[8] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control system: A new type servo system for periodic exogenous signals," *IEEE Trans. Automat. Contr.*, vol. 33, no. 7, pp. 659–667, 1988.

[9] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulators," *ASME J. Dyn. Sys. Meas. Contr.*, vol. 33, no. 7, pp. 659–668, June 1986.

[10] C. G. Kang, W.-W. Kao, M. Boals, and R. Horowitz, "Modeling, identification and simulation of a two link SCARA manipulator," in *Proc. ASME Winter Ann. Meeting* (Chicago), Nov. 1988.

[11] S. Kawamura, F. Miyazaki, and S. Arimoto, "Realization of robot motion based on a learning method," *IEEE Trans. Syst. Man Machine Cybern.*, Jan. 1988.

[12] D. Koditschek, "Adaptive techniques for mechanical manipulators," in *Proc. 5th Yale Workshop Applicat. Adaptive Syst. Theory* (Yale University), May 1987.

[13] N. Sadegh, "Adaptive control of mechanical manipulators: Stability and robustness analysis," Ph.D. dissertation, Dept. Mechanical Eng., Univ. of California at Berkeley, 1987.

[14] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators," *Int. J. Robotics Res.*, vol. 9, no. 3, pp. 74–92, June 1990.

[15] N. Sadegh, R. Horowitz, W.-W. Kao, and M. Tomizuka, "A unified approach to the design of adaptive and repetitive controllers for robotic manipulators," in *Proc. USA–Japan Symp. Flexible Automat.* (Minneapolis, MN), July 1988.

[16] N. Sadegh and K. Guglielmo, "A new repetitive controller for mechanical manipulators," *J. Robotic Syst.*, vol. 8, Aug. 1991.

[17] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," in *Robots: Theory and Applicat., Proc. ASME Winter Ann. Meeting*, Dec. 1986.

[18] M. Togai and O. Yamano, "Learning control and its optimality: Analysis and its application to industrial robots," in *Proc. IEEE Conf. Robotics Automat.* (San Francisco), 1986.

[19] M. Yamamoto and S. Hara, "Relationships between internal and external stability for infinite-dimensional systems with applications to a servo problem," *IEEE Trans. Automat. Contr.*, vol. 33, no. 11, pp. 1044–1052, 1988.

Formulation and Efficient Computation of Inverse Dynamics of Space Robots

Ranjan Mukherjee and Yoshihiko Nakamura

Abstract—A free-flying space robot is expected to perform various tasks for the construction and maintenance of space structures. Such a robotic system has kinematic and dynamic features that differ from those fixed on the earth mainly because of the momentum constraints that govern its motion. This paper presents the solution to the inverse dynamics problem of a space robotic system, in the presence of external generalized forces. While solving for the inverse dynamics, the computations for the inverse kinematics are considered simultaneously, and both computations are developed on the basis of momentum constraints. An efficient computational scheme for the inverse dynamics problem is then established. We also discuss an intrinsic feature of space robotic systems that can be utilized to reduce the computational time by parallel recursion. We conclude the paper with a discussion on the importance of the role of momentum constraints in the solution of inverse dynamics.

Manuscript received April 20, 1990; revised November 18, 1991. Portions of this paper were presented at the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), November 18–20, 1990, Kobe, Japan.

Y. Nakamura is with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan.

R. Mukherjee is with the Department of Mechanical Engineering, Naval Postgraduate School, Monterey, CA 93943.

IEEE Log Number 9107019.

