Computers Math. Applic. Vol. 17, No. 12, pp. 1545–1561, 1989
Printed in Great Britain. All rights reserved

0097-4943/89 $3.00 + 0.00

AN ALGORITHM FOR THE INVERSE DYNAMICS OF n-AXIS GENERAL MANIPULATORS USING KANE'S EQUATIONS

J. ANGELES and O. MA
Department of Mechanical Engineering, Robotic Mechanical Systems Laboratory,
McGill Research Centre for Intelligent Machines, McGill University, Montréal, Québec, Canada

A. ROJAS
National Autonomous University of Mexico, Villa Obregón, Mexico

(*Received* 4 *April* 1988)

Abstract—Presented in this paper is an algorithm for the numerical solution of the inverse dynamics of robotic manipulators of the serial type, but otherwise arbitrary. The algorithm is applicable to manipulators containing n joints of the rotational or the prismatic type. For a given set of Hartenberg–Denavit and inertial parameters, as well as for a given trajectory in the space of joint coordinates, the algorithm produces the time histories of the n torques or forces required to drive the manipulator through the prescribed trajectory. The algorithm is based on Kane's dynamical equations of mechanical multibody systems. Moreover, the complexity of the algorithm presented here is lower than that of the most efficient inverse-dynamics algorithm reported in the literature. Finally, the applicability of the algorithm is illustrated with two fully solved examples.

1. INTRODUCTION

The inverse dynamics of robotic manipulators has been approached with a variety of methods, the most widely applied being those based on the Newton–Euler (NE) and the Euler–Lagrange (EL) formulations [1–11]. Although essentially equivalent [12], the two said formulations regard the dynamics problems associated with rigid-link manipulators from very different viewpoints. In fact, whereas the NE formulation establishes equations of balance of external and inertia forces and moments *acting on each link of the manipulator*, the EL formulation establishes equations of balance on the *projections* of the aforementioned forces and moments onto the axes of the manipulator. Moreover, the NE formulation requires the computation of *nonworking constraint forces and moments*, whereas the EL formulation eliminates those from the outset. For manipulators of the serial type, computing constraint forces and moments poses no problem, because these can be computed recursively. However, for manipulators containing kinematic loops within their architectures, computing constraint forces and moments adds a substantial computational load to the problem, because the said computations require the solution of linear systems of equations. Hence, while the complexity of the inverse dynamics algorithms associated with manipulators of the serial type is linear, i.e. of $O(n)$. where n is the number of joints, that of the algorithms associated with manipulators containing one kinematic loop and m joints within the loop is expected to be cubic, i.e. of $O(p^3)$, where $p \equiv 5m$, since each single-degree-of-freedom revolute or prismatic joint produces five unknown nonworking constraint forces. On the other hand, the derivation of EL equations for serial-type manipulators is too complex if done directly, and too cumbersome if done recursively [3]. Moreover, while many industrial manipulators are constituted essentially by serial-type architectures, a closer look at their transmission mechanisms reveals one or various kinematic loops. Hence, alternate formulations should be investigated, which would lead to simple algorithms of inverse and direct dynamics of manipulators containing such loops. This is the motivation behind the study reported here. With this regard, Kane's equations were investigated first in the realm of serial-type manipulators, the extension to manipulators with kinematic loops being feasible, as shown in a forthcoming paper.

Kane and Levinson [13] proposed a method for the forward and inverse dynamics of robotic manipulators, which was first introduced by Kane [14] for general nonholonomic mechanical systems. Basically, this method consists of writing the dynamical equations of the system at hand in terms of *generalized coordinates*, *generalized speeds*, *partial velocities* and *partial angular velocities*, as the balance of *n generalized active* and *inertia forces*. Moreover, the active and inertia forces are derived using the NE rather than the EL equations, while eliminating forces of constraint. These equations are, thus, the D'Alembert equations of the system written in a Lagrangian form, i.e. in terms of generalized quantities. Kane's dynamical equations are, in fact, equivalent to a set of n independent EL equations, where n is the degree of freedom of the system. What Kane and Levinson [13] proposed was to derive Kane's equations for individual manipulators. Although their procedure is conceptually simple, it requires a considerable experience with the handling of complex mechanical systems in order to set up the suibable transformations of the arising time derivatives of the generalized coordinates. Moreover, for the Stanford arm, which they chose as the example to illustrate the procedure, it is required to define up to 263 auxiliary variables, which change from manipulator to manipulator, and are not computable from simple recursive formulae. Presented in this paper is an algorithm that only requires for the user to supply the following information: (i) the Hartenberg–Denavit parameters of the manipulator; (ii) the inertial parameters of each individual link, namely, its mass, its mass-center location, and its centroidal inertia tensor; and (iii) the time histories of the joint coordinates describing a trajectory in the joint-coordinate space. The algorithm produces the torques or forces that the actuators driving each joint must supply in order to produce the desired motion, an analysis that is known as *inverse dynamics*. This algorithm, based on Kane's dynamical equations, is proven to produce the aforementioned force and torque requirements as systematically as the EL formulation and as efficiently as the NE formulation. As a consequence, the complexity of the algorithm is roughly 5% lower than that of the most efficient algorithm reported in the literature [10]. In addition to this, the algorithm proposed can be extended to mechanical systems with any type of holonomic constraints, e.g. gears, pulleys, sprockets and cam mechanisms, as well as to systems with multiple kinematic loops and nonholonomic constraints.

This paper is organized as follows: in Section 2, Kane's equations are briefly recalled for quick reference, the kinematic analysis being described in Section 3, whereas Section 4 presents the basic relations needed to derive the recursive formulae that are presented in Section 5. Furthermore, the algorithm and its complexity are analyzed in Section 6, while Section 7 includes two fully-solved examples, and the paper ends with conclusions presented in Section 8.

2. KANE'S DYNAMICAL EQUATIONS

A serial-type robot manipulator with either rotational or prismatic pairs is a holonomic mechanical system [16]. Hence, a set of independent generalized coordinates can be defined as $\{q_i\}_1^n$ which is then grouped in the n-dimensional vector of *generalized coordinates* $\mathbf{q}$. The *generalized speeds* are defined simply as the time derivatives of $\mathbf{q}$. Let $\dot{\mathbf{c}}_k$, $\ddot{\mathbf{c}}_k$, ω_k, and $\dot{\omega}_k$ be the velocity and acceleration of the mass center and the angular velocity and acceleration of the kth rigid-body of the system, respectively. Then the contribution of this body to the *generalized inertia force* of the system is defined as

$$\phi_k^* = -\left(\frac{\partial \dot{\mathbf{c}}_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} m_k \ddot{\mathbf{c}}_k - \left(\frac{\partial \omega_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \dot{\mathbf{h}}_k, \tag{1}$$

where ϕ_k^* is an n-dimensional vector and $\dot{\mathbf{h}}_k$ is the time derivative of the angular momentum of the kth body about its mass center, i.e.

$$\dot{\mathbf{h}}_k = \mathbf{I}_k \dot{\omega}_k + \omega_k \times \mathbf{I}_k \omega_k, \tag{2}$$

where $\mathbf{I}_k$ is the centroidal inertia tensor of the kth body. Moreover, $\partial \dot{\mathbf{c}}_k / \partial \dot{\mathbf{q}}$ and $\partial \omega_k / \partial \dot{\mathbf{q}}$, referred to as the *partial velocities* and *partial angular velocities* in the context of Kane's equations, are $3 \times n$ matrices.

The n-dimensional *generalized inertia force* of the system is defined as

$$\phi^* = \sum_{k=1}^{n} \phi_k^* = -\sum_{k=1}^{n} \left[\left(\frac{\partial \dot{\mathbf{c}}_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} m_k \ddot{\mathbf{c}}_k + \left(\frac{\partial \omega_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \dot{\mathbf{h}}_k\right]. \tag{3}$$

On the other hand, if the kth body is acted upon by a system of forces and moments, henceforth referred to as *actions* generically, which produce a resultant force $\mathbf{f}_k$ acting at the mass center of the body, and a moment $\mathbf{n}_k$, then the n-dimensional vector of *generalized force*, ϕ_k, associated with the kth body, is defined as

$$\phi_k = \left(\frac{\partial \dot{\mathbf{c}}_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \mathbf{f}_k + \left(\frac{\partial \omega_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \mathbf{n}_k. \tag{4}$$

The n-dimensional vector of *generalized active force* of the overall system is defined correspondingly as

$$\phi = \sum_{k=1}^{n} \left[\left(\frac{\partial \dot{\mathbf{c}}_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \mathbf{f}_k + \left(\frac{\partial \omega_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \mathbf{n}_k\right], \tag{5}$$

and hence, Kane's dynamical equations take on the simple form [17]

$$\phi + \phi^* = \mathbf{0}. \tag{6}$$

3. KINEMATIC ANALYSIS

3.1. *Hartenberg–Denavit notation*

In robotics research, the Hartenberg–Denavit (HD) notation [18] is widely used. In this algorithm, a modified HD notation [15] is introduced, as shown in Figs 1(a) and 1(b). It is assumed that the manipulator under study is of the serial-type with $n + 1$ links including the base link and n joints of either the rotational or the prismatic type. The links are numbered consecutively from 0 to n, starting from the base link to the end one, and the joints are numbered from 1 to n so that the ith joint couples links $i - 1$ and i. As shown in Fig. 1(a), the coordinate frame X_i, Y_i, Z_i $(i = 1, 2, \ldots, n)$ is defined to be fixed in the ith link with its origin O_i at the ith joint. Moreover, the coordinate frame X_0, Y_0, Z_0 is the base reference frame whose origin can be anywhere in the base link. Additionally, the Z_i axis is defined along the axis of the ith joint, whereas X_i is defined as the common perpendicular to Z_i and Z_{i+1}, directed from the former to the latter. Based on this notation, the relative position and orientation of the $(i - 1)$st and the ith links are completely described by four HD parameters, as shown in Fig. 1(b) and explained as follows:

α_i—angle from the Z_{i-1} to Z_i axes in the direction of the X_{i-1} axis;
a_i—distance between the Z_{i-1} and Z_i axes, hence it is nonnegative;
b_i—Z_i coordinate of the intersection of the X_{i-1} axis and Z_i axis;
θ_i—angle from the X_{i-1} to X_i axes in the direction of the Z_i axis.

