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.

I. 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 architec-
tures, 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
\( n \)-axis linear, is 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,
\( O(p^3) \), 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
directly [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 suitable 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 $\{q\}^n$, which is then grouped in the $n$-dimensional vector of generalized coordinates $q$. The generalized speeds are defined simply as the time derivatives of $q$. Let $\dot{q_i}, \ddot{q}_i, \omega_k$, and $\dot{\omega}_k$ be the velocity and acceleration of the mass center and the angular velocity and acceleration of the $k$th rigid-body of the system, respectively. Then the contribution of this body to the generalized inertia force of the system is defined as

$$I \ddot{q} = -\sum_{k=1}^{n} \left[ \frac{\partial \dot{q}_k}{\partial q} \right]^T m_k \ddot{q}_k - \left[ \frac{\partial \omega_k}{\partial q} \right]^T h_k,$$

(1)

where $\phi^*$ is an $n$-dimensional vector and $h_k$ is the time derivative of the angular momentum of the $k$th body about its mass center, i.e.

$$h_k = I_k \dot{\omega}_k + \omega_k \times I_k \omega_k,$$

(2)

where $I_k$ is the centroidal inertia tensor of the $k$th body. Moreover, $\partial \dot{q}_i / \partial q$ and $\partial \omega_k / \partial q$, 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

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

(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 $f_k$ acting at the mass center of the body, and a moment $n_k$, then the n-dimensional vector of generalized force, $\phi_k$, associated with the kth body, is defined as

$$\phi_k = \left( \frac{\partial \mathbf{e}_k}{\partial \mathbf{q}} \right)^T f_k + \left( \frac{\partial \omega_k}{\partial \mathbf{q}} \right)^T n_k.$$  

(4)

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

$$\phi = \sum_{k=1}^{n} \left[ \left( \frac{\partial \mathbf{e}_k}{\partial \mathbf{q}} \right)^T f_k + \left( \frac{\partial \omega_k}{\partial \mathbf{q}} \right)^T n_k \right],$$

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

$$\phi + \phi^* = 0.$$  

(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:

- $\alpha_i$—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;
- $\theta_i$—angle from the $X_{i-1}$ to $X_i$ axes in the direction of the $Z_i$ axis.

Among these parameters, $\theta_i$ 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.](image-url)
In this paper, unless otherwise indicated, any vector or matrix represented in the $i$th coordinate frame is enclosed in brackets and subscripted accordingly, i.e. $[\cdot]_i$.

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

$$Q_i \equiv [Q_i]_{i-1} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & 0 \\ \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i \end{pmatrix}$$

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

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

Moreover, the unit vector in the direction of the axis of the $i$th joint has the following components:

$$[e_i] = [0, 0, 1]^T.$$

### 3.2. Angular velocities and accelerations

The angular velocity and acceleration of the $i$th link can be computed recursively as follows:

$$\omega_i = \begin{cases} \omega_{i-1} + \dot{\theta}_i e_i, & \text{if the } i\text{th joint is rotational,} \\ \omega_{i-1}, & \text{if the } i\text{th joint is prismatic,} \end{cases}$$

$$\dot{\omega}_i = \begin{cases} \dot{\omega}_{i-1} + \ddot{\theta}_i e_i + \dot{\theta}_i \epsilon_i, & \text{if the } i\text{th joint is rotational,} \\ \dot{\omega}_{i-1}, & \text{if the } i\text{th joint is prismatic,} \end{cases}$$

for $i = 1, 2, \ldots, n$, whereas $\omega_0$ 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 $i$th link
An algorithm for the solution of inverse dynamics

will be expressed in the \( i \)th coordinate frame. Hence, the angular velocities and accelerations are computed recursively as follows:

\[
\begin{align*}
[\omega_i] & = \begin{cases} 
Q_i^T[\omega_{i-1}]_{i-1} + \theta_i[e_i], & \text{if the } i \text{th joint is rotational}, \\
Q_i^T[\alpha_{i-1}]_{i-1}, & \text{if the } i \text{th joint is prismatic},
\end{cases} \\
[\dot{\omega}_i] & = \begin{cases} 
Q_i^T[\dot{\omega}_{i-1}]_{i-1} + [\omega_i \times \theta_i e_i + \theta_i^T e_i], & \text{if the } i \text{th joint is rotational}, \\
Q_i^T[\dot{\omega}_{i-1}]_{i-1}, & \text{if the } i \text{th joint is prismatic}.
\end{cases}
\end{align*}
\]