Nomenclature

frames I, C	The inertial frame and the center-of-mass frame, respectively.
frames V, B	The vehicle frame and the manipulator base frame respectively.
frame E	The manipulator end-effector frame.
frame K	The kth body frame. The kth link frame of manipulator for $k = 1, \cdots, n$. The nth link frame is identical to the manipulator end-effector frame. The vehicle frame for $k = 0$.
n	Degrees of freedom of the manipulator.
${}^I z_k \in R^3$	Unit vector in the direction of the z axis of the kth link coordinates and represented in the inertial frame (m).
m_k	Mass of the kth body (kg). The zeroth body is the vehicle. The kth body ($k \geq 1$) is the kth link of the manipulator.
$\boldsymbol{s}_k \in R^3$	Position vector of the center of mass of the kth link from the origin of the kth link coordinates (m).
$\boldsymbol{l}_k \in R^3$	Position vector from the origin of the $(k-1)$th frame to the origin of the kth frame and measured in the kth frame (m).
${}^I \boldsymbol{r}_k \in R^3$	Position vector from the origin of the inertial frame to the center of mass of the kth body represented in the inertial frame (m).
${}^I \bar{\boldsymbol{r}}_k \in R^3$	Position vector from the origin of the inertial frame to the origin of the kth link frame represented in the inertial frame (m).
${}^C \boldsymbol{r}_k \in R^3$	Position vector from the origin of the center-of-mass frame to the center of mass of the kth body represented in the center-of-mass frame (m).
${}^I \omega_k \in R^3$	Angular velocity of the kth body in the inertial frame (rad/s).
${}^k \boldsymbol{I}_k \in R^{3\times 3}$	Inertia matrix of the kth body about its center of mass in the kth body frame; a constant matrix (kg^2).
${}^I \boldsymbol{I}_k \in R^{3\times 3}$	Inertia matrix of the kth body about its center of mass in the inertial frame (kg^2).
$\dot{\boldsymbol{\theta}}_1 \in R^6$	Linear velocity of the center of mass and angular velocity of the vehicle in the inertial frame (m/s, rad/s).
$\boldsymbol{\theta}_2 \in R^n$	Joint variables $(q_1, \cdots, q_n)$ of the manipulator (rad).
${}^I \boldsymbol{A}_k \in R^{3\times 3}$	Rotation matrix from the inertial frame to the kth body frame (the vehicle frame for $k = 0$, the kth link frame of the manipulator for $k = 1, \cdots, n$).
$\boldsymbol{J}_2^k \in R^{6\times n}$	Jacobian matrix of the position and orientation of the center of mass of kth body ($k = 1, \cdots, n$) in the inertial frame (m).
${}^I \boldsymbol{F}_k^* \in R^3$	Inertial force of the kth link (N).
${}^I \boldsymbol{N}_k^* \in R^3$	Inertial moment of the kth link (N·m).
${}^I \boldsymbol{F}_k \in R^3$	Total force acting on the kth link from the $(k+1)$th link at the origin of the kth link frame (N).
${}^I \boldsymbol{N}_k \in R^3$	Total moment acting on the kth link from the $(k+1)$th link at the origin of the kth link frame. (N·m).
τ_k	Torque at the kth revolute joint (N·m).
$\boldsymbol{E}_i \in R^{i\times i}$	$i \times i$ identity matrix.

I. Introduction

There have been significant technological developments in space over the last decade, and the necessity of deploying robotic systems to perform various tasks in space has been realized over these years. These robotic systems, consisting of two or more manipulators mounted on a vehicle, are expected to make future space missions safe and cost effective. The space environment is significantly different from that on earth. The space environment imposes momentum constraints on the system for which the kinematics and dynamics of space robots have intrinsic features.

Alexander and Cannon [1] discussed the computation of joint torque for manipulator endpoint control assuming that the thrust force on the vehicle was known. Vafa and Dubowsky [12], [13] proposed the concept of a virtual manipulator to simplify the kinematics and dynamics of a space robotic system. Umetani and Yoshida [11] put forth a scheme to continuously control the end-effector by directly controlling only the joints of the manipulator. Longman *et al.* [5] discussed the coupling of manipulator motion and vehicle motion. Miyazaki *et al.* [8] discussed a sensor feedback scheme using the transposed generalized Jacobian matrix. Nakamura and Mukherjee [9], [10] proposed a Lyapunov control scheme to control both the vehicle orientation and the joint angles of the manipulator by directly controlling only the joints of the manipulator. The nonholonomic structure of the system was used to plan the motion without the use of reaction jets or reaction wheels.

The kinematics of a space robotic system in the absence of external forces have been studied by various researchers [9]–[11]. They incorporated the principle of momentum conservation in the kinematic equations. The conservation of momentum is a principle of dynamics, and therefore the kinematic equations are representative of the dynamical behavior of the system, only in the absence of external forces.

In real applications where the robot will come in physical contact with external objects, force control will be an important issue. In such cases, the momentum of the space robot does not remain conserved. Therefore, the above mentioned framework based on momentum conservation cannot deal with these practical problems of great importance. Yamada [14] developed an efficient scheme for the computation of the inverse dynamics for space robots. His method is based on momentum conservation and therefore assumes the absence of external forces. His method is also incapable of computing the linear and angular velocity of the vehicle. It assumes that these variables are to be obtained from sensor readings.

In this paper, the inverse dynamics of a space robotic system in the presence of external generalized forces has been analyzed. The background of space robot kinematics is reviewed in Section II from Nakamura and Mukherjee [10]. In Section III, we develop the computational steps required for inverse dynamics along with inverse kinematics. In Section IV, we present the efficient algorithm for computation, determine the computational burden, and carry out a comparison with the algorithm proposed by Yamada [14]. In spite of the many advantages that our approach has over that of [14], Yamada's approach has only 9.16% fewer multiplications and 4.36% fewer additions. In Section IV, we also explore the possibility of a further reduction in the computational time by using parallel recursion. We conclude this paper with a discussion on the importance of the role of the momentum constraints in the solution of the inverse dynamics.