Among these parameters, θ_i is the *joint variable* if the ith joint is rotational and b_i is the joint variable if the ith joint is prismatic. The remaining parameters are constant.

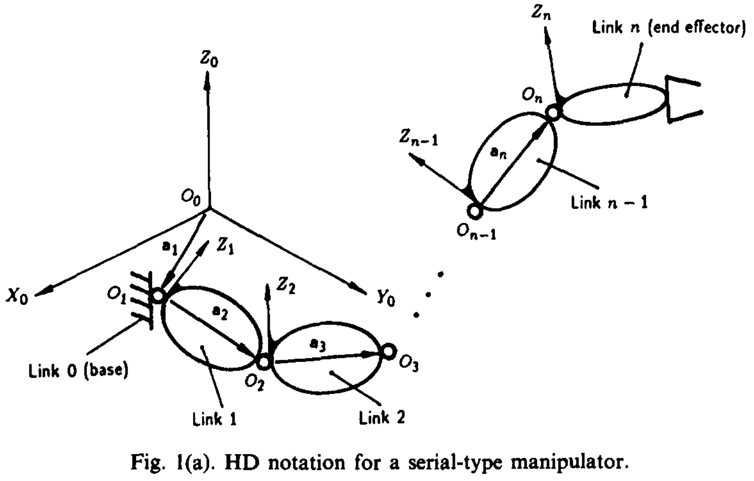

Fig. 1(a). HD notation for a serial-type manipulator.

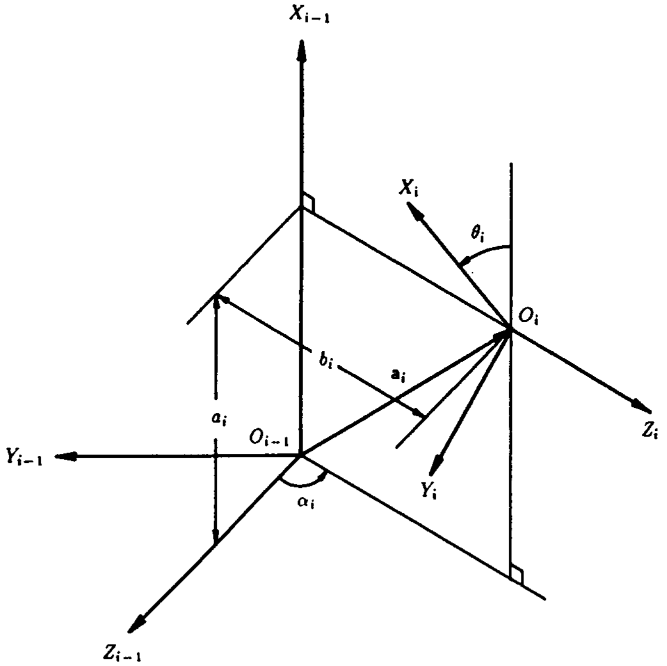

Fig. 1(b). HD parameters.

In this paper, unless otherwise indicated, any vector or matrix represented in the ith coordinate frame is enclosed in brackets and subscripted accordingly, i.e. $[\cdot]_i$.

Based upon the above notation, the rotation matrix of the ith coordinate frame with respect to the $(i-1)$st one, in $(i-1)$st coordinates, is found to be the following:

$$\mathbf{Q}_i \equiv [\mathbf{Q}_i]_{i-1} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_i \end{pmatrix} \tag{7}$$

and the translation vector from origin O_{i-1} to origin O_i. in $(i-1)$st coordinates, is

$$[\mathbf{a}_i]_{i-1} = \begin{pmatrix} a_i \\ b_i \sin\alpha_i \\ -b_i \cos\alpha_i \end{pmatrix}. \tag{8}$$

Moreover, the unit vector in the direction of the axis of the ith joint has the following components:

$$[\mathbf{e}_i]_i = [0, 0, 1]^{\mathrm{T}}. \tag{9}$$

3.2. *Angular velocities and accelerations*

The angular velocity and acceleration of the ith link can be computed recursively as follows:

$$\omega_i = \begin{cases} \omega_{i-1} + \dot\theta_i \mathbf{e}_i, & \text{if the } i\text{th joint is rotational,} \\ \omega_{i-1}, & \text{if the } i\text{th joint is prismatic,} \end{cases} \tag{10}$$

$$\dot\omega_i = \begin{cases} \dot\omega_{i-1} + \omega_{i-1} \times \dot\theta_i \mathbf{e}_i + \ddot\theta_i \mathbf{e}_i, & \text{if the } i\text{th joint is rotational,} \\ \dot\omega_{i-1}, & \text{if the } i\text{th joint is prismatic,} \end{cases} \tag{11}$$

for $i = 1, 2, \ldots, n$, whereas ω_0 and $\dot\omega_0$ are angular velocity and angular acceleration of the base link. In order to reduce the complexity of the algorithm, all vector and tensor quantities of the ith link will be expressed in the ith coordinate frame. Hence, the angular velocities and accelerations are computed recursively as follows:

$$[\omega_i]_i = \begin{cases} \mathbf{Q}_i^{\mathrm{T}}[\omega_{i-1}]_{i-1} + \dot{\theta}_i[\mathbf{e}_i]_i, & \text{if the } i\text{th joint is rotational,} \\ \mathbf{Q}_i^{\mathrm{T}}[\omega_{i-1}]_{i-1}, & \text{if the } i\text{th joint is prismatic,} \end{cases} \tag{12}$$

$$[\dot{\omega}_i]_i = \begin{cases} \mathbf{Q}_i^{\mathrm{T}}[\dot{\omega}_{i-1}]_{i-1} + [\omega_i \times \dot{\theta}_i\mathbf{e}_i + \ddot{\theta}_i\mathbf{e}_i]_i, & \text{if the } i\text{th joint is rotational,} \\ \mathbf{Q}_i^{\mathrm{T}}[\dot{\omega}_{i-1}]_{i-1}, & \text{if the } i\text{th joint is prismatic.} \end{cases} \tag{13}$$

If the base link is chosen as an inertial reference frame, then,

$$[\omega_0]_0 = \mathbf{0}, \quad [\dot{\omega}_0]_0 = \mathbf{0}. \tag{14}$$

3.3. *Velocities and accelerations of the mass centers*

Let $\mathbf{c}_i$ be the position vector of the mass center, C_i, of the ith link, ρ_i being the vector directed from O_i to C_i, as shown in Figs 2(a) and 2(b). The position vectors of two successive mass centers observe the following relationship:

$$\mathbf{c}_i = \mathbf{c}_{i-1} - \rho_{i-1} + \mathbf{a}_i + \rho_i, \tag{15}$$

or, in the ith coordinate frame,

$$[\mathbf{c}_i]_i = \mathbf{Q}_i^{\mathrm{T}}[\mathbf{c}_{i-1} + \mathbf{a}_i - \rho_{i-1}]_{i-1} + [\rho_i]_i. \tag{16}$$

Upon differentiation of equation (15) with respect to time, we derive the following:

(i) if the ith joint is rotational,

$$[\dot{\mathbf{c}}_i]_i = \mathbf{Q}_i^{\mathrm{T}}[\dot{\mathbf{c}}_{i-1} + \omega_{i-1} \times (\mathbf{a}_i - \rho_{i-1})]_{i-1} + [\omega_i \times \rho_i]_i \tag{17}$$

$$\begin{aligned}[\ddot{\mathbf{c}}_i]_i = {} & \mathbf{Q}_i^{\mathrm{T}}[\ddot{\mathbf{c}}_{i-1} + \dot{\omega}_{i-1} \times (\mathbf{a}_i - \rho_{i-1}) + \omega_{i-1} \times (\omega_{i-1} \times (\mathbf{a}_i - \rho_{i-1}))]_{i-1} \\ & + [\dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i)]_i; \end{aligned} \tag{18}$$

(ii) if the ith joint is prismatic,

$$[\dot{\mathbf{c}}_i]_i = \mathbf{Q}_i^{\mathrm{T}}[\dot{\mathbf{c}}_{i-1} + \omega_{i-1} \times (\mathbf{a}_i - \rho_{i-1})]_{i-1} + [\omega_i \times \rho_i - \dot{b}_i\mathbf{e}_i]_i \tag{19}$$

$$\begin{aligned}[\ddot{\mathbf{c}}_i]_i = {} & \mathbf{Q}_i^{\mathrm{T}}[\ddot{\mathbf{c}}_{i-1} + \dot{\omega}_{i-1} \times (\mathbf{a}_i - \rho_{i-1}) + \omega_{i-1} \times (\omega_{i-1} \times (\mathbf{a}_i - \rho_{i-1}))]_{i-1} \\ & + [\dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i) - \ddot{b}_i\mathbf{e}_i - 2\omega_i \times \dot{b}_i\mathbf{e}_i]_i, \end{aligned} \tag{20}$$