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

\[
[\omega_0] = 0, \quad [\dot{\omega}_0] = 0.
\]

3.3. Velocities and accelerations of the mass centers

Let \( c_i \) be the position vector of the mass center, \( C_i \), of the \( i \)th link, \( \rho_i \) 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:

\[
c_i = c_{i-1} + \rho_i + \rho_i,
\]

or, in the \( i \)th coordinate frame,

\[
[c_i] = Q_i^T[c_{i-1} + \rho_i - \rho_i]_{i-1} + [\rho_i].
\]

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

(i) if the \( i \)th joint is rotational,

\[
[\dot{c}_i] = Q_i^T[\dot{c}_{i-1} + \omega_{i-1} \times (a_i - \rho_{i-1})]_{i-1} + [\omega_i \times \rho_i],
\]

(ii) if the \( i \)th joint is prismatic,

\[
[\dot{c}_i] = Q_i^T[\dot{c}_{i-1} + \omega_{i-1} \times (a_i - \rho_{i-1}) + \omega_{i-1} \times (a_i - \rho_{i-1})]_{i-1} + [\dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i)]_i.
\]

\[
\begin{align*}
\begin{align*}
[\ddot{c}_i] & = Q_i^T[\ddot{c}_{i-1} + \omega_{i-1} \times (a_i - \rho_{i-1}) + \omega_{i-1} \times (a_i - \rho_{i-1})]_{i-1} + [\dot{\omega}_i \times \rho_i + \omega_i \times (\omega_i \times \rho_i) - \dot{\theta}_i e_i - 2\omega_i \times \dot{\theta}_i e_i],
\end{align*}
\end{align*}
\]

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 \( \dot{\mathbf{e}}_0 \) and \( \ddot{\mathbf{e}}_0 \) 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,

\[
[\dot{\mathbf{e}}_0]_0 = \mathbf{0}, \quad [\ddot{\mathbf{e}}_0]_0 = \mathbf{0}.
\]

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

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

(22)

\[
\omega_{i-1} \times \mathbf{e}_i = \omega_i \times \mathbf{e}_i.
\]

(23)

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

We consider the active force \( \mathbf{a} \) consisting of the gravity term \( \mathbf{a}_{bg} \), the dissipative-force term \( \mathbf{a}_{bd} \), and the driving-force term \( \mathbf{a} \) which is supplied by the actuators of the manipulator under study, i.e.

\[
\mathbf{a} = \mathbf{a}_{bg} + \mathbf{a}_{bd} + \mathbf{a},
\]

(24)

where, \( \mathbf{a}_{bg}, \mathbf{a}_{bd}, \mathbf{a} \) are the following \( n \)-dimensional vectors:

\[
\mathbf{a}_{bg} = \begin{bmatrix}
d\mathbf{e}_1, d\mathbf{e}_2, \ldots, d\mathbf{e}_n 
\end{bmatrix}^T
\]

(25)

\[
\mathbf{a}_{bd} = \begin{bmatrix}
d\mathbf{a}_{11}, d\mathbf{a}_{12}, \ldots, d\mathbf{a}_{1m} 
\end{bmatrix}^T
\]

(26)

\[
\mathbf{a} = \begin{bmatrix}
d\mathbf{a}_{11}, \ldots, d\mathbf{a}_{m1}, \ldots, d\mathbf{a}_{mn} 
\end{bmatrix}^T
\]

(27)

\(\mathbf{g}\) is the gravity-acceleration vector. Moreover, \( \mathbf{a}_{bd} \) is the dissipative torque/force acting at the \( i \)th joint, and \( \mathbf{a}_i \) is the driving torque/force acting at the \( i \)th joint.

