{Q}_q\,{}^{q-1}\mathbf{A}_p & \text{for } \; p \ge q \ge r \\ \mathbf{0} & \text{for } \; p < q \;\; or \;\; p < r \end{cases} \tag{5b}$$

$$\mathbf{Q}_i = \begin{bmatrix} 0 & -\lambda_i & 0 & 0 \\ \lambda_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-\lambda \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5c}$$

and

$$\lambda_i = \begin{cases} 1\,, & \text{if joint } i \text{ is rotational} \\ 0\,, & \text{if joint } i \text{ is translational} \end{cases} \tag{5d}$$

where ${}^0\mathbf{A}_i = {}^0\mathbf{A}_1\,{}^1\mathbf{A}_2 \cdots {}^{i-1}\mathbf{A}_i$ is the 4×4 homogeneous link transformation matrix which relates the spatial location between the ith and the base coordinate frames, $\mathbf{I}_i$ is the pseudo-inertia matrix of link i about the ith coordinate frame, m_j is the mass for link j, $\bar{\mathbf{r}}_i$ is the position of the center of mass of link i with respect to the ith coordinate system, $\mathbf{g}^T = (g_x, g_y, g_z, 0)$ is the gravity vector expressed in the base coordinate system and $|\mathbf{g}| = 9.8062 m/s^2$, and the superscript T on vectors and matrices indicates

This work was supported in part by the NSF-Purdue Engineering Research Center under Grant CDR8500022.

the transpose operation.

Bejczy indicates that the expansion of these dynamic coefficients is tedious and lengthy [2]. Without any simplification, the computational complexity of computing the joint forces/torques from (1) is of order $O(N^4)$. The most common simplification procedure for obtaining simplified dynamic models is to expand the non-zero dynamic coefficients J_{ij}, H_{ijk}, and G_i symbolically and neglect insignificant terms. For simplifying a specific dynamic coefficient, an exhaustive term-by-term evaluation is carried out so that only the most significant terms are retained [8-10]. Another approach is to utilize useful physical and matrix algebra properties for reducing the complexity of the dynamic coefficients. Some of these reduction rules are general and they apply to any manipulator irrespective of the manipulator's geometric configuration; others depend on the particular manipulator's geometric configuration. These reduction properties include two basic concepts [13]: (1) elimination of the redundant dynamic coefficients or information and (2) identification of zero dynamic coefficient terms. In addition to these reduction rules, some dynamic coefficients may vanish for particular kinematic link parameters; however, there are no general rules to identify these identical zero dynamic coefficients. To reduce the cost of evaluating these configuration-dependent dynamic coefficients, a better approach is to find basis functions of each dynamic coefficient and the minimax curve fitting technique can be used to determine each objective dynamic coefficient which is expressed as a linear combination of the basis functions.

It can be shown that $\mathbf{U}_{ij}$ and $\mathbf{U}_{li}$ (in (5a)) when $\max(i,j) \le l \le N$ consist of either $\cos q_m$ and $\sin q_m$ (for a rotational joint) or q_m (for a translational joint) for $1 \le m \le l$. This is also true for $\mathbf{U}_{ljk}$ (in (5b)) when $\max(i,j,k) \le l \le N$. Thus, from (2) (or (3)), one can find that the expansion of $J_{ij}(l)$ (or $H_{ijk}(l)$) can be expressed in the form of sum of products. If joint m is translational, the expanded product terms will consist of q_m or q_m^2. If joint m is rotational, the expanded product terms will consist of $\cos q_m$ or $\sin q_m$ or $\cos^2 q_m$ or $\sin^2 q_m$ or $\cos q_m \sin q_m$. Among these terms, $\cos^2 q_m$ can be expressed as $(1-\sin^2 q_m)$. Thus, the set of candidate basis functions for J_{ij} (or H_{ijk}) is a subset of the product terms in the expansion of

$$\prod_{\substack{\text{all rotational joint } m_r \\ \text{and } m_r \le N}} (1 + Cm_r + Sm_r + (Sm_r)^2 + Cm_r\, Sm_r) \times \prod_{\substack{\text{all translational joint } m_t \\ \text{and } m_t \le N}} (1 + q_{m_t} + q_{m_t}^2) \tag{6}$$

where $Cm_r = \cos q_{m_r}$ and $Sm_r = \sin q_{m_r}$. Similarly, it can be shown that the basis functions of G_i in (4) will be a subset of the product terms in the expansion of

$$\prod_{\substack{\text{all rotational joint } m_r \\ \text{and } m_r \le N}} (1 + Cm_r + Sm_r) \prod_{\substack{\text{all translational joint } m_t \\ \text{and } m_t \le N}} (1 + q_{m_t}) \tag{7}$$

where $\mathbf{q}^T = [q_1, q_2, \cdots, q_N] \in Q$, and

$$Q = \begin{cases} \mathbf{q} \mid q_i \in [Q_i^{\min}, Q_i^{\max}] & \text{, if joint } i \text{ is rotational} \\ \mathbf{q} \mid q_i \in [d_i^{\min}, d_i^{\max}] & \text{, if joint } i \text{ is translational} \end{cases}$$

where $|Q_i^{\max} - Q_i^{\min}| < 2\pi$ and $d_i^{\max}, d_i^{\min} \ge 0$. Equation (6) shows that J_{ij} and H_{ijk} contain the second-harmonic sinusoidal terms for rotational joints and the power-two terms of q_{m_t} for translational joint and (7) shows that G_i contains fundamental harmonic sinusoidal terms for rotational joints and power-one terms of q_{m_t} for translational joints. It is easy to show that the expanded product terms in (6) and (7) are linear independent and form a set of candidate basis functions for J_{ij} and H_{ijk}, and G_i, respectively, and whose behavior is similar to multi-dimensional Fourier series.

Lemma 1: The expanded product terms in (6) and (7) are linear independent and form a set of candidate basis functions for the corresponding dynamic coefficients of J_{ij} and H_{ijk}, and G_i, respectively.

Proof: Because the elements in $\{1, C_i, S_i, S_i^2, C_iS_i\}$ or $\{1, q_i, q_i^2\}$ are linear independent, thus, one can prove that the product terms of (6) or (7) are also linear independent by mathematical induction.

Looking at (6) and (7), one finds that the maximum number of possible terms in the set of candidate basis functions is 5^N for J_{ij} or H_{ijk}, and 3^N for G_i for manipulators with rotary joints. Because of manipulator's particular dynamics characteristics, a more compact set of candidate basis functions for J_{ij}, H_{ij}, and G_i may be possible. Tourassis and Neuman [13] pointed out several useful dynamics characteristics and functional dependence properties of the dynamic coefficients which are useful tools for obtaining a more compact set of candidate basis functions. From these sets of candidate basis functions, each dynamic coefficient can be expressed as a linear combination of these basis functions and the minimax curve fitting technique can be used to determine the unknown coefficients of the basis functions within the workspace of the manipulator.

All the above techniques yield simplified dynamic models in which the computed torque for each actuator does not contribute significant error as compared to the complete dynamic model. However, due to the inherent nonlinearity of the dynamic model, a five-percent error in the torque computation does not directly translate to a five-percent error in the angular position error of the joint. Thus, the effects of a simplified dynamic model on the manipulator system performance must be investigated and analyzed and the generation of a simplified dynamic model must be based on the specification of the desired system performance. Hence, this concept of using the desired manipulator system performance specifications to determine the unknown coefficients of the basis functions on the workspace of the manipulator is the main contribution of this paper.

III. SPECIFICATION OF MANIPULATOR SYSTEM PERFORMANCE

In this section, system performance of a manipulator system under a PD control will be investigated and an analytical relationship between the dynamic coefficients and the parameters for specifying the performance of the manipulator system will be derived. The results will be useful in selecting significant basis functions for obtaining a simplified manipulator dynamic model to achieve the desired system performance.

Each joint subsystem of the manipulator is modeled as a single-input single-output, second-order dc motor system. The coupling effects coming from other joints are considered as disturbances to the joint control system. A conventional PD control is utilized to servo the system to track a desired trajectory [4]. In addition to the PD controller, feedforward torques are computed to compensate the coupling effects between joints in order to produce good control performance. This control scheme is quite similar to the so-called "inverse problem" technique or "computed torque" technique [4] and is shown in Fig. 1. We shall use this control scheme to investigate the influence of the dynamic performance under the effects of a simplified dynamic model.