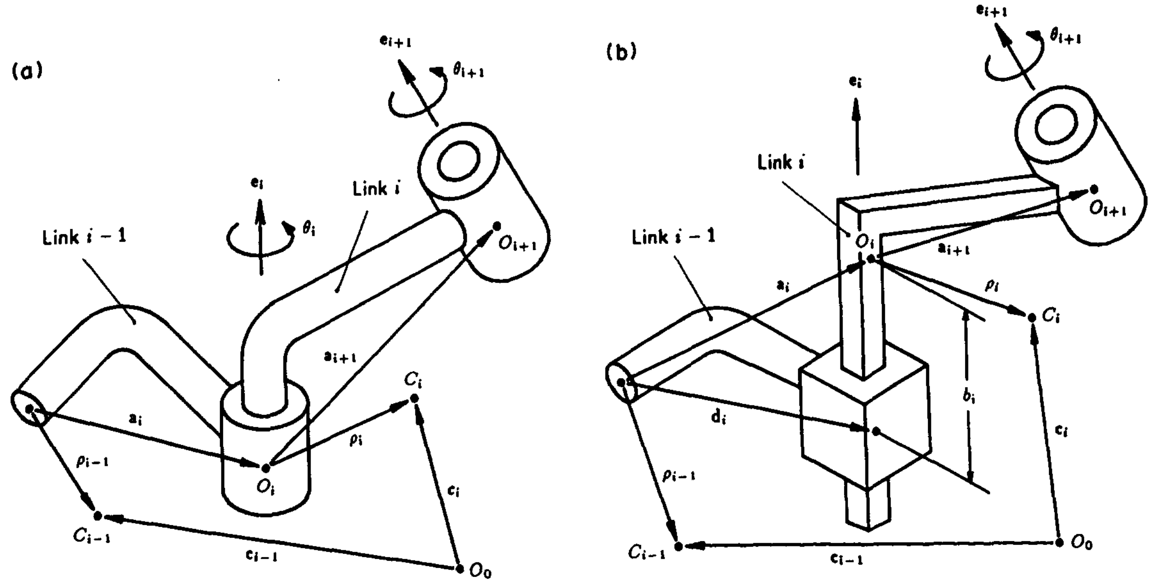

Fig. 2. (a) Successive links coupled by a rotational joint. (b) Successive links coupled by a prismatic joint.

for $i = 1, 2, \ldots, n$, whereas $\dot{\mathbf{c}}_0$ and $\ddot{\mathbf{c}}_0$ are the velocity and acceleration of the mass center of the base link. If the base link is chosen as an inertial reference frame, then,

$$[\dot{\mathbf{c}}_0]_0 = \mathbf{0}, \quad [\ddot{\mathbf{c}}_0]_0 = \mathbf{0}. \tag{21}$$

In deriving equations (17)–(20), the following relations have been used:

$$\dot{\mathbf{a}}_i = \begin{cases} \omega_{i-1} \times \mathbf{a}_i, & \text{if the } i\text{th joint is rotational,} \\ \omega_{i-1} \times \mathbf{a}_i - \dot{b}_i \mathbf{e}_i, & \text{if the } i\text{th joint is prismatic,} \end{cases} \tag{22}$$

$$\omega_{i-1} \times \mathbf{e}_i = \omega_i \times \mathbf{e}_i. \tag{23}$$

4. THE DRIVING TORQUES/FORCES OF A SERIAL-TYPE MANIPULATOR

We consider the active force ϕ consisting of the gravity term ϕ_g, the dissipative-force term ϕ_d, and the driving-force term τ which is supplied by the actuators of the manipulator under study, i.e.

$$\phi = \phi_g + \phi_d + \tau, \tag{24}$$

where, ϕ_g, ϕ_d, τ are the following n-dimensional vectors:

$$\phi_g = \sum_{k=1}^{n} \left(\frac{\partial \dot{\mathbf{c}}_k}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} m_k \mathbf{g} \tag{25}$$

$$\phi_d = [\phi_{d1}, \phi_{d2}, \ldots, \phi_{dn}]^{\mathrm{T}} \tag{26}$$

$$\tau = [\tau_1, \tau_2, \ldots, \tau_n]^{\mathrm{T}} \tag{27}$$

and $\mathbf{g}$ is the gravity-acceleration vector. Moreover, ϕ_{di} is the dissipative torque/force acting at the ith joint, and τ_i is the driving torque/force acting at the ith joint.

Kane's dynamical equations thus become

$$\phi_g + \phi_d + \tau + \phi^* = \mathbf{0}. \tag{28}$$

From the above equations, we derive

$$\tau = -\phi_d - (\phi^* + \phi_g) = -\phi_d + \sum_{j=1}^{n} \left[\left(\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} \dot{\mathbf{h}}_j + \left(\frac{\partial \dot{\mathbf{c}}_j}{\partial \dot{\mathbf{q}}}\right)^{\mathrm{T}} m_j(\ddot{\mathbf{c}}_j - \mathbf{g})\right]. \tag{29}$$

Based on the definition of the HD coordinate systems, introduced in the previous section, the joint variables are chosen as the generalized coordinates, namely.

$$q_i = \begin{cases} \theta_i, & \text{if the } i\text{th joint is rotational,} \\ b_i, & \text{if the } i\text{th joint is prismatic.} \end{cases} \tag{30}$$

As shown in Ref. [19], the partial angular velocities and the partial velocities can be expressed as:

$$\frac{\partial \omega_j}{\partial \dot{\mathbf{q}}} = \mathbf{A}_j, \quad \frac{\partial \dot{\mathbf{c}}_j}{\partial \dot{\mathbf{q}}} = \mathbf{B}_j, \quad \text{for } j = 1, 2, \ldots, n, \tag{31}$$

where $\mathbf{A}_j$ and $\mathbf{B}_j$ are $3 \times n$ matrices defined as follows:

(i) if all the joints are rotational, then

$$\mathbf{A}_j = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_j, \mathbf{0}, \ldots, \mathbf{0}] \tag{32}$$

$$\mathbf{B}_j = [\mathbf{e}_1 \times \mathbf{s}_{j1}, \mathbf{e}_2 \times \mathbf{s}_{j2}, \ldots, \mathbf{e}_j \times \mathbf{s}_{jj}, \mathbf{0}, \ldots, \mathbf{0}]; \tag{33}$$

(ii) if the ith joint is prismatic, for $i \in \{1, 2, \ldots, j\}$, then

$$\mathbf{A}_j = [\mathbf{e}_1, \ldots, \mathbf{e}_{i-1}, \mathbf{0}, \mathbf{e}_{i-1}, \ldots, \mathbf{e}_j, \mathbf{0}, \ldots, \mathbf{0}] \tag{34}$$

$$\mathbf{B}_j = [\mathbf{e}_1 \times \mathbf{s}_{j1}, \ldots, \mathbf{e}_{i-1} \times \mathbf{s}_{j,i-1}, \mathbf{e}_i, \mathbf{e}_{i+1} \times \mathbf{s}_{j,i+1}, \ldots, \mathbf{e}_j \times \mathbf{s}_{jj}, \mathbf{0}, \ldots, \mathbf{0}], \tag{35}$$

where $\mathbf{s}_{ji}$ (for $i = 1, 2, \ldots, j$) is defined as the vector from the ith origin, O_i to the mass center of the jth link, C_j, namely,

$$\mathbf{s}_{ji} = \begin{cases} \mathbf{a}_{i+1} + \cdots + \mathbf{a}_j + p_j, & \text{if } i < j, \\ \rho_j, & \text{if } i = j. \end{cases} \tag{36}$$

Thus, the generalized driving torque/force can be expressed in terms of $\mathbf{A}_j$ and $\mathbf{B}_j$ as follows:

$$\tau = -\phi_d + \sum_{j=1}^{n} [\mathbf{A}_j^{\mathrm{T}}\dot{\mathbf{h}}_j + \mathbf{B}_j^{\mathrm{T}} m_j(\ddot{\mathbf{c}}_j - \mathbf{g})], \tag{37}$$

in which items $\mathbf{A}_j^{\mathrm{T}}\dot{\mathbf{h}}_j$ and $\mathbf{B}_j^{\mathrm{T}} m_j(\ddot{\mathbf{c}}_j - \mathbf{g})$ are given next. If all joints are rotational, then

$$\mathbf{A}_j^{\mathrm{T}}\mathbf{h}_j = \begin{bmatrix} \mathbf{e}_1 \cdot \dot{\mathbf{h}}_j \\ \mathbf{e}_2 \cdot \dot{\mathbf{h}}_j \\ \vdots \\ \mathbf{e}_j \cdot \dot{\mathbf{h}}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{38}$$

$$\mathbf{B}_j^{\mathrm{T}} m_j(\ddot{\mathbf{c}}_j - \mathbf{g}) = \begin{bmatrix} \mathbf{e}_1 \times \mathbf{s}_{j1} \cdot m_j(\ddot{\mathbf{c}}_j - \mathbf{g}) \\ \mathbf{e}_2 \times \mathbf{s}_{j2} \cdot m_j(\ddot{\mathbf{c}}_j - \mathbf{g}) \\ \vdots \\ \mathbf{e}_j \times \mathbf{s}_{jj} \cdot m_j(\ddot{\mathbf{c}}_j - \mathbf{g}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{39}$$

If the ith joint is prismatic, for $1 \leqslant i \leqslant n$, then the ith entry of $\mathbf{A}_j^{\mathrm{T}}\dot{\mathbf{h}}_j$ changes to 0, whereas that of $\mathbf{B}_j^{\mathrm{T}} m_j(\ddot{\mathbf{c}}_j - \mathbf{g})$ changes to $\mathbf{e}_j \cdot m_j(\ddot{\mathbf{c}}_j - \mathbf{g})$. All the other entries remain unchanged.