Kane's dynamical equations thus become

\[
\mathbf{a}_{bg} + \mathbf{a}_{bd} + \mathbf{a} + \dot{\mathbf{a}} = 0.
\]

(28)

From the above equations, we derive

\[
\mathbf{a} = -\mathbf{a}_{bg} - \mathbf{a}_{bd} - \dot{\mathbf{a}} = -\mathbf{a}_{bg} + \sum_{j=1}^n \left( \frac{\partial \mathbf{a}_{ij}}{\partial \mathbf{q}} \right)^T \mathbf{h}_j + \left( \frac{\partial \mathbf{a}_i}{\partial \mathbf{q}} \right)^T \mathbf{m}_j (\dot{\mathbf{q}} - \mathbf{g}).
\]

(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.

\[
\mathbf{q}_i = \begin{cases} 
\theta_i, & \text{if the } i\text{th joint is rotational}, \\
b_i, & \text{if the } i\text{th joint is prismatic}.
\end{cases}
\]

(30)

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

\[
\frac{\partial \mathbf{a}_i}{\partial \mathbf{q}} = \mathbf{A}_j, \quad \frac{\partial \mathbf{a}_i}{\partial \mathbf{q}} = \mathbf{B}_j, \quad \text{for } j = 1, 2, \ldots, n,
\]

(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

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

(32)

\[
\mathbf{B}_j = [\mathbf{e}_1 \times \mathbf{s}_1, \mathbf{e}_2 \times \mathbf{s}_2, \ldots, \mathbf{e}_j \times \mathbf{s}_j, 0, \ldots, 0];
\]

(33)

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

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

(34)

\[
\mathbf{B}_j = [\mathbf{e}_1 \times \mathbf{s}_1, \ldots, \mathbf{e}_{i-1} \times \mathbf{s}_{i-1}, \mathbf{e}_i, \mathbf{e}_{i-1} \times \mathbf{s}_{i-1}, \ldots, \mathbf{e}_j \times \mathbf{s}_j, 0, \ldots, 0],
\]

(35)
where \( s_{ij} \) (for \( i = 1, 2, \ldots, j \)) is defined as the vector from the \( i \)th origin, \( O_i \), to the mass center of the \( j \)th link, \( C_j \), namely,

\[
s_{ij} = \begin{cases} \mathbf{a}_{i+1} + \cdots + \mathbf{a}_j + \mathbf{p}_j, & \text{if } i < j, \\ \mathbf{p}_i, & \text{if } i = j. \end{cases}
\]  

(36)

Thus, the generalized driving torque/force can be expressed in terms of \( A_j \) and \( B_j \) as follows:

\[
\tau = -\phi_d + \sum_{j=1}^{n} [A_j^T \dot{\mathbf{h}}_j + B_j^T m_j (\ddot{\mathbf{e}}_j - \mathbf{g})].
\]  

(37)

in which items \( A_j^T \dot{\mathbf{h}}_j \) and \( B_j^T m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \) are given next. If all joints are rotational, then

\[
A_j^T \dot{\mathbf{h}}_j = \begin{bmatrix} \mathbf{e}_1 \cdot \dot{\mathbf{h}}_j \\ \mathbf{e}_2 \cdot \dot{\mathbf{h}}_j \\ \vdots \\ \mathbf{e}_j \cdot \dot{\mathbf{h}}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\]

(38)

\[
B_j^T m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) = \begin{bmatrix} \mathbf{e}_1 \times \mathbf{s}_{j1} \cdot m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \\ \mathbf{e}_2 \times \mathbf{s}_{j2} \cdot m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \\ \vdots \\ \mathbf{e}_j \times \mathbf{s}_{j1} \cdot m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}
\]

(39)

If the \( i \)th joint is prismatic, for \( 1 \leq i \leq n \), then the \( i \)th entry of \( A_j^T \dot{\mathbf{h}}_j \) changes to 0, whereas that of \( B_j^T m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \) changes to \( \mathbf{e}_j \cdot m_j (\ddot{\mathbf{e}}_j - \mathbf{g}) \). All the other entries remain unchanged.