In Fig. 1, $J_{ij}^c(\mathbf{q})$, f_i^c, $H_{ijm}^c(\mathbf{q})$, $G_i^c(\mathbf{q})$, and V_i^c are, respectively, the computed counterparts of the actual effective inertia terms $J_{ij}(\mathbf{q})$, effective damping coefficient f_i, effective velocity-related terms $H_{ijk}(\mathbf{q})$, effective gravity term $G_i(\mathbf{q})$, and Coulomb friction V_i at the actuator shaft of joint i of the manipulator. n^i is the gear ratio, armature resistance R_a^i, motor constant K_a^i, and back electromagnetic force constant K_b^i are known dc motor parameters for joint i. K_p^i and K_v^i are, respectively, position and velocity feedback gains of the controller. Their values are chosen subject to the mechanical resonant frequency constraint of the manipulator [3,4]. Using the Mason's gain formula and the superposition concept, the closed-loop input-output equation including the disturbances in the Laplace transform domain can be written as

$$\Theta_i(s) = \frac{R_a^i J_{ii}^c s^2 + \left(R_a^i f_i^c + K_a^i K_b^i + K_a^i K_v^i\right)s + K_p^i K_a^i}{R_a^i J_{ii} s^2 + \left(R_a^i f_i + K_a^i K_b^i + K_a^i K_v^i\right)s + K_p^i K_a^i}\, \Theta_i^d(s) + \frac{n^i R_a^i \left(T_D^i(s) - T_D^{ic}(s)\right)}{R_a^i J_{ii} s^2 + \left(R_a^i f_i + K_a^i K_b^i + K_a^i K_v^i\right)s + K_p^i K_a^i} \tag{8}$$

where $\Theta_i^d(s)$ and $\Theta_i(s)$ are, respectively, the desired and actual angular displacement of joint i, $T_D^i(s)$ is the disturbance in Laplace transform and in the time domain, it is equal to $\sum_{j\ne i} J_{ij}(\mathbf{q})\,\ddot{q}_j + \sum_j \sum_k H_{ijk}\dot{q}_j\dot{q}_k + G_i(\mathbf{q}) + V_i$, $T_D^{ic}(s)$ is the computed feedforward torque to compensate for the disturbance and in the time domain, it is equal to $\sum_{j\ne i} J_{ij}^c(\mathbf{q})\ddot{q}_j + \sum_j \sum_k H_{ijk}^c\, \dot{q}_j \dot{q}_k + G_i^c(\mathbf{q}) + V_i^c$. In order not to excite the mechanical resonant frequency of the manipulator, the undamped natural frequency is set to no more than one-half of the structural resonant frequency [3]. Complying to this constraint, one can obtain the following relation and the upper bound of K_p^i [4]

$$0 \le K_p^i \le \frac{J_0^i(\omega_0^i)^2 R_a^i}{4\,K_a^i} \tag{9}$$

where ω_0^i and J_0^i are, respectively, the measured structural resonant frequency and inertia of joint i at a known location. Relating the characteristic equation of (8) to the standard notation for a second-order system, the damping coefficient ς_i of the system in (8) can be obtained as [4]

$$\varsigma_i = \frac{R_a^i f_i + K_a^i K_b^i + K_a^i K_v^i}{2\sqrt{K_p^i K_a^i R_a^i J_{ii}}} \ . \tag{10}$$

In order to avoid the oscillatory underdamped response, ς_i is set to ≥ 1. Thus, K_v^i becomes

$$K_v^i \ge \frac{2\sqrt{K_a^i K_p^i R_a^i J_{ii}} - R_a^i f_i - K_a^i K_b^i}{K_a^i} \ . \tag{11}$$

The equality of (11) yields a critically-damped system response while the inequality yields an overdamped system response. Because the actual J_{ii} and f_i are usually not available, we use their computed counterparts J_{ii}^c and f_i^c from the manipulator dynamic model instead of the actual J_{ii} and f_i. Substituting

$$K_p^i = \frac{J_0^i(\omega_0^i)^2 R_a^i}{4\,K_a^i} \tag{12}$$

into the equality of (11), we obtain the velocity feedback gain K_v^i for achieving a critically-damped system response

$$K_v^i = \frac{R_a^i \omega_0^i \sqrt{J_0^i J_{ii}^c} - R_a^i f_i^c - K_a^i K_b^i}{K_a^i} \quad . \tag{13}$$

Since the computed J_{ii}^c and f_i^c can not be found exactly equal to the actual J_{ii} and f_i due to the modelling error, if we substitute (13) into (10), we obtain the damping coefficient ς_i'

$$\varsigma_i' = \sqrt{\frac{J_{ii}^c(\mathbf{q})}{J_{ii}(\mathbf{q})}} + \frac{(f_i - f_i^c)}{\omega_0^i \sqrt{J_0^i J_{ii}}} \quad . \tag{14}$$

In (14), the damping coefficient f_i, usually a measurable constant value, is a function of joint rate and independent of the arm configuration. The computed f_i^c can be found experimentally to be near the actual value of f_i and the term $(f_i - f_i^c)/\omega_0^i \sqrt{J_0^i J_{ii}}$ will become negligible. Thus, $\varsigma_i' \approx \sqrt{J_{ii}^c(\mathbf{q})/J_{ii}(\mathbf{q})}$ and $J_{ii}(\mathbf{q}) = J_{ii}^c(\mathbf{q}) + \Delta J_{ii}(\mathbf{q})$, where $\Delta J_{ii}(\mathbf{q})$ is the modelling error of the inertia term. The first-order approximation of ς_i' can be found to be

$$\varsigma_i' \approx \sqrt{1 - \Delta J_{ii}(\mathbf{q})/J_{ii}(\mathbf{q})} \approx 1 - \frac{1}{2}\left[\Delta J_{ii}(\mathbf{q})/J_{ii}(\mathbf{q})\right] \tag{15}$$

Assuming $-0.1 \le \Delta J_{ii}(\mathbf{q})/J_{ii}(\mathbf{q}) \le 0.1$, that is $\max_{\mathbf{q}} \left| \frac{\Delta J_{ii}(\mathbf{q})}{J_{ii}(\mathbf{q})} \right| \le 0.1$, the damping coefficient ς_i' will be within the range $0.95 \sim 1.05$. This shows that the system will be near critically-damped when the computed inertia J_{ii}^c has a maximum relative tolerance error of 10 percent. In other words, a 10 percent modelling error in inertia information is the maximum tolerance in achieving a satisfactory critically-damped response [15].

In addition to the damping coefficient ς_i which characterizes the transient response of a second-order system, other performance measures of the controller such as steady-state errors due to some standard test signals (e.g. step, ramp, parabolic functions) must be investigated for the simplified dynamic model. These three standard test signals approximate most command inputs (e.g. trajectory inputs) and provide a means for comparing the performance of the steady-state behavior of different control systems [15].

To facilitate the analysis of steady-state errors based on the test signals, a system error equation in Laplace transform can be obtained from (8)

$$E^i(s) \triangleq \Theta_i^d(s) - \Theta_i(s) \tag{16}$$

$$= \frac{R_a^i(\Delta J_{ii} s^2 + \Delta f_i s)}{\Omega^i(s)} \Theta_i^d(s) + \frac{n^i R_a^i (T_D^{ic}(s) - T_D^i(s))}{\Omega^i(s)}$$

where $\Delta J_{ii} = J_{ii} - J_{ii}^c$, $\Delta f_i = f_i - f_i^c$, and $\Omega^i(s) = R_a^i J_{ii} s^2 + (R_a^i f_i + K_a^i K_b^i + K_a^i K_v^i)s + K_p^i K_a^i$.

If the command input to each joint is a unit step, then the velocity-related and acceleration-related terms of the disturbance torque $T_D^i(s)$ and the computed counterpart $T_D^{ic}(s)$ for joint i will vanish as $t \to \infty$. The uncompensated disturbance $(T_D^{ic}(s) - T_D^i(s))$ will become $\int_0^\infty (G_i^c(\mathbf{q}(t)) - G_i(\mathbf{q}(t)))\, e^{-st} dt$ at steady state. Taking the absolute value of the uncompensated disturbance, we have

$$| T_D^{ic}(s) - T_D^i(s) | = | \int_0^\infty [G_i^c(\mathbf{q}) - G_i(\mathbf{q})]\, e^{-st}]dt \,| \tag{17}$$

$$\le \int_0^\infty | G_i^c(\mathbf{q}) - G_i(\mathbf{q}) \,| e^{-st} dt \le \max_{\mathbf{q}} | G_i^c(\mathbf{q}) - G_i(\mathbf{q}) \,| /s \quad .$$

Applying the final value theorem to (16) with $\Theta_j^d(s) = 1/s$ (unit step), $j = 1, 2, \cdots, N$, one obtains the steady-state step (or position) error for joint i as

$$e_{ssp}^i \triangleq \lim_{s \to 0} s\, E^i(s) = \frac{n^i R_a^i}{K_p^i K_a^i} \lim_{s \to 0} s\, (T_D^{ic}(s) - T_D^i(s)) \tag{18}$$

and using (17) and (12), its absolute value is

$$|e_{ssp}^i| = \frac{n^i R_a^i}{K_p^i K_a^i} \,| \lim_{s \to 0} s\, (T_D^{ic}(s) - T_D^i(s)) \,| \tag{19}$$

$$\le \frac{n^i R_a^i}{K_p^i K_a^i} \max_{\mathbf{q}} | G_i^c(\mathbf{q}) - G_i(\mathbf{q}) \,| \le \frac{4\, n^i}{J_0^i(\omega_0^i)^2} \max_{\mathbf{q}} | \Delta G_i(\mathbf{q}) \,|$$

where $\Delta G_i(\mathbf{q}) = G_i(\mathbf{q}) - G_i^c(\mathbf{q})$. If there are N steady-state position error specifications ϵ_{ssp}^i for the manipulator, $1 \le i \le N$, then the maximum error tolerance requirement for the computed gravity term for each joint can be determined,

$$\max_{\mathbf{q}} | \Delta G_i(\mathbf{q}) \,| \le \frac{J_0^i(\omega_0^i)^2}{4\, n^i} \epsilon_{ssp}^i \quad . \tag{20}$$

Equation (20) implies that $|e_{ssp}^i| \le \epsilon_{ssp}^i$. So, the value $\epsilon_{ssp}(G_i) = \frac{J_0^i(\omega_0^i)^2}{4\, n^i} \epsilon_{ssp}^i$ is the maximum error tolerance for the computed gravity term G_i^c and it depends on the system performance specification of the steady-state position error ϵ_{ssp}^i.