From equations (38) and (39), it is clear that, if the kth joint is rotational, then the kth component of the first sum appearing in equation (37) is the sum of the projections of the time derivatives of the angular momenta, $\dot{\mathbf{h}}_j$, of the individual links, with respect to their mass centers, onto the axis of the kth joint, for $j = k, k+1, \ldots, n$. Furthermore, the kth component of the second sum of equation (37) is the sum of the projections of the moments of the inertia forces, $m_j\ddot{\mathbf{c}}_j$, with respect to O_k, for $j = k, k+1, \ldots, n$, onto the axis of the kth joint. Finally, the contribution of the gravity forces to the kth component of the generalized driving force is simply the sum of the projections of the moments of the weights, $-m_j\mathbf{g}$, with respect to O_k, for $j = k$, $k+1, \ldots, n$, onto the axis of the kth joint. If the kth joint is prismatic, then the kth component of the first sum appearing in equation (3) vanishes, whereas the second sum of that equation is the sum of the projections of the inertia forces, $m_j\ddot{\mathbf{c}}_j$, on the axis of the kth joint, for $j = k, k+1, \ldots, n$.

Furthermore the dissipative-force term of equation (37) is a function of joint positions $\mathbf{q}$ and joint velocities $\dot{\mathbf{q}}$, i.e.

$$\phi_d = \phi_d(\mathbf{q}, \dot{\mathbf{q}}), \tag{40}$$

which can be computed if the joint coordinates as well as their time derivatives are given.

5. RECURSIVE FORMULAE OF GENERALIZED DRIVING FORCE

Since the dissipative-force term is manipulator dependent, its computation will not be accounted for in this paper. Hence, we set ϕ_d equal to zero. The problem at hand is, thus, to derive efficient formulae to compute each τ_i. First, we define

$$\ddot{\mathbf{p}}_i = \ddot{\mathbf{c}}_i - \mathbf{g}. \tag{41}$$

From the dynamical equation (37), by ignoring the dissipative-force term, we derive the following formulae:

(i) if the ith joint is rotational,

$$\tau_i = \sum_{j=i}^{n} (\mathbf{e}_i \cdot \dot{\mathbf{h}}_j + \mathbf{e}_i \times \mathbf{s}_{ji} \cdot m_j \ddot{\mathbf{p}}_j) = \mathbf{e}_i \cdot \sum_{j=i}^{n} (\dot{\mathbf{h}}_j + \mathbf{s}_{ji} \times m_j \ddot{\mathbf{p}}_j); \tag{42}$$

(ii) if the ith joint is prismatic,

$$\tau_i = \sum_{j=i}^{n} \mathbf{e}_i \cdot m_j \ddot{\mathbf{p}}_j = \mathbf{e}_i \cdot \sum_{j=i}^{n} m_j \ddot{\mathbf{p}}_j. \tag{43}$$

Since the dot product of two vectors is frame invariant, τ_i can be computed in the ith coordinate frame as follows:

(i) if the ith joint is rotational,

$$\tau_i [\mathbf{e}_i]_i \cdot \sum_{j=i}^{n} [\dot{\mathbf{h}}_j + \mathbf{s}_{ji} \times m_j \ddot{\mathbf{p}}_j]_i; \tag{44}$$

(ii) if the ith joint is prismatic,

$$\tau_i = [\mathbf{e}_i]_i \cdot \sum_{j=i}^{n} m_j [\ddot{\mathbf{p}}_j]_i, \tag{45}$$

where vector $[\mathbf{e}_i]_i$ has the simple form given in equation (9).

Notice that, if we compute τ_i backwards, i.e. successively from τ_n to τ_1, then the term $\Sigma[\dot{\mathbf{h}}_j + \mathbf{s}_{ji} \times m_j \ddot{\mathbf{p}}_j]_i$ of equation (44), and the term $\Sigma m_j [\ddot{\mathbf{p}}_j]_i$ of equation (45), can be computed from the $\Sigma[\dot{\mathbf{h}}_j + \mathbf{s}_{j,i+1} \times m_j \ddot{\mathbf{p}}_j]_{i+1}$ and $\Sigma m_j [\ddot{\mathbf{p}}_j]_{i+1}$ terms, respectively [15]. These have already been computed while computing τ_{i+1}, namely,

$$\begin{aligned}
\sum_{j=i}^{n} [\dot{\mathbf{h}}_j + \mathbf{s}_{ji} \times m_j \ddot{\mathbf{p}}_j]_i &= \sum_{j=i+1}^{n} [\dot{\mathbf{h}}_j + (\mathbf{s}_{j,i+1} + \mathbf{a}_{i+1}) \times m_j \ddot{\mathbf{p}}_j]_i + [\dot{\mathbf{h}}_i + \mathbf{s}_{ii} \times m_i \ddot{\mathbf{p}}_i]_i \\
&= \sum_{j=i+1}^{n} [\dot{\mathbf{h}}_j + \mathbf{s}_{j,i+1} \times m_j \ddot{\mathbf{p}}_j]_i + [\mathbf{a}_{i+1}]_i \times \sum_{j=i+1}^{n} m_j [\ddot{\mathbf{p}}_j]_i + [\dot{\mathbf{h}}_i + \mathbf{s}_{ii} \times m_i \ddot{\mathbf{p}}_i]_i \\
&= \mathbf{Q}_{i+1} \sum_{j=i+1}^{n} [\dot{\mathbf{h}}_j + \mathbf{s}_{j,i+1} \times m_j \ddot{\mathbf{p}}_j]_{i+1} + \mathbf{k}_i,
\end{aligned} \tag{46}$$

for $i = n, n-1, \ldots, 1$. where $\mathbf{k}_i$ is defined as

$$\mathbf{k}_i = [\dot{\mathbf{h}}_i]_i + [\mathbf{s}_{ii}]_i \times m_i [\ddot{\mathbf{p}}_i]_i + [a_{i+1}]_i \times \sum_{j=i+1}^{n} m_j [\ddot{\mathbf{p}}_j]_i \tag{47}$$

and

$$[s_{ii}]_i = [\rho_i]_i. \tag{48}$$

Moreover,

$$\sum_{j=i}^{n} m_j [\ddot{\mathbf{p}}_j]_i = m_i [\ddot{\mathbf{p}}_i]_i + \mathbf{Q}_{i+1} \sum_{j=i+1}^{n} m_j [\ddot{\mathbf{p}}_j]_{i+1}. \tag{49}$$

Finally, we explain how to compute $\ddot{\mathbf{p}}_i$. For a serial-type manipulator, it can be readily shown that

$$\ddot{\mathbf{c}}_i = \mathbf{f}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + \ddot{\mathbf{c}}_0, \quad i \in \{1, 2, \ldots, n\}, \tag{50}$$

where $\ddot{\mathbf{c}}_0$ is the acceleration of the origin of the base frame, which is equal to zero if the base is an inertial frame. Substituting equation (50) into equation (41), we derive the expression of $\ddot{\mathbf{p}}_i$ in terms of $\mathbf{f}_i$, that follows:

$$\ddot{\mathbf{p}}_i = \mathbf{f}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) + (\ddot{\mathbf{c}}_0 - \mathbf{g}). \tag{51}$$

Equation (51) means that $\ddot{\mathbf{p}}_i$ is nothing but $\ddot{\mathbf{c}}_i$ when the base frame undergoes an additional acceleration of $-\mathbf{g}$. Hence, $\ddot{\mathbf{p}}_i$ can be computed using the same formulae which are used for the computation of $\ddot{\mathbf{c}}_i$, just by setting $\ddot{\mathbf{p}}_0 = -\mathbf{g}$, a technique already introduced by Luh *et al.* [2].

6. DESCRIPTION OF THE ALGORITHM AND ITS COMPLEXITY ANALYSIS

The algorithm can be divided into three stages. Each stage consists of several steps. Each step is a complete computational scheme with its complexity appearing to the right of it. The symbols used in this section are defined as follows:

n—number of one-degree-of-freedom joints;
m, a—units of multiplication and addition, respectively, e.g. $4m + 2a$ means 4 multiplications and 2 additions;
M_k—number of multiplications required in the kth stage;
A_k—number of additions required in the kth stage;
R—revolute joint which allows rotation only;
P—prismatic joint which allows translation only;

The algorithm and its complexity are analyzed as follows:
Stage 1. Computation of rotation matrices and translation vectors.

```
for $i \leftarrow 1$ until $n$ step 1 do
    if $P$ then
        $[\mathbf{a}_i]_{i-1} \leftarrow \begin{pmatrix} a_i \\ -b_i \sin\alpha_i \\ b_i \cos\alpha_i \end{pmatrix}$        $\ldots 2m + 0a$
enddo
```

If the ith joint is rotational, $[\mathbf{a}_i]_{i-1}$ can be calculated off-line since it is constant. Hence, zero multiplications and zero additions are required in this case. To reduce the computational complexity, matrices $\mathbf{Q}_i$ are not computed in this stage. Instead, the product of $\mathbf{Q}_i$ times an arbitrary three-dimensional array can be computed using an efficient scheme, as shown in the next stage. Hence, no computational cost is taken into account with regard to matrices $\mathbf{Q}_i$.