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

\[
\phi_d = \phi_d(\mathbf{q}, \dot{\mathbf{q}}),
\]

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 \( \phi_d \) equal to zero. The problem at hand is, thus, to derive efficient formulae to compute each \( \tau_i \). First, we define

\[
\mathbf{\dot{\bar{p}}}_i = \ddot{\mathbf{e}}_i - \mathbf{g}.
\]

(41)
From the dynamical equation (37), by ignoring the dissipative-force term, we derive the following formulae:

(i) if the $i$th joint is rotational,

$$\tau_i = \sum_{j=i}^{n} (e_i \cdot \mathbf{h}_j + e_i \times s_{ij} \times m_j \mathbf{p}_j) = e_i \cdot \sum_{j=i}^{n} (\mathbf{h}_j + s_{ij} \times m_j \mathbf{p}_j);$$

(ii) if the $i$th joint is prismatic,

$$\tau_i = \sum_{j=i}^{n} e_i \cdot m_j \mathbf{p}_j = e_i \cdot \sum_{j=i}^{n} m_j \mathbf{p}_j.$$  

Since the dot product of two vectors is frame invariant, $\tau_i$ can be computed in the $i$th coordinate frame as follows:

(i) if the $i$th joint is rotational,

$$\tau_i = \sum_{j=i}^{n} e_i \cdot [\mathbf{h}_j + s_{ij} \times m_j \mathbf{p}_j];$$

(ii) if the $i$th joint is prismatic,

$$\tau_i = [e_i]_i \sum_{j=i}^{n} m_j [\mathbf{p}_j]_i,$$

where vector $[e_i]_i$ has the simple form given in equation (9).

Notice that, if we compute $\tau_i$ backwards, i.e. successively from $\tau_n$ to $\tau_1$, then the term $\sum_{j=i}^{n} [\mathbf{h}_j + s_{ij} \times m_j \mathbf{p}_j]$ of equation (44), and the term $\sum m_j [\mathbf{p}_j]_i$ of equation (45), can be computed from the $\sum [\mathbf{h}_j + s_{ij+1} \times m_j \mathbf{p}_j]_{i+1}$ and $\sum m_j [\mathbf{p}_j]_{i+1}$ terms, respectively [15]. These have already been computed while computing $\tau_{i+1}$, namely,

$$\sum_{j=i}^{n} [\mathbf{h}_j + s_{ij} \times m_j \mathbf{p}_j] = \sum_{j=i}^{n} [\mathbf{h}_j + (s_{ij+1} + a_{ij+1}) \times m_j \mathbf{p}_j] + [\mathbf{h}_i + s_{ij} \times m_j \mathbf{p}_j],$$

$$= \sum_{j=i}^{n} [\mathbf{h}_j + s_{ij+1} \times m_j \mathbf{p}_j]_{i+1} + [a_{ij+1}]_i \times \sum_{j=i+1}^{n} m_j [\mathbf{p}_j]_{i+1} + [\mathbf{h}_i + s_{ij} \times m_j \mathbf{p}_j] + [k_i],$$

for $i = n, n-1, \ldots, 1$, where $k_i$ is defined as

$$k_i = [\mathbf{h}_i]_i + [s_{ij}]_i \times m_j [\mathbf{p}_j]_i + [a_{ij+1}]_i \times \sum_{j=i+1}^{n} m_j [\mathbf{p}_j]_{i+1},$$

and

$$[s_{ij}]_i = [\rho_i]_{i+1}.$$  

Moreover,

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

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

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

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 $\ddot{\mathbf{c}}_i$, that follows:

$$\ddot{\mathbf{p}}_i = f_i(q, \dot{q}, \ddot{q}) + (\ddot{\mathbf{c}}_0 - \ddot{\mathbf{g}}).$$
Equation (51) means that \( \ddot{p}_i \) is nothing but \( \ddot{e}_i \) when the base frame undergoes an additional acceleration of \( -g \). Hence, \( \ddot{p}_i \) can be computed using the same formulae which are used for the computation of \( \ddot{e}_i \), just by setting \( \ddot{p}_0 = -g \), 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 \( k \)th stage;
- \( A_k \) — number of additions required in the \( k \)th 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.**

\[
\text{for } i \leftarrow 1 \text{ until } n \text{ step 1 do }
\]

\[
\text{if } P \text{ then }
\]

\[
[a_i]_{i-1} \leftarrow \begin{pmatrix} a_i \\ -b_i \sin \alpha_i \\ b_i \cos \alpha_i \end{pmatrix}
\]

\[
\text{enddo}
\]

If the \( i \)th joint is rotational, \( \ddot{a}_i \) 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 \( Q_i \) are not computed in this stage. Instead, the product of \( Q_i \) 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 \( Q_i \).