Similarly, if the command input to each joint is a ramp, after some lengthy manipulation of equations, the maximum error tolerances for $|\Delta G_i|_{\max}$, $|\Delta H_{ijk}|_{\max}$ and $|\Delta V_i|_{\max}$ are, respectively [16],

$$|\Delta G_i|_{\max} \le \frac{|G_i|_{\max}}{\tau_0^i} \frac{J_0^i(\omega_0^i)^2}{4 n^i} \epsilon_{ssv}^i \triangleq \epsilon_{ssv}(G_i) \tag{21a}$$

$$|\Delta H_{ijk}|_{\max} \le \frac{|H_{ijk}|_{\max}}{\tau_0^i} \frac{J_0^i(\omega_0^i)^2}{4 n^i} \epsilon_{ssv}^i \triangleq \epsilon_{ssv}(H_{ijk}) \tag{21b}$$

$$|\Delta V_i|_{\max} \le \frac{|V_i|_{\max}}{\tau_0^i} \frac{J_0^i(\omega_0^i)^2}{4 n^i} \epsilon_{ssv}^i \triangleq \epsilon_{ssv}(V_i) \tag{21c}$$

where $|\Delta G_i|_{\max} = \max_{\mathbf{q}} |G_i(\mathbf{q}) - G_i^c(\mathbf{q})|$, $|\Delta H_{ijk}|_{\max} = \max_{\mathbf{q}} |H_{ijk}(\mathbf{q}) - H_{ijk}^c(\mathbf{q})|$, $|\Delta V_i|_{\max} = \max |V_i - V_i^c|$, $1 \le j, k \le N$ for a fixed i, $\tau_0^i = (|G_i|_{\max} + \sum_j \sum_k |H_{ijk} \dot{q}_j \dot{q}_k|_{\max} + |V_i|_{\max})$, and $\epsilon_{ssv}(G_i)$, $\epsilon_{ssv}(H_{ijk})$, and $\epsilon_{ssv}(V_i)$ are, respectively, the maximum error tolerances for the computed G_i^c, H_{ijk}^c and V_i^c which can be easily computed because the parameters $|G_i|_{\max}$, $|H_{ijk}|_{\max}$, $|V_i|_{\max}$, τ_0^i, J_0^i, ω_0^i, and n^i are all known.

If the command input to each joint is a parabolic input, the same mathematical approach can be used to find the maximum error tolerance for $|\Delta G_i|_{\max}$, $|\Delta H_{ijk}|_{\max}$, and $|\Delta J_{ij}|_{\max} (j \ne i)$, and one can obtain the following [16]:

$$|\Delta G_i|_{\max} \le \frac{|G_i|_{\max}}{\tau_1^i} \bar{\epsilon}_{ssa}^i \triangleq \epsilon_{ssa}(G_i) \tag{22a}$$

$$|\Delta H_{ijk}|_{\max} \le \frac{|H_{ijk}|_{\max}}{\tau_1^i} \bar{\epsilon}_{ssa}^i \triangleq \epsilon_{ssa}(H_{ijk}) \tag{22b}$$

$$|\Delta V_i|_{\max} \le \frac{|V_i|_{\max}}{\tau_1^i} \bar{\epsilon}_{ssa}^i \triangleq \epsilon_{ssa}(V_i) \tag{22c}$$

$$|\Delta J_{ij}|_{\max} \le \frac{|J_{ij}|_{\max}}{\tau_1^i} \bar{\epsilon}_{ssa}^i \triangleq \epsilon_{ssa}(J_{ij}) \tag{22d}$$

where $\left[\frac{J_0^i(\omega_0^i)^2}{4\, n^i} \epsilon_{ssa}^i - \frac{|J_{ii}|_{\max}}{10\, n^i} \right] \triangleq \bar{\epsilon}_{ssa}^i$, $1 \le j, k \le N$ for a fixed i, $1 \le j \le N$, $j \ne i$, and $\tau_1^i = |G_i|_{\max} + \sum_j \sum_k |H_{ijk} \dot{q}_j \dot{q}_k|_{\max} + |V_i|_{\max} + \sum_{j \ne i} |J_{ij} \ddot{q}_j|_{\max}$. Because J_{ij} represents coupling inertia at joint i due to joint j and has a symmetric property, one can show that $|\Delta J_{ij}|_{\max} = |\Delta J_{ji}|_{\max}$. But the maximum error tolerance $\epsilon_{ssa}(J_{ji})$ due to joint j specification is likely to give a cross maximum error tolerance $\epsilon_{ssa}^*(J_{ij}) \triangleq \min(\epsilon_{ssa}(J_{ij}), \epsilon_{ssa}(J_{ji}))$ which can satisfy the sufficient condition if $|\Delta J_{ij}| \le \epsilon_{ssa}^*(J_{ij})$ and $|\Delta J_{ji}| \le \epsilon_{ssa}^*(J_{ij})$, then $|\Delta J_{ij}| \le \epsilon_{ssa}(J_{ij})$ and $|\Delta J_{ji}| \le \epsilon_{ssa}(J_{ji})$. (Note that $\epsilon_{ssa}^*(J_{ij}) = \epsilon_{ssa}^*(J_{ji})$).

In the above derivation for steady-state errors due to step, ramp, and parabolic inputs, one can determine the maximum error tolerances (or *maximum modelling errors*) of the dynamic coefficients that the manipulator control system can still achieve the desired performance specifications (such as $\epsilon_{ssp}^i, \epsilon_{ssv}^i, \epsilon_{ssa}^i$). Since the combination of these three standard test signals can represent most command inputs, the best way to measure the maximum error tolerances for G_i, H_{ijk}, V_i, and J_{ij} can be based on their steady-state error specifications as

$$\epsilon_{G_i} \triangleq \min(\epsilon_{ssp}(G_i), \epsilon_{ssv}(G_i), \epsilon_{ssa}(G_i)) \tag{23a}$$

$$\epsilon_{H_{ijk}} \triangleq \min(\epsilon_{ssv}(H_{ijk}), \epsilon_{ssa}(H_{ijk})) \tag{23b}$$

$$\epsilon_{V_i} \triangleq \min(\epsilon_{ssv}(V_i), \epsilon_{ssa}(V_i)) \tag{23c}$$

$$\epsilon_{J_{ij}} \triangleq \epsilon_{ssa}^*(J_{ij}), \text{ for } j \ne i \quad . \tag{23d}$$

It should be noted that the maximum error tolerance between J_{ii}^c and J_{ii} is a relative value which compares with the actual inertia J_{ii}, while the maximum error tolerance between G_i^c (or H_{ijk}^c or V_i^c or J_{ij}^c) and G_i (or H_{ijk} or V_i or J_{ij}) is an absolute value. The above derivation and analytical expression relating the maximum error tolerances of the dynamic coefficients to the performance specifications of the manipulator control system leads us to an interesting question: Given the desired manipulator control system performance specifications under a PD controller, how can we determine the complexity of the manipulator dynamic model such that the manipulator control system will still achieve the desired performance? Thus, the complexity of the simplified dynamic model depends on the performance specification of the manipulator system. An efficient minimax decision scheme for reducing the cost of obtaining the dynamic coefficients and the simplified dynamic model to satisfy the desired performance specifications will be developed in the next section.

IV. MULTI-LAYERED MINIMAX SIMPLIFICATION SCHEME

Before discussing the simplification procedure, we need to review some properties of the minimax fitting technique as follows:

Definition 1: The set of basis functions $B_i(\mathbf{q})$, $i = 1,2,\cdots,n$, forms a Chebyshev set on a domain Q (i.e. workspace of the robot) if any nontrivial linear combination has at most $(n-1)$ zeros in Q.