Thus, for this stage, one has

$$M_1 = 2p, \quad A_1 = 0 \tag{52}$$

Stage 2. Kinematical computations.

In this stage, the following 3×3 tensors are introduced:

$$\mathbf{\Omega}_i = \frac{\partial \omega_i \times \mathbf{v}}{\partial \mathbf{v}}, \quad \dot{\mathbf{\Omega}}_i = \frac{\partial \dot{\omega}_i \times \mathbf{v}}{\partial \mathbf{v}}, \quad \mathbf{W}_i = \dot{\mathbf{\Omega}}_i + \mathbf{\Omega}_i^2, \tag{53}$$

where $\mathbf{v}$ is an arbitrary three-dimensional vector. It is clear that both $\mathbf{\Omega}_i$ and $\dot{\mathbf{\Omega}}_i$ are skew symmetric matrices. Now,

```
$[\mathbf{e}_1]_1 \leftarrow [0, 0, 1]^T$                                  $\ldots 0m + 0a$
$[\omega_1]_1 \leftarrow \dot{\theta}_1 [\mathbf{e}_1]_1$                  $\ldots 0m + 0a$
$[\dot{\omega}_1]_1 \leftarrow \ddot{\theta}_1 [\mathbf{e}_1]_1$           $\ldots 0m + 0a$
$[\mathbf{W}_1]_1 \leftarrow \dot{\mathbf{\Omega}}_1 + \mathbf{\Omega}_1^2$   $\ldots 1m + 0a$
$[\ddot{\mathbf{p}}_1]_1 \leftarrow -\mathbf{Q}_1^T [\mathbf{g}]_0 + [\mathbf{W}_1 \rho_1]_1$   $\ldots 12m + 9a$
```

```
for $i \leftarrow 2$ until $n$ step 1 do
        $[\mathbf{e}_i]_i \leftarrow [0, 0, 1]^T$                                                    ...$0m + 0a$
        $[\omega_i]_i \leftarrow \mathbf{Q}_i^T[\omega_{i-1}]_{i-1} + [\dot{\theta}_i \mathbf{e}_i]_i$                      ...$8m + 5a$
        $[\dot{\omega}_i]_i \leftarrow \mathbf{Q}_i^T[\dot{\omega}_{i-1}]_{i-1} + [\mathbf{\Omega}_i \dot{\theta}_i \mathbf{e}_i + \ddot{\theta}_i \mathbf{e}_i]_i$             ...$10m + 7a$
        $[\mathbf{W}_i]_i \leftarrow \dot{\mathbf{\Omega}}_i + \mathbf{\Omega}_i^2$                                          ...$6m + 9a$
        $[\ddot{\mathbf{p}}_i]_i \leftarrow \mathbf{Q}_i^T[\ddot{\mathbf{p}}_{i-1} - \mathbf{W}_{i-1}(\mathbf{a}_i - \rho_{i-1})]_{i-1} + [\mathbf{W}_i \rho_i]_i$   ...$26m + 25a$
    if $P$ then
        $[\ddot{\mathbf{p}}_i]_i \leftarrow [\ddot{\mathbf{p}}_i]_i - [2\mathbf{\Omega}_i \dot{b}_i \mathbf{e}_i + \ddot{b}_i \mathbf{e}_i]_i$                  ...$2m + 4a$
    endif
enddo
```

The product of matrix $\mathbf{Q}_i$ times an arbitrary three-dimensional vector $\mathbf{v} = [v_1, v_2, v_3]^T$ is performed as shown below:

$$\mathbf{Q}_i \mathbf{v} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} v_1 \cos\theta_i - v_2 \sin\theta_i \\ u \cos\alpha_i - v_3 \sin\alpha_i \\ u \sin\alpha_i + v_3 \cos\alpha_i \end{pmatrix}, \tag{54}$$

where u is an intermediate variable defined as

$$u = v_1 \sin\theta_i + v_2 \cos\theta_i. \tag{55}$$

From equations (54) and (55), it is apparent that computing $\mathbf{Q}_i \mathbf{v}$ requires only eight multiplications and four additions. Similarly, computing $\mathbf{Q}_i^T \mathbf{v}$ requires the same numbers of operations. Moreover $[\mathbf{e}_i]_i$ has the simple form of $[0, 0, 1]^T$, and hence, any vector times $[\mathbf{e}_i]_i$ requires zero multiplications and zero additions.

Thus, for this stage, one has

$$M_2 = 50n + 2p - 47, \quad A_2 = 46n + 2p - 47. \tag{56}$$

Stage 3. Computation of the generalized driving force τ.

In this stage, we define the following three intermediate vector variables:

$$\mathbf{u}_i = m_i [\ddot{\mathbf{p}}_i]_i$$

$$\mathbf{v}_i = \sum_{j=i}^{n} m_j [\ddot{\mathbf{p}}_j]_i$$

$$\mathbf{w}_i = \sum_{j=i}^{n} [\dot{\mathbf{h}}_j + \mathbf{s}_{ji} \times m_j \ddot{\mathbf{p}}_j]_i.$$

```
        $\mathbf{v}_{n+1} \leftarrow \mathbf{w}_{n+1} \leftarrow \mathbf{0}$                                          ...$0m + 0a$
for $i \leftarrow n$ until 1 step $-1$ do
        $[\dot{\mathbf{h}}_i]_i \leftarrow [\mathbf{I}_i]_i [\dot{\omega}_i]_i + [\mathbf{\Omega}_i]_i [\mathbf{I}_i]_i [\omega_i]_i$                 ...$24m + 18a$
        $\mathbf{u}_i \leftarrow m_i [\ddot{\mathbf{p}}_i]_i$                                              ...$3m + 0a$
        $\mathbf{v}_{i+1} \leftarrow \mathbf{Q}_{i+1} \mathbf{v}_{i+1}$                                         ...$8m + 4a$
        $\mathbf{k}_i \leftarrow [\dot{\mathbf{h}}_i]_i + [\rho_i]_i \times \mathbf{u}_i + [\mathbf{a}_{i+1}]_i \times \mathbf{v}_{i+1}$              ...$12m + 12a$
        $\mathbf{w}_i \leftarrow \mathbf{Q}_{i+1} \mathbf{w}_{i+1} + \mathbf{k}_i$                                     ...$8m + 7a$
        $\mathbf{v}_i \leftarrow \mathbf{v}_{i+1} + \mathbf{u}_i$                                           ...$0m + 3a$
        if $R$ then
                $\tau_i \leftarrow [\mathbf{e}_i]_i \cdot \mathbf{w}_i$                              ... $0m + 0a$
        else
                $\tau_i \leftarrow [\mathbf{e}_i]_i \cdot \mathbf{v}_i$                              ... $0m + 0a$
        endif
    enddo
```

So far, one has

$$M_3 = 55n - 38, \quad A_3 = 44n - 38,$$

where the term -38 appears in both expressions of M_3 and A_3 because neither multiplications nor additions are required for computing $\mathbf{Q}_{n+1}\mathbf{v}_{n+1}$, $\mathbf{Q}_{n+1}\mathbf{w}_{n+1}$, and $[\mathbf{a}_{n+1}]_n \times \mathbf{v}_{n+1}$, since $\mathbf{v}_{n+1} = \mathbf{w}_{n+1} = \mathbf{0}$. Moreover, only eight multiplications and two additions are required for computing $[\dot{\mathbf{h}}_1]_1$, since $[\omega_1]_1 = [0, 0, \theta_1]^{\mathrm{T}}$ and $[\dot{\omega}_1]_1 = [0, 0, \ddot{\theta}_1]^{\mathrm{T}}$

Furthermore, τ_i is equal to the third component of either $\mathbf{w}_i$ or $\mathbf{v}_i$. Hence, when $i = n$, it is unneccessary to compute all components of $\mathbf{w}_i$ and $\mathbf{v}_i$, as well as other terms involved. Upon this consideration, the numbers of operations of the last cycle of the do loop in this stage can be reduced to only $15m + 10a$. Therefore, the total numbers of operations of this stage are further reduced as follows:

$$M_3 = 55n - 62, \quad A_3 = 44n - 58. \tag{57}$$

Based on the previous step-level analysis, we arrive at the following general formulae:

$$M = \sum_{k=1}^{3} M_k = 105n - 109 \tag{58}$$

$$A = \sum_{k=1}^{3} M_k = 90n - 105, \tag{59}$$

and M and A are the total numbers of multiplications and additions, respectively, of the overall algorithm.