Thus, for this stage, one has

\[
M_1 = 2p, \quad A_1 = 0 \quad (52)
\]

**Stage 2. Kinematical computations.**

In this stage, the following \( 3 \times 3 \) tensors are introduced:

\[
\Omega_i = \frac{\partial \omega_i \times v}{\partial v}, \quad \Omega_i = \frac{\partial \dot{\omega}_i \times v}{\partial v}, \quad \mathbf{W}_i = \Omega_i + \Omega_i^2
\]

where \( v \) is an arbitrary three-dimensional vector. It is clear that both \( \Omega_i \) and \( \Omega_i \) are skew symmetric matrices. Now,

\[
[a_i]_1 \leftarrow [0, 0, 1]^T \quad \ldots 0m + 0a
\]

\[
[\omega_i]_1 \leftarrow \dot{\theta}_i [a_i]_1 \quad \ldots 0m + 0a
\]

\[
[\dot{\omega}_i]_1 \leftarrow \ddot{\theta}_i [a_i]_1 \quad \ldots 0m + 0a
\]

\[
[\mathbf{W}_i]_1 \leftarrow \Omega_i + \Omega_i^2 \quad \ldots 1m + 0a
\]

\[
[\dot{p}_i]_1 \leftarrow -Q_i^T [g]_0 + [\mathbf{W}_i, \rho_i]_1 \quad \ldots 12m + 9a
\]
for $i \leftarrow 2$ until $n$ step 1 do

$$[e_i] \leftarrow [0, 0, 1]^T$$

$$[\omega_i] \leftarrow Q_i^T[\omega_{i-1}], + \Omega_i, [e_i]$$

$$[\omega_i] \leftarrow Q_i^T[\omega_{i-1}], + \Omega_i, [\hat{\theta}_i, e_i]$$

$$[\tilde{W}_i] \leftarrow \Omega_i + \Omega_i^2$$

$$[\tilde{p}_i] \leftarrow Q_i^T[\tilde{p}_{i-1} - W_{i-1}(a_i - \rho_{i-1})] - [\tilde{W}_i \rho_i]$$

if $P$ then

$$[\tilde{p}_i] \leftarrow [\tilde{p}_i] - [2\Omega_i, \hat{\theta}_i, e_i]$$

endif

enddo

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

$$Q_i v = \begin{pmatrix}
\cos \theta_i & -\sin \theta_i & 0 \\
\cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \\
\sin \alpha_i \sin \theta_i & \sin \alpha_i \cos \theta_i & \cos \alpha_i
\end{pmatrix}
\begin{pmatrix}
v_1 \\
v_2 \\
v_3
\end{pmatrix}
= \begin{pmatrix}
v_1 \cos \theta_i - v_2 \sin \theta_i \\
u \cos \alpha_i - v_1 \sin \alpha_i \\
u \sin \alpha_i + v_3 \cos \alpha_i
\end{pmatrix},$$

(54)

where $u$ is an intermediate variable defined as

$$u = v_1 \sin \theta_i + v_2 \cos \theta_i.$$ 

(55)

From equations (54) and (55), it is apparent that computing $Q_i v$ requires only eight multiplications and four additions. Similarly, computing $Q_i^T v$ requires the same numbers of operations. Moreover $[e_i]$ has the simple form of $[0, 0, 1]^T$, and hence, any vector times $[e_i]$ requires no multiplications and zero additions.

Thus, for this stage, one has

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

(56)

Stage 3. Computation of the generalized driving force $\tau$.