Lemma 2: The set of candidate basis functions for any dynamic coefficient which contains the product terms of sinusoidal functions (for rotational joints) and polynomial functions (for translational joints) forms a Chebyshev set on a workspace Q, where $Q \subseteq \{\mathbf{q} \mid \mathbf{q} = [q_1,q_2,\cdots,q_N]^T$, $q_i \in [Q_i^{\min},Q_i^{\max}]$, if joint i is rotational; $q_i \in [d_i^{\min},d_i^{\max}]$, if joint i is translational$\}$, where $|Q_i^{\max} - Q_i^{\min}| < 2\pi$ and $d_i^{\min},d_i^{\max} \geq 0$.

The proof of Lemma 2 for one variable case can be found in [14]. The extension to multivariable may be proved by mathematical induction.

Lemma 3: Linear minimax (or Chebyshev) approximation problem is to find a best approximate $\hat{\phi}(\mathbf{q}) = \sum_{i=1}^{n} \hat{a}_i B_i(\mathbf{q})$ for the objective function $\phi(\mathbf{q})$ (i.e. respective dynamic coefficients of J_{ij}, H_{ijk}, G_i) on a workspace Q which will satisfy the following condition:

$$\max_{\mathbf{q}\in Q} ||E(\hat{a}_i,\mathbf{q})|| \leq \max_{\mathbf{q}\in Q} ||E(a_i,\mathbf{q})|| \quad \text{for any } a_i$$

where $E(a_i,\mathbf{q}) \triangleq \phi(\mathbf{q}) - \sum_{i=1}^{n} a_i\, B_i(\mathbf{q})$.

Lemma 4: The set of basis functions $B_i(\mathbf{q}), i = 1,2,\cdots,n$, possesses a unique best minimax or Chebyshev approximation by a linear combination of the basis functions if and only if the basis functions form a Chebyshev set on a domain Q.

The proof of Lemma 4 can be found in [14].

Theorem 1: The expression of any dynamic coefficient $\phi(\mathbf{q})$ (i.e. $\phi(\mathbf{q}) = J_{ij}$ or H_{ijk} or G_i) can be determined by applying the minimax fitting technique to the set of candidate basis functions in (6) or (7).

The proof of Theorem 1 can be found in [16]. Theorem 1 indicates that, if the set of candidate basis functions for each dynamic coefficient forms a Chebyshev set on the workspace of the manipulator, then the expression of any dynamic coefficient can be determined by the minimax fitting technique [14]. To reduce the cost and computation of evaluating the dynamic coefficients while maintaining desired manipulator control performance, it is possible to find a best minimax approximate model for the dynamic coefficients subject to the maximum error tolerances in (23a-d). The minimax solution, from Lemma 4, is always unique. Prior to introducing the simplification procedure, we shall define three error tolerance basis sets which will be needed for the development of the simplification procedure.

Definition 2: An ϵ-error tolerance basis set $BASIS_\epsilon(Q,\phi) = \{B_i(\mathbf{q})\}_{i=1}^{n}$ for $\phi(\mathbf{q})$ on a workspace Q satisfies the following condition:

$$E_{minimax}(BASIS_\epsilon(Q,\phi)) = \min_{a_i}\max_{\mathbf{q}\in Q} ||\phi(\mathbf{q}) - \sum_{i=1}^{n} a_i B_i(\mathbf{q})|| \leq \epsilon$$

where ϵ is the maximum error tolerance for the corresponding dynamic coefficients in (23a-d).

Definition 3: $BASIS_\epsilon^{SIG}(Q,\phi)$ is called a significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on a workspace Q which satisfies the following conditions:

(i) $BASIS_\epsilon^{SIG}(Q,\phi)$ is an ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on a workspace Q.

(ii) Removing any basis term $B_i(\mathbf{q})$ of $BASIS_\epsilon^{SIG}(Q,\phi)$, i.e., $S_i(Q) \triangleq (BASIS_\epsilon^{SIG}(Q,\phi)/\{B_i\})$ $\triangleq (BASIS_\epsilon^{SIG}(Q,\phi) - BASIS_\epsilon^{SIG}(Q,\phi) \cap \{B_i\})$, for any $B_i \in BASIS_\epsilon^{SIG}(Q,\phi)$, the resulting set $S_i(Q)$ is *not* an ϵ-error tolerance basis set, that is, $E_{minimax}(S_i(Q)) > \epsilon$.

Definition 4: $BASIS_\epsilon^{MSIG}(Q,\phi)$ is called a minimal significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on a workspace Q which satisfies the following conditions:

(i) $BASIS_\epsilon^{MSIG}(Q,\phi)$ is a significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on the workspace Q.

(ii) Minimal number of basis functions, that is, $|(BASIS_\epsilon^{MSIG}(Q,\phi))| \leq |(BASIS_\epsilon^{SIG}(Q,\phi))|$, for any significant ϵ-error tolerance basis set $BASIS_\epsilon^{SIG}(Q,\phi)$, where $|S|$ is the number of elements in the set S.

The minimal significant ϵ-error tolerance basis set $BASIS_\epsilon^{MSIG}(Q,\phi)$ can form the best minimax approximate model $\hat{\phi}(\mathbf{q}) = \sum_{i=1}^{n} \hat{a}_i B_i(\mathbf{q})$ for $\phi(\mathbf{q})$, where $n = |(BASIS_\epsilon^{MSIG}(Q,\phi))|$. The number of basis functions in $BASIS_\epsilon^{MSIG}(Q)$ depends on the workspace Q and the manipulator control system performance specification ϵ. A "small" workspace and a "large" control system specification of ϵ result in a small number of basis functions. A simplification procedure for obtaining the minimal significant ϵ-error tolerance basis set $BASIS_\epsilon^{MSIG}(Q,\phi)$ for $\phi(\mathbf{q})$ subject to a control system specification ϵ and a workspace Q has been developed. The basic idea of the simplification procedure is to construct a best minimax approximate model by selecting minimal significant basis functions from the set of candidate basis functions in (6) or (7). Starting with the set of n candidate basis functions, our approach is to test all possible n basis subsets which are children of the set of candidate basis functions by eliminating a basis term from each subset which contains $(n-1)$ basis functions. If the resulting subset is ϵ-error tolerance, then it becomes a possible set of candidate basis functions for $\phi(\mathbf{q})$ with a less number of basis functions. This in turn may generate $(n-1)$ children and each of its children needs to be tested for ϵ-error tolerance. The procedure is repeated until the children of all the possible sets of candidate basis functions are *not* ϵ-error tolerance. Obviously, this approach is very exhaustive, and to avoid testing all the possible basis function sets, the candidate set can be divided into two groups: the *SURVIVOR* group is a set containing the ϵ-error tolerance basis sets and the *DEAD* group is a set containing the sets that are *not* ϵ-error tolerance. It can be easily proved that the children of the sets in the *DEAD* group are always not ϵ-error tolerance and, thus, require no more testings. Hence, only the children of the *SURVIVOR* basis sets need to be tested.

Theorem 2: Assume there is a set of basis functions $S(Q) = \{B_i(\mathbf{q})\}_{i=1}^{n}$ which is not ϵ-error tolerance for $\phi(\mathbf{q})$ on a workspace Q, that is, $E_{minimax}(S(Q)) \triangleq \min_{a_i=\hat{a}_i}\max_{\mathbf{q}\in Q} ||\phi(\mathbf{q}) - \sum_{i=1}^{n} a_i B_i(\mathbf{q})|| > \epsilon$, then its children or split subsets, i.e., $SS_j(Q) \triangleq (S(Q)/\{B_j(\mathbf{q})\})$, $1 \leq j \leq n$, are also not ϵ-error tolerance, that is, $E_{minimax}(SS_j(Q)) > \epsilon$, for $1 \leq j \leq n$.

The proof of Theorem 2 can be found in [16]. Based on the above splitting the basis function set concept and Theorem 2, a procedure for finding the minimal significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on a workspace Q is summarized in the following FIND_MIN_BASIS algorithm.

Algorithm FIND_MIN_BASIS $(\epsilon, \phi(\mathbf{q}), Q, BASIS\_CAN_{\epsilon_0}(Q,\phi), BASIS_\epsilon^{MSIG}(Q,\phi))$. This algorithm finds the minimal significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on a workspace Q given the maximum error tolerance ϵ. If the algorithm performs in relative error sense, then $\phi(\mathbf{q})$ will become 1 and each basis function is divided by the original $\phi(\mathbf{q})$ in the following procedure.

Input: ϵ is the desired maximum error tolerance; $\phi(\mathbf{q})$ is the objective dynamic coefficient on a workspace Q; $BASIS\_CAN_{\epsilon_0}(Q,\phi)$ is a set of candidate ϵ-error tolerance basis functions for $\phi(\mathbf{q})$ on the workspace Q.

Output: $BASIS_\epsilon^{MSIG}(Q,\phi)$ is a minimal significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on the workspace Q.

Step 0. [*Initialization.*] Let $S_1(Q,\phi) = BASIS\_CAN_{\epsilon_0}(Q,\phi)$ and $SURVIVOR = \{S_1(Q,\phi)\}$, $DEAD = \{\}$, $m_S = |SURVIVOR| = 1$, $m_D = |DEAD| = 0$.