II. Background

In this section, the kinematic equations from [10] are summarized. Fig. 1 shows a model of a space robot system with six reference

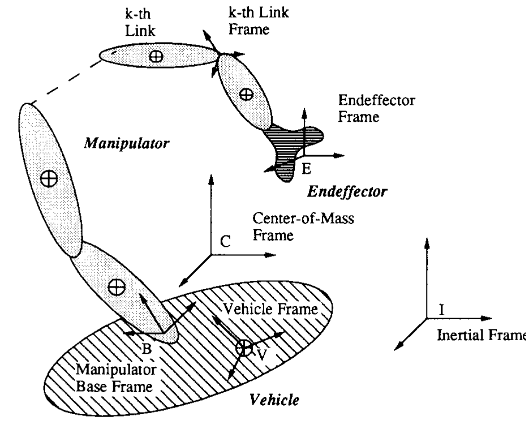

Fig. 1. Six coordinate frames for the space vehicle/manipulator system.

frames denoted by I, C, V, B, K, and E. The link frames of the manipulator are defined by the Denavit–Hartenberg convention [2]. The center-of-mass frame, which is fixed at the center of mass of the whole system, translates with respect to the inertial frame when the linear momentum of the system is nonzero. The orientation of the center-of-mass frame is always maintained parallel to the inertial frame. The vehicle frame is assumed to be fixed at the center of mass of the vehicle. The inertial frame and the center-of-mass frame coincide at the initial time. This simplifies the discussion and does not cause any loss of generality.

In this section, we assume that there are no external forces acting on the total system. The system is controlled only by manipulator joints that generate internal forces. This assumption will be removed in the next section and we will discuss the general case where there exist external forces that may include thruster forces of the vehicle. If the system is assumed to have zero linear and angular momentum at the initial time, the linear and angular momentum conservations are represented by

$$\sum_{k=0}^{n} m_k\, {}^I\dot{\boldsymbol{r}}_k = 0 \tag{1}$$

$$\sum_{k=0}^{n} \left({}^I\boldsymbol{I}_k\, {}^I\omega_k + m_k\, {}^I\boldsymbol{r}_k \times {}^I\dot{\boldsymbol{r}}_k \right) = 0 \tag{2}$$

where ${}^I\dot{\boldsymbol{r}}_0$ and ${}^I\omega_0$ are the linear and angular velocities of the vehicle center of mass, respectively. ${}^I\dot{\boldsymbol{r}}_k$ is computed as

$${}^I\dot{\boldsymbol{r}}_k = [\boldsymbol{E}_3 \quad -{}^I\boldsymbol{R}_{0k}]\dot{\boldsymbol{\theta}}_1 + [\boldsymbol{E}_3 \quad 0]\,\boldsymbol{J}_2^k\,\dot{\boldsymbol{\theta}}_2 \tag{3}$$

where ${}^I\boldsymbol{R}_{0k}$, ${}^I\boldsymbol{r}_{0k}$, and $\boldsymbol{J}_2^k$ are defined by

$${}^I\boldsymbol{R}_{0k} \triangleq \begin{bmatrix} 0 & -{}^I r_{0k\,z} & {}^I r_{0k\,y} \\ {}^I r_{0k\,z} & 0 & -{}^I r_{0k\,x} \\ -{}^I r_{0k\,y} & {}^I r_{0k\,x} & 0 \end{bmatrix}$$

$${}^I\boldsymbol{r}_{0k} \triangleq ({}^I\boldsymbol{r}_k - {}^I\boldsymbol{r}_0) \triangleq \begin{bmatrix} {}^I r_{0k\,x} \\ {}^I r_{0k\,y} \\ {}^I r_{0k\,z} \end{bmatrix} \tag{4}$$

$$\boldsymbol{J}_2^k \triangleq \begin{bmatrix} {}^I\boldsymbol{\beta}_1^k & {}^I\boldsymbol{\beta}_2^k & \cdots & {}^I\boldsymbol{\beta}_n^k \\ {}^I\boldsymbol{\gamma}_1 & {}^I\boldsymbol{\gamma}_2 & \cdots & {}^I\boldsymbol{\gamma}_n \end{bmatrix}. \tag{5}$$