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

$$u_i = m_i \tilde{p}_i,$$

$$v_i = \sum_{j=1}^{n} m_j \tilde{p}_j,$$

$$w_i = \sum_{j=1}^{n} [\hat{h}_j + s_{ji} \times m_j \tilde{p}_j].$$

for $i \leftarrow n$ until $1$ step $-1$ do

$$[\hat{h}_i] \leftarrow \{1\} + \{\omega_i\} + \{\Omega_i\} + \{1\} + \{\omega_i\},$$

$$u_i \leftarrow m_i \tilde{p}_i,$$

$$v_{i+1} \leftarrow Q_{i+1} v_{i+1},$$

$$k_i \leftarrow \{\hat{h}_i\} + \{\rho_i\} \times u_i + \{a_{i+1}\} \times v_{i+1},$$

$$w_i \leftarrow Q_{i+1} w_{i+1} + k_i,$$

$$v_{i+1} \leftarrow v_{i+1} + u_i,$$

$$\ldots 0m + 0a$$

$$\ldots 24m + 18a$$

$$\ldots 3m + 0a$$

$$\ldots 8m + 4a$$

$$\ldots 12m + 12a$$

$$\ldots 8m + 7a$$

$$\ldots 0m + 3a$$
An algorithm for the solution of inverse dynamics

if \( R \) then
\[ \tau_i \leftarrow [\theta_i] \cdot w_i, \]
\[ \ldots \quad 0m + 0a \]
else
\[ \tau_i \leftarrow [\theta_i] \cdot v_i, \]
\[ \ldots \quad 0m + 0a \]
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 \( Q_{n+1}w_{n+1} \), \( Q_{n+1}w_{n+1}w_{n+1} \), and \([\omega_{n+1}] \times v_{n+1}\), since \( v_{n+1} = w_{n+1} = 0 \). Moreover, only eight multiplications and two additions are required for computing \([h_i]_i\), since \([\omega_i]_i = [0, 0, \theta_i]^T\) and \([\omega_i]_i = [0, 0, \theta_i]^T\). Furthermore, \( \tau_i \) is equal to the third component of either \( w_i \) or \( v_i \). Hence, when \( i = n \), it is unnecessary to compute all components of \( w_i \) and \( v_i \), 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:

\[ M_3 = 55n - 62, \quad A_3 = 44n - 58. \]  \hfill (57)

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

\[ M = \sum_{k=1}^{3} M_k = 105n - 109 \]  \hfill (58)

\[ A = \sum_{k=1}^{3} A_k = 90n - 105, \]  \hfill (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>
<thead>
<tr>
<th>Author(s)</th>
<th>Method</th>
<th>Multiplications</th>
<th>Additions</th>
<th>Multiplications (n = 6)</th>
<th>Additions (n = 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hollerbach [3]</td>
<td>EL</td>
<td>412n - 277</td>
<td>320n - 201</td>
<td>2195</td>
<td>1719</td>
</tr>
<tr>
<td>Luh et al. [2]</td>
<td>NE</td>
<td>150n - 48</td>
<td>131n - 48</td>
<td>852</td>
<td>738</td>
</tr>
<tr>
<td>Walker and Orin [20]</td>
<td>NE</td>
<td>137n - 22</td>
<td>101n - 11</td>
<td>800</td>
<td>595</td>
</tr>
<tr>
<td>Hollerbach [21]</td>
<td>NE</td>
<td>(not available)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kane and Levinson [13]</td>
<td>Kane</td>
<td>(not available)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Khalil et al. [10]</td>
<td>NE</td>
<td>105n - 92</td>
<td>94n - 86</td>
<td>538</td>
<td>478</td>
</tr>
<tr>
<td>This paper's</td>
<td>Kane</td>
<td>105n - 109</td>
<td>90n - 105</td>
<td>521</td>
<td>435</td>
</tr>
</tbody>
</table>
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:

\[
m_1 = 9.0, \quad \rho_1 = \begin{pmatrix} 0 \\ 0 \\ -0.1 \end{pmatrix}, \quad l_1 = \begin{pmatrix} 0.02 & 0 & 0 \\ 0.01 & 0 \end{pmatrix};
\]

\[
m_2 = 6.0, \quad \rho_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad l_2 = \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0 & 0.06 \end{pmatrix};
\]

\[
m_3 = 4.0, \quad \rho_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad l_3 = \begin{pmatrix} 0.4 & 0 \end{pmatrix};
\]

\[
m_4 = 1.0, \quad \rho_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad l_4 = \begin{pmatrix} 0.001 & 0 & 0 \\ 0 & 0.0005 & 0 \end{pmatrix};
\]