Step 1. [*Splitting.*]

(i) $SS \triangleq SPLIT\ (SURVIVOR) \triangleq \bigcup_{i=1}^{m_S} (SPLIT(\{S_i(Q,\phi)\}))$ and $SPLIT(\{S_i(Q,\phi)\}) \triangleq \bigcup_{j=1}^{|S_i(Q,\phi)|} (\{(S_i(Q,\phi)/\{B_i^j(\mathbf{q})\})\})$, where $B_i^j(\mathbf{q})$ indicates the jth basis term in the ϵ-error tolerance basis set $S_i(Q,\phi)$, and $SPLIT(\cdot)$ is a splitting operator defined on a set of sets.

(ii) $SD \triangleq SPLIT\ (DEAD) \triangleq \bigcup_{i=1}^{m_D} (SPLIT(\{D_i(Q,\phi)\}))$ where $D_i(Q,\phi) \in DEAD$ is not an ϵ-error tolerance basis set.

Step 2. [*Deleting the children of the DEAD group.*] Let $TEST \triangleq SS/SD \triangleq SS - (SS \cap SD)$, and $m_T = |TEST|$. If $m_T = 0$, go to *Step* 5; otherwise continue.

Step 3. [*Selecting the ϵ-error tolerance children.*]

(i) Order the elements of the set $TEST = \{T_i(Q,\phi)\}_{i=1}^{m_T}$.

(ii) Test each basis set $T_i(Q,\phi) \in TEST$ which satisfies the ϵ-error tolerance condition.

(iii) Divide the set $TEST$ into two groups, that is, $SURVIVOR1 = \{T_i(Q,\phi) | E_{minimax}(T_i(Q,\phi)) \leq \epsilon$, $T_i(Q,\phi) \in TEST\}$ and $DEAD1 = \{T_i(Q,\phi) | E_{minimax}(T_i(Q,\phi)) > \epsilon$, $T_i(Q,\phi) \in TEST\}$

(iv) If $|SURVIVOR1| = 0$, go to *Step* 5; otherwise continue.

Step 4. [*Constructing new SURVIVOR and DEAD groups.*]

(i) Let $SURVIVOR = SURVIVOR1$, and the basis sets $T_i(Q,\phi)$ in $SURVIVOR1$ will be reordered and assigned to the corresponding basis sets $S_i(Q,\phi)$ in $SURVIVOR$.

(ii) Let $DEAD = SD \cup DEAD1$, and the basis sets in $DEAD$ will be reordered with notation D_i, $1 \leq i \leq |DEAD|$.

(iii) Set $m_S = |\ SURVIVOR\ |$, $m_D = |\ DEAD\ |$, and go to *Step* 1 for next iteration.

Step 5. [*Finding a basis set in the SURVIVOR with minimum minimax approximation error.*] Select a basis set $S^*(Q,\phi)$ in the *SURVIVOR* which satisfies the following condition:

$$E_{minimax}(S^*(Q,\phi)) \le E_{minimax}(S(Q,\phi)), \text{ for any } S(Q,\phi) \in SURVIVOR\ .$$

Set $BASIS_\epsilon^{MSIG}(Q,\phi) = S^*(Q,\phi)$.

Step 6. [*Output and return.*] Output $BASIS_\epsilon^{MSIG}(Q,\phi)$ and return.

END FIND_MIN_BASIS

It should be noted that the input set of candidate ϵ_0-error tolerance basis functions, $BASIS\_CAN_{\epsilon_0}(Q,\phi)$, is an ϵ-error tolerance with $\epsilon_0 \le \epsilon$. So, one possible choice for the set of candidate ϵ_0-error tolerance basis functions for the dynamic coefficient is the basis set that contains the product terms of sinusoidal functions or polynomial functions proposed in (6) and (7). In this case, ϵ_0 is equivalent to zero. Because the basis term $B_i(\mathbf{q})$ always contains product of sinusoidal functions and polynomial functions, the maximum absolute value of basis term is usually less than one (for all rotational joints case) or product of some maximum translational joint ranges. If these translational joint variables are normalized to one using their maximum ranges, then the maximum absolute value of any basis term $B_i(\mathbf{q})$ will be less than one. This provides a heuristic criterion to drop the insignificant terms, that is, if the absolute value of the corresponding minimax fitting coefficient $|\hat{a}_i|$ is relatively insignificant and less than a small specified value, then the basis term $B_i(\mathbf{q})$ will be filtered out. This heuristic criterion can be added into the algorithm FIND_MIN_BASIS as a new step between *Step* 0 and *Step* 1. Computer simulation shows that there are so many insignificant terms with very small absolute coefficient values that the heuristic criterion will filter out many insignificant basis functions from the set of candidate basis functions.

The above algorithm tests all the children coming from the *SURVIVOR* group of the set of candidate basis functions for satisfying the ϵ-error tolerance. In the worst case, the number of sets in the *SURVIVOR* group may be very large. For example, if we assume that the number of input candidate sets is n and there are no sets in the *DEAD* group and the resulting minimal significant ϵ-error tolerance basis function set contains k basis terms, then the required number of testing, $n \gg k$, is

$$\begin{pmatrix} n \\ n-1 \end{pmatrix} + \begin{pmatrix} n \\ n-2 \end{pmatrix} + \cdots + \begin{pmatrix} n \\ k \end{pmatrix} \approx 2^n\ .$$

This indicates that the time complexity of the testing algorithm is exponential. Furthermore, it has also been shown that n is usually equal to 5^N for J_{ij}, H_{ijk} or 3^N for G_i for an N-link manipulator with rotary joints. So, the problem for the algorithm to test the dynamic coefficients for ϵ-error tolerance becomes intractable. Thus, a more efficient way of finding minimal significant ϵ-error tolerance basis function set for dynamic coefficients must be devised. One possible method is based on the decomposition of an N-link manipulator into two $(N/2)$-link pseudo-manipulators, and each pseudo-manipulator has its own dynamic coefficients. Then the total number of testing for the two $(N/2)$-link pseudo-manipulators becomes $2\cdot 2^{n_1}$, where $n_1 = 5^{N/2}$ for J_{ij}, H_{ijk} and $n_1 = 3^{N/2}$ for G_i. Generalization of the idea results in dividing an N-link manipulator into 2^k $(N/2^k)$-link pseudo-manipulators. Then the total number of testing for the 2^k $(N/2^k)$-link pseudo-manipulators becomes $2^k \cdot 2^{n_k}$, where $n_k = 5^{N/2^k}$ for J_{ij}, H_{ijk} and $n_k = 3^{N/2^k}$ for G_i and $1 \le k \le \lceil \log_2 N \rceil$. If $k = \lceil \log_2 N \rceil$, the total number of testing will be a linear order $O(N)$, that is, $2^5 \cdot N$ for J_{ij}, H_{ijk} and $2^3 \cdot N$ for G_i. Based on this concept, a multi-layered minimax simplification scheme is developed for finding the minimal significant ϵ-error tolerance basis set for the dynamic coefficients.

Consider the decomposition of the manipulator workspace, the workspace Q may be divided into two sub-workspaces, Q_1 and Q_2, where $Q_1 = \{\mathbf{q} \mid \mathbf{q} = [q_1, \cdots, q_k, q_{k+1}^c, \cdots, q_N^c]^T \in Q$, where $q_i, 1 \le i \le k$, are joint variables, and q_i^c, $k+1 \le i \le N$, are selected constants$\}$ and $Q_2 = \{\mathbf{q} \mid \mathbf{q} = [q_1^c, \cdots, q_k^c, q_{k+1}, \cdots, q_N]^T \in Q$, where $q_i^c, 1 \le i \le k$, are selected constants, and q_i, $k+1 \le i \le N$, are joint variables$\}$. One can see that $Q = Q_1 \cup Q_2$ and $Q_1 \cap Q_2 = \{\mathbf{q}^c = [q_1^c, \cdots, q_N^c]^T\}$. The dynamic coefficient $\phi(\mathbf{q})$ ($J_{ij}(\mathbf{q})$ or $H_{ijk}(\mathbf{q})$ or $G_i(\mathbf{q})$) on these two different sub-workspaces will possess different formulations and meanings. The new dynamic coefficient $\phi^1(\mathbf{q}^1) \triangleq \phi(\mathbf{q})|_{\mathbf{q}\in Q_1}$, where $\mathbf{q}^1 = [q_1, q_2, \cdots, q_k]^T$, is similar to the dynamic coefficient of a k-link manipulator in which the first k links of the N-link manipulator with the last $(N-k+1)$ links fixed or considered as an end-effector attached to the first k-link manipulator. Similarly, the dynamic coefficient, $\phi^2(\mathbf{q}^2) \triangleq \phi(\mathbf{q})|_{\mathbf{q}\in Q_2}$, where $\mathbf{q}^2 = [q_{k+1}, \cdots, q_N]^T$, means that the fixed first k links are considered as a fixed level platform attached to the base of the $(N-k+1)$-link manipulator. Thus, the sets of candidate basis functions for $\phi^1(\mathbf{q}^1)$ and $\phi^2(\mathbf{q}^2)$ can be obtained as follow:

(i) The set of candidate basis functions for $\phi^1(\mathbf{q}^1) = \{B_m \mid B_m$ is any product term of

$$\prod_{\substack{\text{all rotational joint } m_r \\ 1\le m_r\le k}} (1 + Cm_r - Sm_r - (Sm_r)^2 + Cm_r Sm_r) \times \prod_{\substack{\text{all translational joint } m_t \\ 1\le m_t\le k}} (1 + q_{m_t} + q_{m_t}^2) \quad (\text{for } \phi(\mathbf{q}^1) = J_{ij}(\mathbf{q}^1) \text{ or } H_{ijk}(\mathbf{q}^1))$$

and

$$\prod_{\substack{\text{all rotational joint } m_r \\ 1\le m_r\le k}} (1 + Cm_r + Sm_r) \times \prod_{\substack{\text{all translational joint } m_t \\ 1\le m_t\le k}} (1 + q_{m_t}) \quad (\text{for } \phi(\mathbf{q}^1) = G_i(\mathbf{q}^1))\}.$$

(ii) The set of candidate basis functions for $\phi^2(\mathbf{q}^2) = \{B_m \mid B_m$ is any product term of

$$\prod_{\substack{\text{all rotational joint } m_r \\ k< m_r\le N}} (1 + Cm_r + Sm_r + (Sm_r)^2 + Cm_r Sm_r) \times \prod_{\substack{\text{all translational joint } m_t \\ k< m_t\le N}} (1 + q_{m_t} + q_{m_t}^2) \quad (\text{for } \phi(\mathbf{q}^2) = J_{ij}(\mathbf{q}^2) \text{ or } H_{ijk}(\mathbf{q}^2))$$

and

$$\prod_{\substack{\text{all rotational joint } m_r \\ k< m_r\le N}} (1 + Cm_r + Sm_r) \times \prod_{\substack{\text{all translational joint } m_t \\ k< m_t\le N}} (1 + q_{m_t}) \quad (\text{for } \phi(\mathbf{q}^2) = G_i(\mathbf{q}^2))\}.$$

Obviously, the product terms of the basis functions for $\phi^1(\mathbf{q}^1)$ and $\phi^2(\mathbf{q}^2)$ form a set of candidate basis functions for the dynamic coefficient $\phi(\mathbf{q})$. Based on this splitting of workspace Q, we would like to introduce some new notations and operators before we discuss the simplification procedure.

Definition 5: Let $d^{(i)}$ denote the ith element in a set S. Then the set product of $S_a = \{d_a^{(1)}, d_a^{(2)}, \cdots, d_a^{(i)}\}$ and $S_b = \{d_b^{(1)}, d_b^{(2)}, \cdots, d_b^{(j)}\}$ defined as $S_a * S_b$ is the set that includes possible product terms of $(d_a^{(1)} + d_a^{(2)} + \cdots + d_a^{(i)})(d_b^{(1)} + d_b^{(2)} + \cdots + d_b^{(j)})$. Note that the set product is an associative operator, i.e., grouping of the elements does not matter, $(S_a * S_b) * S_c = S_a * (S_b * S_c) = S_a * S_b * S_c$.

Let $S_{0,i}$ be the set defined for $J_{ij}(\mathbf{q})$ and $H_{ijk}(\mathbf{q})$

$$S_{0,i} = \begin{cases} \{1, C_i, S_i, S_i^2, C_i S_i\} & ; \text{ if joint } i \text{ is rotational} \\ \{1, q_i, q_i^2\} & ; \text{ if joint } i \text{ is translational} \end{cases} \tag{24a}$$

and for $G_i(\mathbf{q})$, we have

$$S_{0,i} = \begin{cases} \{1, C_i, S_i\} & ; \text{ if joint } i \text{ is rotational} \\ \{1, q_i\} & ; \text{ if joint } i \text{ is translational} \end{cases} \tag{24b}$$

From (6) and (7), it can be shown that the set of candidate basis functions for the dynamic coefficient $\phi(\mathbf{q})$ ($J_{ij}(\mathbf{q})$ or $H_{ijk}(\mathbf{q})$ or $G_i(\mathbf{q})$), $BASIS\_CAN(Q,\phi)$, is equivalent to the set products $S_{0,1} * S_{0,2} * \cdots * S_{0,N}$. Since the set product is associative, $S_{0,1} * S_{0,2} * \cdots * S_{0,N}$ can be expressed as $(S_{0,1} * \cdots * S_{0,k}) * (S_{0,(k+1)} * \cdots * S_{0,N})$, where $(S_{0,1} * \cdots * S_{0,k})$ and $(S_{0,(k+1)} * \cdots * S_{0,N})$ are respectively the sets of candidate basis functions for $\phi^1(\mathbf{q}^1)$ and $\phi^2(\mathbf{q}^2)$. By choosing suitable ϵ_0 and $\mathbf{q}^c$ and applying the FIND_MIN_BASIS algorithm to $(S_{0,1} * \cdots * S_{0,k})$ and $(S_{0,(k+1)} * \cdots * S_{0,N})$ separately, one may obtain the minimal significant ϵ_0-error tolerance basis sets $(S_{0,1} * \cdots * S_{0,k})'$ and $(S_{0,(k+1)} * \cdots * S_{0,N})'$. Based on the above concepts, the set product of $(S_{0,1} * \cdots * S_{0,k})'$ and $(S_{0,(k+1)} * \cdots * S_{0,N})'$ forms a set of candidate ϵ-error tolerance basis for $\phi(\mathbf{q})$. This decomposition idea can be extended to a more general case and to any manipulator with prismatic and rotary joints. There are two sets of candidate basis functions for the dynamic coefficients of two $N/2$-link manipulators at the first splitting by setting $k = N/2$, four sets of candidate basis functions for the dynamic coefficients of four $N/4$-link manipulators at the second splitting, and 2^k sets of candidate basis functions for the dynamic coefficients of 2^k $(N/2^k)$-link manipulators at the kth until $\lceil \log_2 N \rceil$ splittings. For example, $S_{0,1} * S_{0,2} \cdots * S_{0,8} = (((S_{0,1}) * (S_{0,2})) * ((S_{0,3}) * (S_{0,4}))) * (((S_{0,5}) * (S_{0,6})) * ((S_{0,7}) * (S_{0,8})))$, it can be seen that there are $\lceil \log_2 8 \rceil = 3$ splittings involved in the above expression. Applying the FIND_MIN_BASIS algorithm to each set of candidate basis functions in the above expression from inside to outside, step by step, and tuning the parameters ϵ_0 and $\mathbf{q}^c$ subject to the feedback information for each step, a minimal significant ϵ-error tolerance basis function set can be obtained. The generation of the minimal significant ϵ-error tolerance basis function set is shown in Fig. 2 and the procedure is described as follows:

(i) Apply the FIND_MIN_BASIS algorithm to $S_{0,1}, S_{0,2}, \cdots, S_{0,N}$ and obtain the reduced ϵ_0-error tolerance basis sets, $S'_{0,1}, S'_{0,2}, \cdots, S'_{0,N}$, where usually $\epsilon_0 = \epsilon$.

(ii) Form a new set of candidate ϵ-error tolerance basis functions for the next step, that is, $S_{1,1} = S'_{0,1} * S'_{0,2}, S_{1,2} = S'_{0,3} * S'_{0,4}, \cdots, S_{1,(N/2)} = S'_{0,N-1} * S'_{0,N}$

(iii) Check and test $S_{1,1}, S_{1,2}, \cdots, S_{1,N/2}$ for satisfying the ϵ-error tolerance criterion. If one of them is not an ϵ-error tolerance, then a smaller ϵ_0 or another constant vector $\mathbf{q}^c$ should be selected and the procedure is repeated again. Otherwise, continue next iteration until the $\lceil \log_2 N \rceil$th step.

It is suggested that ϵ_0 is tuned for each step and the constant vector $\mathbf{q}^c$ is tuned if the final ϵ-error tolerance basis set for $\phi(\mathbf{q})$ cannot be obtained. Generally, $S_{r,i}$ represents the ith ϵ-error tolerance basis set for the dynamic coefficient of the ith pseudo 2^r-link manipulator (i.e., the N-link manipulator with $(N-2^r)$ links fixed) in the rth iteration. There are $(N/2^r)$ ϵ-error tolerance basis sets $S_{r,i}$ which will be considered and reduced to the minimal significant ϵ-error tolerance basis sets $S'_{r,i}$. After $\lceil \log_2 N \rceil$ iterations, the final minimal significant ϵ-error tolerance basis set, $S'_{\lceil \log_2 N \rceil, 1}$, will be obtained. Basically, this decomposition approach is a multi-layered decision scheme for fitting a dynamic coefficient with a suggested set of candidate basis functions. Each layer consists of a bank of ϵ-error tolerance basis functions with inputs from the previous layer having been passed through a selection layer performing the FIND_MIN_BASIS algorithm. This approach is also a feedback scheme in a sense that if any basis set in any layer is not ϵ-error tolerance, then feedback signal will require previous layers to adjust its parameters, either ϵ_0 or $\mathbf{q}^c$. This multi-layered decision scheme is shown in Fig. 2 which illustrates the rth layer and the selection layer deals with the problem of $(N/2^r)$ 2^r-link manipulators. This multi-layered decision scheme overcomes the difficulty of obtaining a minimal significant ϵ-error tolerance basis set from an N-link manipulator which has a large number of basis functions in its candidate set.