The present algorithm is compared with others, from the viewpoint of computational complexity, in Table 1. In that table, EL and NE represent Euler–Lagrange and Newton–Euler formulations, respectively. From that table, one can see that the most efficient methods arise from either the recursive NE formulation or the Kane formulation. One can also see that the numbers of both multiplications and additions of the algorithm presented in this paper are less than those of the most efficient algorithm in the literature, namely, the one introduced by Khalil *et al.* [10]. In addition, Khalil *et al.* assumed in their HD notation that the Z_{n-1} and Z_n axes, as well as the O_{n-1} and O_n origins, are coincident. With this assumption, as mentioned in their paper, the complexity is reduced. It can be easily shown that, if a similar assumption is made with the HD notation used in this paper, i.e. assuming that, referring to Fig. 1(a), the Z_0 and Z_1 axes, as well as the O_0 and O_1 origins are coincident, the number of multiplications and additions for a six-axis manipulator of the algorithm presented here will be further reduced to, for a six-joint manipulator, 513 and 432, respectively.

Table 1. Complexities of several different algorithms

Author(s)	Method	Multiplications	Additions	Multiplications ($n = 6$)	Additions ($n = 6$)
Hollerbach [3]	EL	$412n - 277$	$320n - 201$	2195	1719
Luh *et al.* [2]	NE	$150n - 48$	$131n - 48$	852	738
Walker and Orin [20]	NE	$137n - 22$	$101n - 11$	800	595
Hollerbach [21]	NE	(not available)		688	558
Kane and Levinson [13]	Kane	(not available)		646	394
Khalil *et al.* [10]	NE	$105n - 92$	$94n - 86$	538	478
This paper's	Kane	$105n - 109$	$90n - 105$	521	435

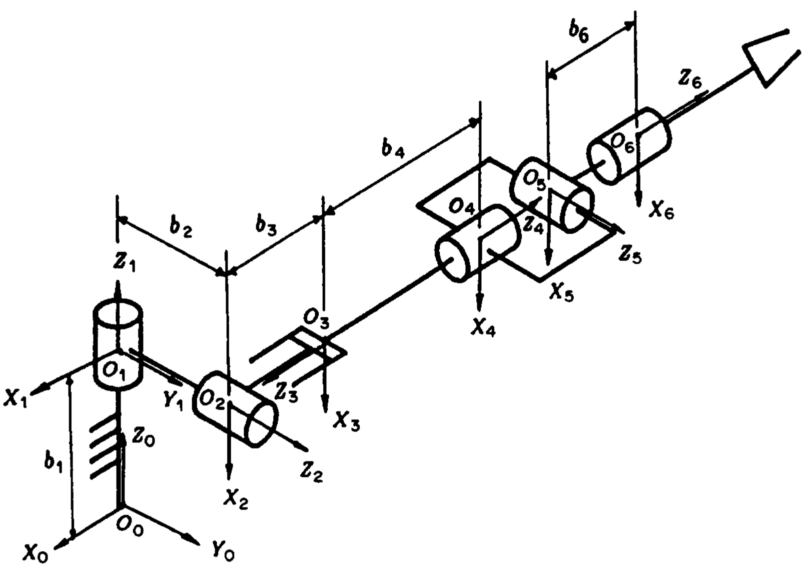

Fig. 3. Stanford Arm with HD notation.

7. EXAMPLES

Two examples are now introduced to illustrate the application of this algorithm.

7.1. Example 1

In the first example, a Stanford Arm whose data and joint trajectories appeared in Ref. [3] is analyzed. The HD parameters of the manipulator, as shown in Fig. 3, are listed in Table 2.

The inertial properties of the manipulator are listed as follows:

$$m_1 = 9.0,\quad [\rho_1]_1 = \begin{pmatrix} 0 \\ 0 \\ -0.1 \end{pmatrix},\quad [\mathbf{I}_1]_1 = \begin{pmatrix} 0.02 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix};$$

$$m_2 = 6.0,\quad [\rho_2]_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},\quad [\mathbf{I}_2]_2 = \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.06 \end{pmatrix};$$

$$m_3 = 4.0,\quad [\rho_3]_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},\quad [\mathbf{I}_3]_3 = \begin{pmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.01 \end{pmatrix};$$

$$m_4 = 1.0,\quad [\rho_4]_4 = \begin{pmatrix} 0 \\ 0 \\ 0.1 \end{pmatrix},\quad [\mathbf{I}_4]_4 = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.0005 & 0 \\ 0 & 0 & 0.001 \end{pmatrix};$$

Table 2. HD parameters of the Stanford Arm

Joint i	α_i (deg)	a_i (m)	b_i (m)	Initial θ_i (deg)
1	0	0	−0.1	0
2	−90	0	−0.1	90
3	−90	0	0	0
4	0	0	−0.6	0
5	90	0	0	0
6	−90	0	0	0

$$m_5 = 0.6, \quad [\rho_5]_5 = \begin{pmatrix} 0 \\ 0.06 \\ 0 \end{pmatrix}, \quad [\mathbf{I}_5]_5 = \begin{pmatrix} 0.0005 & 0 & 0 \\ 0 & 0.0002 & 0 \\ 0 & 0 & 0.0005 \end{pmatrix};$$

$$m_6 = 0.5, \quad [\rho_6]_6 = \begin{pmatrix} 0 \\ 0 \\ 0.2 \end{pmatrix}, \quad [\mathbf{I}_6]_6 = \begin{pmatrix} 0.003 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.002 \end{pmatrix},$$

where the units of m_i, ρ_i and $\mathbf{I}_i$ are $(\mathrm{N \cdot s^2/m})$, (m) and $(\mathrm{N \cdot s^2 \cdot m})$, respectively.

The gravity acceleration used is

$$[\mathbf{g}]_0 = [0, 0, -9.81]^{\mathrm{T}} \ (\mathrm{m/s^2}).$$

The joint trajectories of interest are defined as follows:

$$\begin{cases} \theta_i = \theta_i(0) + A_i[t - \sin(Bt)/B], & \text{if } i = 1, 2, 4, 5, 6; \\ b_3 = A_3[t - \sin(Bt)/B], & \text{if } i = 3, \end{cases}$$

where

$$\theta_i(0) = \begin{cases} 0 & \text{if } 1, 4, 5, 6, \\ \pi/2, & \text{if } i = 2, \end{cases}$$

$$A_i = \begin{cases} \pi/30, & \text{if } i = 1, 4, 5, 6 \\ -\pi/60, & \text{if } i = 2, \\ 0.01, & \text{if } i = 3, \end{cases}$$

$$B = 2\pi/10,$$

and time t varies from 0 to 10 s. The joint trajectories defined above are shown in Fig. 4 and the computed driving torques and force are shown in Fig. 5.

7.2. *Example 2*

In this example, a six-axis, revolute-coupled industrial manipulator whose data appeared in Ref. [4] is analyzed. The HD parameters of the manipulator are listed in Table 3. The Cartesian trajectory of interest is a straight line defined as follows:

$$x = 1.33, \quad y = s(t), \quad z = 1.80,$$

where $s(t)$ is the normal spline introduced by Angeles *et al.* [22], which has zero initial and final velocities and accelerations, as shown in Fig. 6. The end effector is planned to move along the said trajectory without any orientation changes over a time period of one second.

The inertial properties of the manipulator are listed as follows:

$$m_1 = 680, \quad [\rho_1]_1 = \begin{pmatrix} 0 \\ -0.33 \\ 0 \end{pmatrix}, \quad [\mathbf{I}_1]_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 62 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$m_2 = 360, \quad [\rho_2]_2 = \begin{pmatrix} -0.87 \\ 0 \\ -0.13 \end{pmatrix}, \quad [\mathbf{I}_2]_2 = \begin{pmatrix} 11 & 0 & 0 \\ 0 & 53 & 0 \\ 0 & 0 & 44 \end{pmatrix}$$

$$m_3 = 180, \quad [\rho_3]_3 = \begin{pmatrix} -0.64 \\ 0.04 \\ 0 \end{pmatrix}, \quad [\mathbf{I}_3]_3 = \begin{pmatrix} 1.1 & 0 & 0 \\ 0 & 44 & 0 \\ 0 & 0 & 44 \end{pmatrix}$$