<table>
<thead>
<tr>
<th>Joint</th>
<th>( \theta_i ) (deg)</th>
<th>( x_i ) (m)</th>
<th>( b_i ) (m)</th>
<th>Initial ( \theta_i ) (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-0.1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-90</td>
<td>0</td>
<td>-0.1</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>-0.6</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
An algorithm for the solution of inverse dynamics

\[
m_5 = 0.6, \quad [\rho_5]_5 = \begin{pmatrix} 0 \\ 0.06 \\ 0 \end{pmatrix}, \quad [I_5]_5 = \begin{pmatrix} 0.0005 & 0 & 0 \\ 0 & 0.0002 & 0 \\ 0 & 0 & 0.0005 \end{pmatrix};
\]

\[
m_6 = 0.5, \quad [\rho_6]_6 = \begin{pmatrix} 0 \\ 0.2 \end{pmatrix}, \quad [I_6]_6 = \begin{pmatrix} 0.003 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.002 \end{pmatrix},
\]

where the units of \(m_i\), \(\rho_i\), and \(I_i\) are \((N \cdot s^2/m)\), \((m)\), and \((N \cdot s^2 \cdot m)\), respectively. The gravity acceleration used is

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

The joint trajectories of interest are defined as follows:

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

where

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

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

\[
B = 2\pi/10,
\]

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:

\[
m_1 = 680, \quad [\rho_1]_1 = \begin{pmatrix} 0 \\ -0.33 \\ 0 \end{pmatrix}, \quad [I_1]_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 62 & 0 \\ 0 & 0 & 0 \end{pmatrix},
\]

\[
m_2 = 360, \quad [\rho_2]_2 = \begin{pmatrix} -0.87 \\ 0 \\ -0.13 \end{pmatrix}, \quad [I_2]_2 = \begin{pmatrix} 11 & 0 & 0 \\ 0 & 53 & 0 \\ 0 & 0 & 44 \end{pmatrix},
\]

\[
m_3 = 180, \quad [\rho_3]_3 = \begin{pmatrix} -0.64 \\ 0.04 \\ 0 \end{pmatrix}, \quad [I_3]_3 = \begin{pmatrix} 1.1 & 0 & 0 \\ 0 & 44 & 0 \\ 0 & 0 & 44 \end{pmatrix}.
\]
Fig. 4. Joint trajectories of Example 1.

Fig. 5. Driving torques and forces of Example 1.
Table 3. HD parameters of the manipulator

<table>
<thead>
<tr>
<th>Joint</th>
<th>( x_i ) (deg)</th>
<th>( x_i ) (m)</th>
<th>( b_i ) (m)</th>
<th>Initial ( \theta_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>68.5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1.02</td>
<td>0</td>
<td>-135</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1.02</td>
<td>0</td>
<td>39.6</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0.2</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
<td>0</td>
<td>-0.41</td>
<td>90</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
m_4 &= 55, & \quad \begin{bmatrix} \rho_4 \\ \eta_4 \end{bmatrix} &= \begin{pmatrix} -0.12 \\ 0.04 \end{pmatrix}, & \quad \begin{bmatrix} \kappa \end{bmatrix} &= \begin{pmatrix} 0.44 \\ 0.91 \end{pmatrix} \\

m_5 &= 36, & \quad \begin{bmatrix} \rho_5 \\ \eta_5 \end{bmatrix} &= \begin{pmatrix} -0.05 \\ -0.08 \end{pmatrix}, & \quad \begin{bmatrix} \kappa \end{bmatrix} &= \begin{pmatrix} 0.47 \\ 0.38 \end{pmatrix} \\

m_6 &= 68, & \quad \begin{bmatrix} \rho_6 \\ \eta_6 \end{bmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, & \quad \begin{bmatrix} \kappa \end{bmatrix} &= \begin{pmatrix} 0.44 \\ 0.64 \end{pmatrix}
\end{align*}
\]

where the units of \( m_i \), \( \rho_i \), and \( \kappa \) are \( (N \cdot s^2/m) \), (m) and \( (N \cdot s^2 \cdot m) \), respectively.

The gravity acceleration used is

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

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. 7. Joint trajectories of Example 2. Fig. 8. Driving torques of Example 2.
nonworking constraint forces, which leads to a systematic 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 (Fonds pour la Formation de Chercheurs et l’Aide à 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