Notations relating to the multi-layered simplification procedure are defined as:

(i) $\mathbf{q}_{r,i}$ = A $(2^r \times 1)$ joint variable vector which contains the joint variables of two $(2^{r-1} \times 1)$ joint variable vectors $\mathbf{q}_{r,(2i-1)}$ and $\mathbf{q}_{r,2i}$ in the previous layer.

(ii) $Q_{r,i}$ = The projection subspace which contains the $(2^r \times 1)$ joint variable vector $\mathbf{q}_{r,i}$ within the range limitations of the workspace Q.

(iii) $\overline{Q}_{r,i}$ = A subspace which contains any $(N \times 1)$ joint variable vector belonging to Q which has 2^r joint variable components and $(N-2^r)$ constant components corresponding to the $(2^r \times 1)$ joint variable vector $\mathbf{q}_{r,i}$ and the remaining $(N-2^r)$ components of the constant vector $\mathbf{q}^c$, respectively.

With these new notations, the details of the multi-layered decision scheme for simplifying the dynamic coefficients of a manipulator with prismatic and/or rotary joints is summarized in the following Algorithm MIN_BASIS_GEN.

Algorithm MIN_BASIS_GEN. This algorithm generates minimal significant ϵ-error tolerance basis functions using the multi-layered decision scheme.

Input: N = number of joints, ϵ = maximum tolerance error, Q = manipulator workspace, and $\phi(\mathbf{q})$ = the objective dynamic coefficient.

Output: minimal significant ϵ-error tolerance basis set for $\phi(\mathbf{q})$ on the workspace Q, $BASIS_\epsilon^{MSIG}(Q,\phi)$.

Step 0. [*Initialization.*]

(0.a) Select an $(N \times 1)$ constant vector $\mathbf{q}^c$ within the working ranges of the manipulator, $\mathbf{q}^c$ will change if *Step* (3.b) indicates that the final ϵ-error tolerance basis set cannot be obtained.

(0.b) The set of candidate basis functions for 0th iteration $S_{0,i}$, $1 \le i \le N$, is defined in (24a) and (24b).

(0.c) $\mathbf{q}_{0,i}, Q_{0,i}, \overline{Q}_{0,i}$ correspond to joint i variable q_i, $1 \le i \le N$.

Step 1. [*Start the* $\lceil \log_2 N \rceil$*-layered decision procedure.*]
For $r = 0$ step 1 until $\lceil \log_2 N \rceil$, Do

(1.a) Construct $\mathbf{q}_{r,i}, Q_{r,i}, \overline{Q}_{r,i}$ which correspond to the set $S_{r,i}$, $1 \le i \le \lceil N/2^r \rceil$.

(1.b) Start with $\epsilon_0 = \epsilon$, and ϵ_0 will change to a smaller value if *Step* (3.b) responds with a change command (ϵ_0 is selected to control the number of passing basis functions).

Step 2. [*Find minimal significant* ϵ_0*-error tolerance basis functions.*]
For $i = 1$ step 1 until $\lceil N/2^r \rceil$, Do

(2.a) Set $\phi^{(r,i)}(\mathbf{q}_{r,i}) \triangleq \phi(\mathbf{q})\,|_{\mathbf{q} \in \overline{Q}_{r,i}}$.

(2.b) Perform FIND_MIN_BASIS $(\epsilon_0, \phi^{(r,i)}(\mathbf{q}_{r,i}), Q_{r,i}, S_{r,i}, BASIS_\epsilon^{MSIG}(Q_{r,i}, \phi^{(r,i)}))$.

(2.c) Set $S'_{r,i} = BASIS_\epsilon^{MSIG}(Q_{r,i}, \phi^{(r,i)})$.
End Do_Block of *Step* 2.

Step 3. [*Form* ϵ*-error tolerance basis candidates for the next iteration.*]
For $i = 1$ step 1 until $\lceil N/2^{r+1} \rceil$, Do

(3.a) Perform the set product $S_{r+1,i} = S'_{r,(2i-1)} * S'_{r,(2i)}$

(3.b) Check $S_{r+1,i}$ for satisfying ϵ-error tolerance. If $S_{r,i}$ is not ϵ-error tolerance, Do

(i) If $r = (\lceil \log_2 N \rceil - 1)$, then go to *Step* (0.a); otherwise continue.

(ii) Go to *Step* (1.b); otherwise continue
End Do_Block.

End Do_Block of *Step* 3.
End Do_Block of *Step* 1.

Step 4. [*Output and return.*]
Set $BASIS_\epsilon^{MSIG}(Q,\phi) = S'_{\lceil \log_2 N \rceil, 1}$ and output the result.

END MIN_BASIS_GEN

In order to reduce the complexity and avoid the redundant operations of the above algorithm, it is better to use the compact set of candidate basis functions based on the functional dependency of the dynamic coefficient instead of the general formulation in (6) and (7). Thus, the dimension of the problem will become $\bar{N}$ which is the number of effective joint variables involved in the dynamic coefficient. Similarly, these algorithm-related notations and parameters $\mathbf{q}_{r,i}, Q_{r,i}, \overline{Q}_{r,i}$, and the basis candidates for the ith iteration, $S_{0,i}$, also correspond to the effective joint variables involved in the dynamic coefficient. Thus, the number of layers in the algorithm would be reduced to $\lceil \log_2 \bar{N} \rceil$. For example, the dynamic coefficient J_{22} of an N-link manipulator only needs $\lceil \log_2 \bar{N} \rceil$ (i.e. $\bar{N} = N-2$) iterations in the algorithm. It must also be mentioned that, if the total number of the reduced basis sets generated in the previous layer, $\lceil N/2^r \rceil$, is not even, then a pseudo-basis set $S'_{r,\lceil N/2^r \rceil + 1} = \{1\}$ must be added in the set product operation in *Step* (3.a).

The proposed multi-layered minimax simplification procedure is implemented in a "C" program and can be used to generate the simplified dynamics coefficients for any manipulator with prismatic and/or rotary joints. The Stanford arm which consists of rotational and translational joints is used as an example to verify the simplification algorithm. The computer results are tabulated in Table 1. Computer simulation shows that our simplified expressions are better than Bejczy's simplified expressions [2]. This is due to the fact that Bejczy deleted the insignificant terms which have some correlations with other basis terms. Our results obtained from the minimax approximation will always provide a better fitting. Another important characteristic of our technique is that our simplified expressions are generated to achieve the specified maximum error tolerances. Details about the computer simulation results can be found in [16].

V. CONCLUSION

This paper presents the development of an efficient minimax simplification scheme for the automatic generation of closed-form manipulator motion equations in symbolic form while maintaining the desired manipulator system performance under a PD controller. Each joint-position-dependent dynamic coefficient which involves in the motion equations has been shown to be a linear combination of basis functions consisting of the product terms of sinusoidal and polynominal functions of the generalized coordinates. The significant basis functions of the dynamic coefficient can be selected, after performing the multi-layered decision procedure. The linear combination of these significant basis functions are then utilized to construct the simplified expression for the dynamic coefficient based on the minimax fitting technique. Computer simulation shows that our simplified expressions are better than Bejczy's simplified expressions. Another important characteristic of our technique is that our simplified expressions are generated to achieve the desired control performance specifications.

REFERENCES

[1] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line Computational Scheme for Mechanical Manipulator," *Trans. of ASME J. of Dynam. Syst., Meas. and Contrl.*, Vol. 102, pp. 69-76, June 1980.

[2] A. K. Bejczy, "Robot Arm Dynamics and Control," JPL Technical Memorandum 33-609, Pasadena, CA, February 1974.

[3] R. P. Paul, *Robot Manipulator: Mathematics, Programming and Control*. MIT Press, Cambridge, MA (1981).

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

[5] C. S. G. Lee and M. J. Chung, "A Adaptive Control Strategy for Mechanical Manipulator," *IEEE Trans. on Autom. Contrl.*, Vol. AC-29, No. 9, pp. 837-840, Sept. 1984.

[6] J.-J. E. Slotine, "The Robust Control of Robot Manipulators," *Int'l J. Robotics Research*, Vol. 4, No. 2, 1985.

[7] T. J. Tarn et al., "Nonlinear Feedback in Robot Arm Control," *Proc. 1984 Conf. on Decision and Control*, Las Vegas, Nevada, pp. 736-751, December 1984.