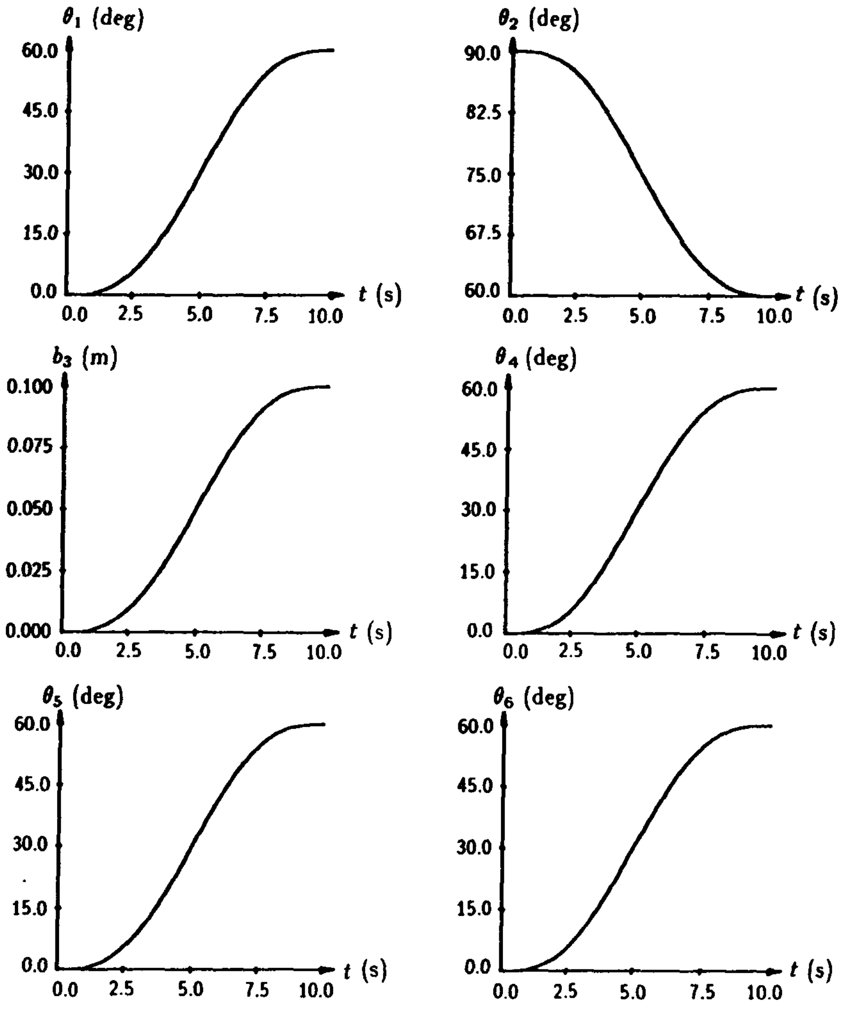

Fig. 4. Joint trajectories of Example 1.

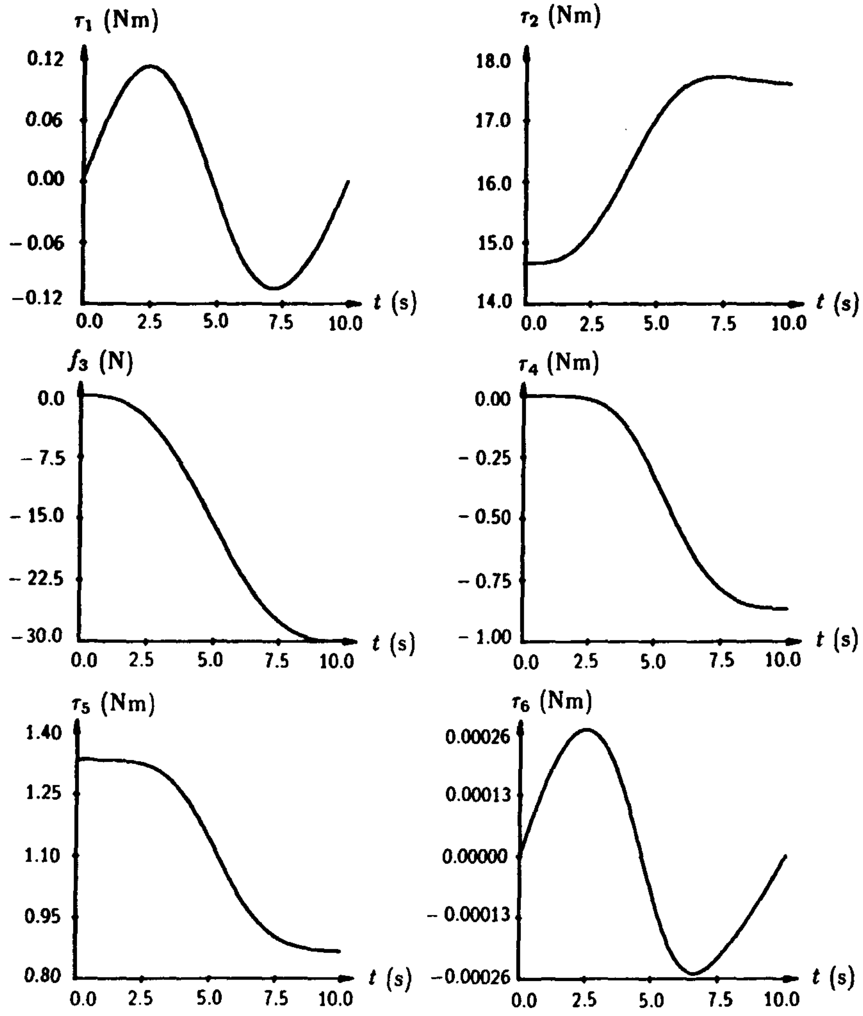

Fig. 5. Driving torques and force of Example 1.

Table 3. HD parameters of the manipulator

Joint i	α_i (deg)	a_i (m)	b_i (m)	Initial θ_i
1	0	0	−1.5	0
2	90	0	0	68.5
3	0	1.02	0	−135
4	0	1.02	0	39.6
5	90	0.2	0	90
6	90	0	−0.41	90

$$m_4 = 55, \quad [\rho_4]_4 = \begin{pmatrix} -0.12 \\ 0.04 \\ 0 \end{pmatrix}, \quad [\mathbf{I}_4]_4 = \begin{pmatrix} 0.44 & 0 & 0 \\ 0 & 0.91 & 0 \\ 0 & 0 & 0.82 \end{pmatrix}$$

$$m_5 = 36, \quad [\rho_5]_5 = \begin{pmatrix} 0 \\ -0.05 \\ -0.08 \end{pmatrix}, \quad [\mathbf{I}_5]_5 = \begin{pmatrix} 0.47 & 0 & 0 \\ 0 & 0.38 & 0 \\ 0 & 0 & 0.18 \end{pmatrix}$$

$$m_6 = 68, \quad [\rho_6]_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad [\mathbf{I}_6]_6 = \begin{pmatrix} 0.44 & 0 & 0 \\ 0 & 0.64 & 0 \\ 0 & 0 & 0.73 \end{pmatrix},$$

where the units of m^i, ρ_i and $\mathbf{I}_i$ are $(\mathrm{N \cdot s^2/m})$, (m) and $(\mathrm{N \cdot s^2 \cdot m})$, respectively.

The gravity acceleration used is

$$[\mathbf{g}]_0 = [0, 0, -9.81]^{\mathrm{T}} \ (\mathrm{m/s^2}).$$

Since the manipulator of interest is not wrist-partitioned, its inverse kinematic analysis is not straightforward. The required joint trajectories, as shown in Fig. 7 were thus computed numerically using KINVERS, a computer code for a manipulator's forward and inverse kinematics developed at McRCIM [23]. The driving torques computed with this algorithm are shown in Fig. 8.

8. CONCLUSIONS

In this paper, an efficient algorithmical approach to the inverse dynamics of general serial-type robotic manipulators was presented. The formulation is based on Kane's dynamical equations. Such as the EL formulation, Kane's considers the whole multibody system as one integral entity and uses the concepts of generalized coordinates and forces without considering any

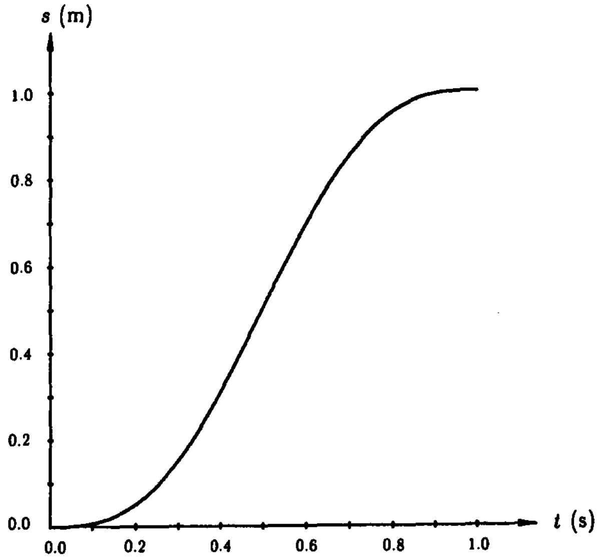

Fig. 6. Spline defining Y coordinate for Example 2.

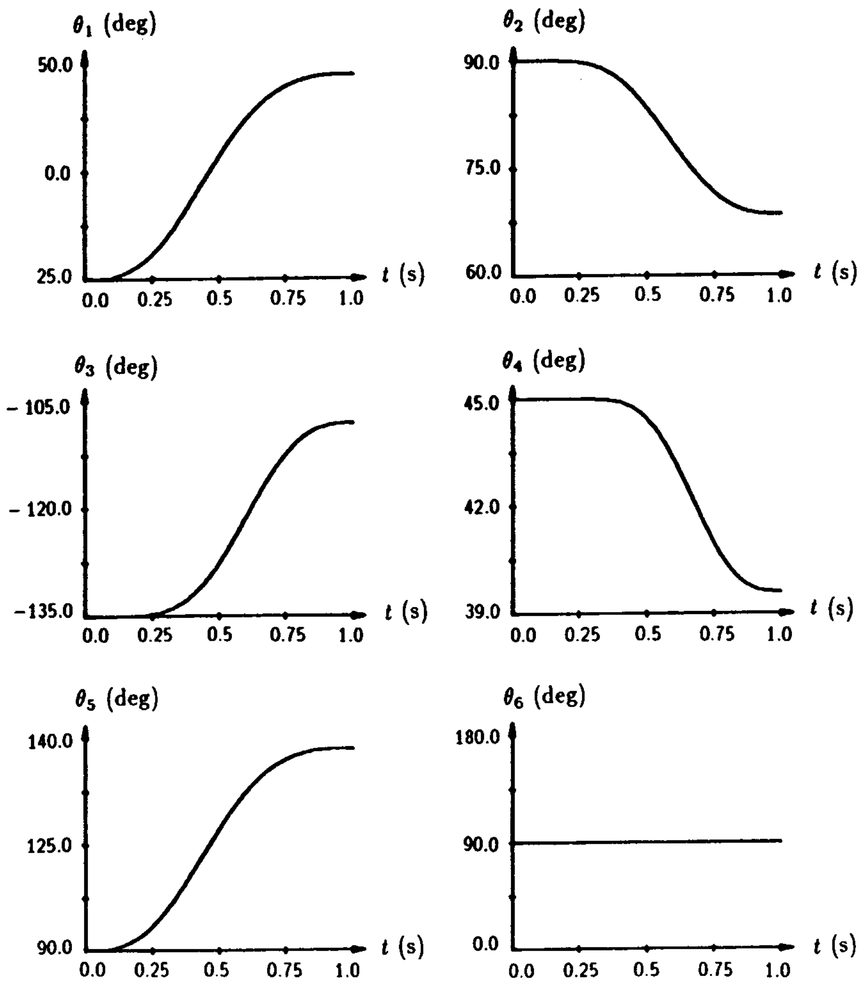

Fig. 7. Joint trajectories of Example 2.

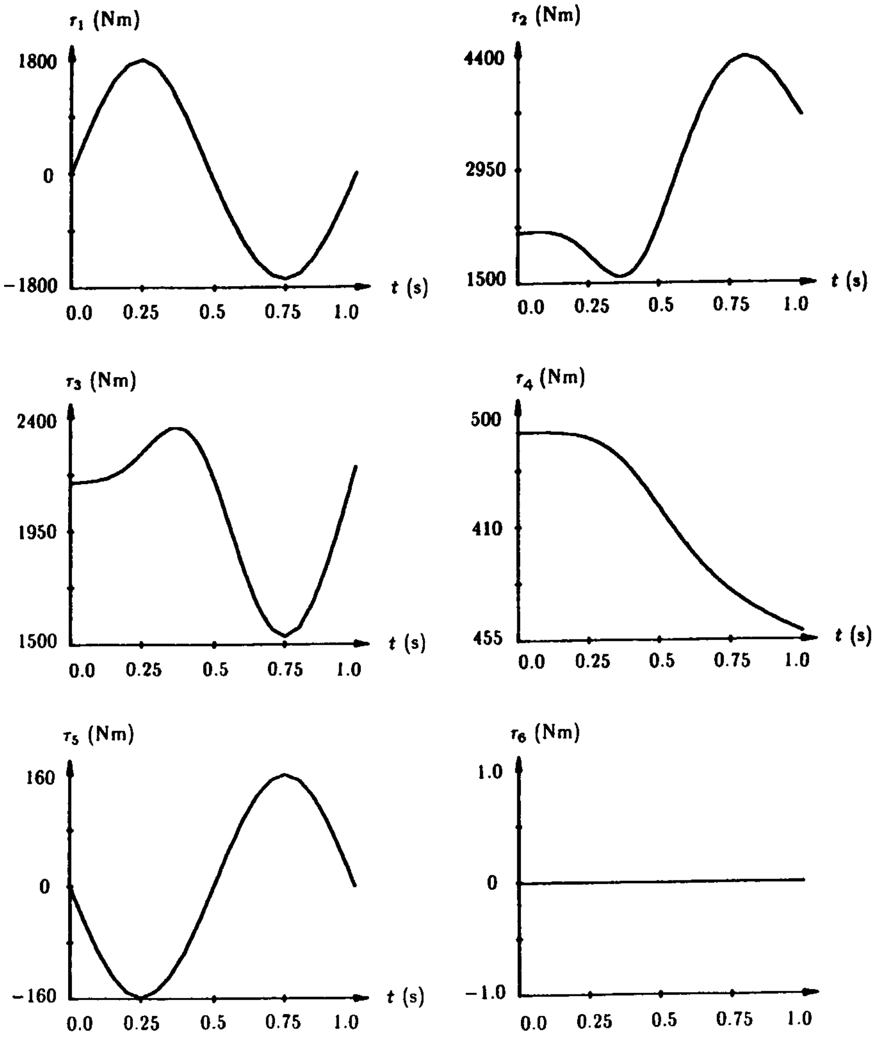

Fig. 8. Driving torques of Example 2.

nonworking constraint forces, which leads to a systematical approach to the problem at hand. Moreover, in Kane's equations the n components of the generalized driving force appear uncoupled in n equations, such as in the EL equations. However, the coefficients of Kane's equations, coupled by the partial velocities and partial angular velocities, are more suitable for recursive computations than those of the EL equations because, in the latter, the said coupling is stronger. Therefore, the algorithm based on Kane's equations can be as efficient as that using the NE equations, and can be more efficient than that using the EL equations.

Acknowledgements—The research work reported here was possible under FCAR (Funds pour la Formation de Chercheurs et l'Aide á la Recherche, of Québec) Grant No. EQ 3072. Mr O. Ma also acknowledges the support of McGill University through the David Stewart Memorial Fellowship.

REFERENCES

1. J. J. Uicker, On the dynamics analysis of spatial linkages using 4 × 4 matrices, Ph.D. Thesis, Northwestern Univ., Ill. (1965).
2. J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, On-line computational scheme for mechanical manipulators. *J. Dynamic Syst. Meas. Control* **102,** 103–110 (1980).
3. J. M. Hollerbach, A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Syst. Man. Cybern* **SMC-10**(11), 730–736 (1980).
4. M. Thomas and D. Tesar, Dynamic modeling of serial manipulator arms. *J. Dynamic Syst. Meas. Control* **104,** 218–228 (1982).
5. M. R. Driels and Z. H. Yang. Alternative methods for the analysis of robot arm dynamics. *J. Robotics Syst.* **1,** 351–378 (1984).
6. C. P. Neuman, and J. J. Murray, Computational robot dynamics: functions and applications. *J. Robotics Syst.* **2.** 425–452 (1985).
7. L. T. Wang and B. Ravani, Recursive computations of kinematic and dynamic equations for mechnical manipulators. *IEEE J. Robotics Automn* **3,** 124–131 (1985).
8. T. Wang and D. Kohli, Closed and expanded form of manipulator dynamics using Lagrangian approach. *J. Mech. Trans. Auto. Design* **107,** 223–225 (1985).
9. M. C. Leu and N. Hemati, Automated symbolic derivation of dynamic equations of motion for robotic manipulators. *Trans. ASME* **108,** 172–179 (1986).
10. W. Khalil, J. F. Kleinfinger and M. Gautier, Reducing the computational burden of the dynamic models of robots. *Proc. IEEE Conf. Robotics and Automation*, San Francisco, Vol. 1. pp. 525–531 (1986).
11. M. Renaud, Quasi-minimal computation of the dynamic model of a robot manipulator utilizing the Newton–Euler formulism and the notion of augmented body. *Proc. IEEE Conf. Robotics Automn* Raleigh, Vol. 3, pp. 1677–1682 (1987).
12. W. M. Silver, On the equivalence of Lagrangian and Newton–Euler dynamics for manipulators. *Int. J. Robotics Res.* **1,** 118–128 (1982).
13. T. R. Kane, and D. A. Levinson, The use of Kane's dynamical equations in robotics. *Int. J. Robotics Res* **2,** 3–21 (1983).
14. T. R. Kane, Dynamics of nonholonomic systems. *Trans. ASME J. appl. Mech.* **83,** 574–578 (1961).
15. O. Ma. Dynamics of serial-type robotic manipulators. M. Eng. Thesis. Department of Mechanical Enginering and McRCIM, McGill University, Montreal (1987).
16. I. J. Wittenburg, *Dynamics of Systems of Rigid Bodies.* Teubner Stuttgart, Germany (1977).
17. T. R. Kane, P. W. Likins, and D. A. Levinson, *Spacecraft Dynamics.* McGraw-Hill, New York (1983).
18. R. S. Hartenberg and J. Denavit, A kinematic notation for lower pair mechanisms. *J. appl. Mech.* **22,** 215–221 (1955).
19. J. Angeles, On the numerical solution of the inverse kinematic problem. *Int. J. Robotics Res.* **4,** 21–37 (1985).
20. M. W. Walker and D. E. Orin, Efficient dynamic computer simulation of robotic mechanisms. *J. Dynamic Syst. Meas. Control* **104,** 205–211 (1982).
21. J. M. Hollerbach and G. Sahar, Wrist-partitioned inverse kinematic accelerations and manipulator dynamics. *Int. J. Robotics Res.* **2,** 61–76 (1983).
22. J. Angeles, A. Alivizatos and P. J. Zsombor-Murray, The synthesis of smooth trajectories for pick-and-place operations. *IEEE Trans. Syst. Man Cybern.* **18**(1), 173–178 (1988).
23. J. Angeles, K. Anderson and C. S. López-Cajún, KINVERS: An integrated computer package for robot manipulators' forward and inverse kinematics. *Proc. AMSE Int. Comput. Engng Conf.*, New York, Vol. 1, 251–258 (1987).
24. H. Faessler, Computer-assisted generation of dynamical equations for multibody systems. *Int. J. Robotics Res.* **5,** 129–141 (1986).