[8] A. K. Bejczy and R. P. Paul, "Simplified Robot Arm Dynamics for Control," *Proc. of the 20th IEEE Conf. on Decision and Control*, pp. 261-262, San Diego, CA, Dec. 1981.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] V. D. Tourassis and C. P. Neuman, "Properties and Structure of Dynamic Robot Models for Control Engineering Applications," *Mechanism and Machine Theory*, Vol. 20, No. 1, pp. 27-40, Jan. 1985.

[14] G. A. Watson, *Approximation Theory and Numerical Methods*, John Wiley and Sons, New York, 1980.

[15] K. Ogata, *Modern Control Engineering*, Prentice-Hall, N.J., 1970.

[16] P. R. Chang and C. S. G. Lee, "A Multi-Layered Minimax Decision Scheme for Simplification of Robot Dynamic Model Based on System Performance," Technical Report TR-EE 86-28, School of Electrical Engineering, Purdue University, October 1986.

Table 1. The Significant Dynamic Coefficients for Simplified Dynamic Model of the Stanford Arm

Simplified expression	Minmax error	Maximum error tolerance
$J_{11} = 1.455 + 2.594S_2^2 - 5.754S_2^2 q_3 + 6.627S_2^2 q_3^2$	0.0545†	0.1
$J_{22} = 4.667 - 5.332q_3 + 6.361q_3^2$	0.0412†	0.1
$J_{33} = 6.48$	0.00†	0.1
$J_{44} = 0.1148$	0.0663†	0.1
$J_{55} = 0.114$	0.00†	0.1
$J_{66} = 0.0203$	0.00†	0.1
$J_{12} = 0.543\,C_2 - 1.144\,q_3 C_2$	0.1071	0.128
$J_{13} = -1.0453\,S_2$	0.1126	0.151
$J_{14} = 0.1316S_2 q_3 S_4 S_5$	0.0328	0.041
$J_{15} = -0.1138S_2 q_3 C_4 C_5$	0.0359	0.046
$J_{23} = -0.1150S_4 S_5$	0.0063	0.015
$J_{24} = 0.1164\,q_3 C_4 C_5$	0.0091	0.021
$J_{25} = 0.1162\,q_3 S_4 C_5$	0.0165	0.022
$J_{35} = -0.115\,S_5$	0.0007	0.015
$G_2 = -27.346S_2 + 63.381S_2 q_3 + 1.093S_2 C_5 + 1.118C_2 S_4 S_5$	0.06	0.31
$G_3 = -63.406C_2$	4.1×10^{-5}	0.276
$G_4 = 1.127S_2 C_4 S_5 + 0.0554S_2 S_4$	6.1×10^{-5}	0.038
$G_5 = 1.126S_2 S_4 C_5 + 1.1266C_2 S_4$	0.007	0.038
$H_{112} = 2.289C_2 S_2 - 4.267C_2 S_2 q_3 + 5.46C_2 S_2 q_3^2 + 0.299C_2 S_2 q_3^2 C_5$	0.108	0.314
$H_{113} = -2.723S_2^2 + 6.467S_2^2\,q_3$	0.090	0.61
$H_{115} = -0.0733S_2^2 S_5 + 0.0128C_2 S_2 S_5 + 0.02C_2 S_2 S_4 S_5$	0.065	0.076
$H_{122} = -0.462S_2 + 1.069S_2 q_3$	0.099	0.106
$H_{123} = -1.047C_2 + 0.114C_2 C_4 S_5$	0.006	0.068
$H_{144} = -0.0273S_2 C_4 C_5 + 0.0252S_2 q_3 C_4 C_5$	0.0131	0.0168
$H_{145} = 0.1132S_2 q_3 S_4 C_5$	0.0128	0.0168
$H_{155} = 0.1132S_2 q_3 C_4 S_5$	0.0126	0.0154
$H_{211} = -2.791C_2 S_2 + 6.468C_2 S_2 q_3 - 7.167C_2 S_2 q_3^2$	0.1255	0.572
$H_{214} = -0.115C_2 q_3 S_4 S_5$	0.0153	0.028
$H_{215} = 0.101C_2 q_3 C_4 C_5$	0.0293	0.03
$H_{223} = -2.712 + 6.47q_3$	0.0769	0.1124
$H_{225} = -0.1186\,q_3 C_5$	0.0071	0.032
$H_{244} = -0.1166\,q_3 S_4 S_5$	0.0088	0.0318
$H_{245} = 0.07798\,q_3 C_4 C_5$	0.0252	0.0318
$H_{255} = -0.1148\,q_3 S_4 S_5$	0.0006	0.0318
$H_{311} = 2.726S_2^2 - 6.467S_2^2\,q_3$	0.0923	0.725
$H_{314} = -0.1161S_2 S_4 S_5$	0.006	0.018
$H_{315} = 0.0555S_2 C_4 C_5$	0.006	0.0187
$H_{322} = 2.712 - 6.47q_3$	0.0769	0.748
$H_{324} = -0.115C_4 S_5$	0.0063	0.0186
$H_{325} = -0.115S_4 C_5$	7×10^{-4}	0.032
$H_{355} = -0.1075C_5$	0.0075	0.0186
$H_{411} = -0.1125C_2 S_2 q_3 C_4 S_5$	0.0233	0.0359
$H_{412} = 0.1168C_2 q_3 S_4 S_5$	0.0154	0.07
$H_{413} = 0.05S_2 S_4 S_5$	0.0637	0.07
$H_{511} = 0.1365S_2^2\,q_3 S_5 - 0.0184S_2^2\,q_3 S_4 S_5 + 0.0142C_2 S_2 q_3 S_5$	0.06	0.08
$H_{512} = -0.1135C_2 q_3 C_4 C_5$	0.016	0.062
$H_{513} = -0.1149S_2 C_4 C_5$	6.8×10^{-4}	0.059
$H_{522} = 0.11865\,q_3 S_5$	0.007	0.059
$H_{523} = 0.115S_4 C_5$	7×10^{-4}	0.058

† These minmax errors for J_{ii} are in the sense of relative error.

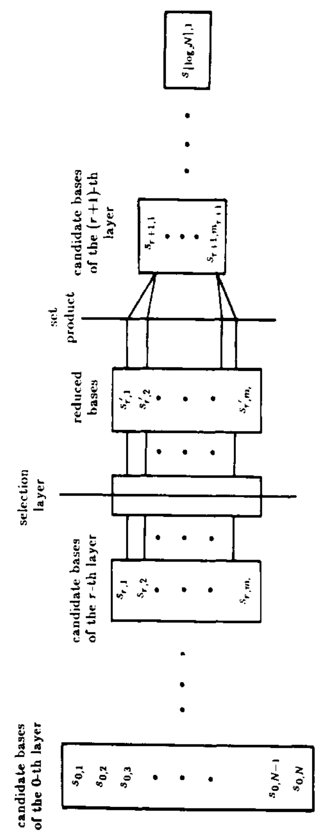

Figure 2. Multi-Layered Minmax Simplification Scheme

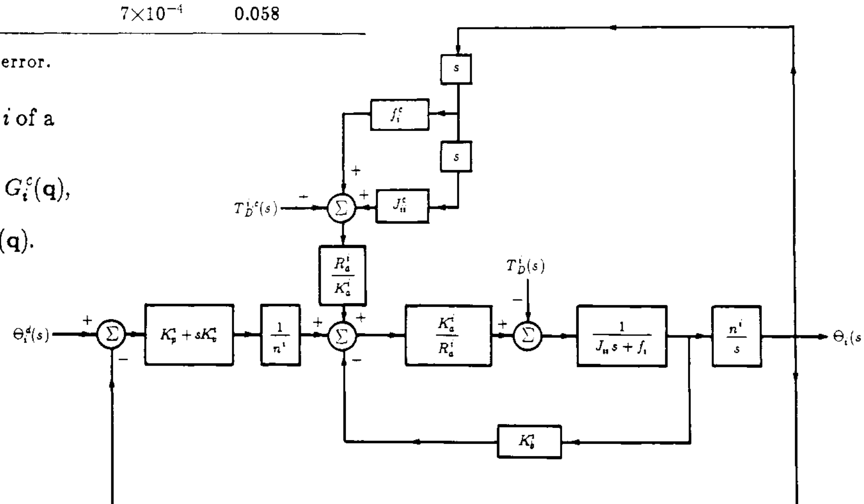

Figure 1. Complete PD Controller for Joint i of a Manipulator having N Joints,

where $T_D^{i,c}(s) = \sum_{j\neq i} J_{ij}^c(\mathbf{q})\,\ddot{q}_j + \sum_j \sum_k H_{ijk}^c\,\dot{q}_j\dot{q}_k + G_i^c(\mathbf{q})$,

and $T_D^i(s) = \sum_{j\neq i} J_{ij}(\mathbf{q})\,\ddot{q}_j + \sum_j \sum_k H_{ijk}\,\dot{q}_j\dot{q}_k + G_i(\mathbf{q})$.