${}^I\boldsymbol{\beta}_j^k \in R^3$ and ${}^I\boldsymbol{\gamma}_j \in R^3$ are defined as follows:

$${}^I\boldsymbol{\gamma}_j \triangleq \begin{cases} {}^I\boldsymbol{z}_{j-1} = {}^I\boldsymbol{A}_{j-1}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \text{for } j = 1, \cdots, k \\ 0 & \text{for } j = k+1, \cdots, n \end{cases} \tag{6}$$

$$ {}^I\beta_j^k \triangleq \begin{cases} {}^I\gamma_j \times ({}^I\boldsymbol{r}_k - {}^I\bar{\boldsymbol{r}}_{j-1}) & \text{for } j = 1, \cdots, k \\ 0 & \text{for } j = k+1, \cdots, n. \end{cases} \tag{7} $$

The following relations hold for ${}^I\boldsymbol{I}_k$ and ${}^I\omega_k$:

$$ {}^I\boldsymbol{I}_k = {}^I\boldsymbol{A}_k\, {}^k\boldsymbol{I}_k\, {}^I\boldsymbol{A}_k^T \tag{8} $$

$$ {}^I\omega_k = \begin{cases} [0 \quad \boldsymbol{E}_3]\dot{\boldsymbol{\theta}}_1 & \text{for } k = 0 \\ {}^I\omega_0 + \sum_{j=1}^k {}^I z_{j-1}\, \dot{q}_j & \text{for } k = 1, \cdots, n. \end{cases} \tag{9} $$

By substituting (3) and (8) into (1) and (2) and summarizing them in a matrix form, the momentum conservation laws are represented by the following single equation:

$$ \boldsymbol{H}_1\dot{\boldsymbol{\theta}}_1 + \boldsymbol{H}_2\dot{\boldsymbol{\theta}}_2 = 0 \tag{10} $$

where

$$ \boldsymbol{H}_1 \triangleq \begin{bmatrix} M\boldsymbol{E}_3 & M\,{}^I\boldsymbol{R}_{C0} \\ M\,{}^I\boldsymbol{R}_C & \sum_{k=0}^n {}^I\boldsymbol{A}_k\, {}^k\boldsymbol{I}_k\, {}^I\boldsymbol{A}_k^T - \sum_{k=0}^n m_k \left({}^I\boldsymbol{R}_k\right)^2 + M\,{}^I\boldsymbol{R}_C\, {}^I\boldsymbol{R}_0 \end{bmatrix} \tag{11} $$

$$ \boldsymbol{H}_2 \triangleq \begin{bmatrix} \sum_{k=1}^n m_k\,[\boldsymbol{E}_3 \quad 0]\,\boldsymbol{J}_2^k \\ \sum_{k=1}^n m_k\, {}^I\boldsymbol{R}_k\,[\boldsymbol{E}_3 \quad 0]\,\boldsymbol{J}_2^k + \boldsymbol{P} \end{bmatrix} \tag{12} $$

and where

$$ M \triangleq \sum_{k=0}^n m_k $$

$$ {}^I\boldsymbol{R}_k \triangleq \begin{bmatrix} 0 & -{}^I r_{kz} & {}^I r_{ky} \\ {}^I r_{kz} & 0 & -{}^I r_{kx} \\ -{}^I r_{ky} & {}^I r_{kx} & 0 \end{bmatrix} $$

$$ {}^I\boldsymbol{r}_k \triangleq \begin{bmatrix} {}^I r_{kx} \\ {}^I r_{ky} \\ {}^I r_{kz} \end{bmatrix} \tag{13} $$

$$ \boldsymbol{P} \triangleq [\boldsymbol{P}_1 \quad \boldsymbol{P}_2 \quad \cdots \quad \boldsymbol{P}_n] \tag{14} $$

$$ \boldsymbol{P}_i \triangleq \left[\sum_{k=i}^n {}^I\boldsymbol{A}_k\, {}^k\boldsymbol{I}_k\, {}^I\boldsymbol{A}_k^T\right] {}^I\boldsymbol{z}_{i-1}. $$

The pure geometrical relationship between the end-effector motion and $\dot{\theta}_1$ and $\dot{\theta}_2$ is described in the following form:

$$ \dot{\boldsymbol{x}}_E = \boldsymbol{J}_1\dot{\boldsymbol{\theta}}_1 + \boldsymbol{J}_2\dot{\boldsymbol{\theta}}_2 \tag{15} $$

where

$$ \dot{\boldsymbol{x}}_E = \begin{bmatrix} {}^I\dot{\boldsymbol{r}}_E \\ {}^I\omega_E \end{bmatrix} = \begin{bmatrix} {}^I\dot{\boldsymbol{r}}_n \\ {}^I\omega_n \end{bmatrix}, \quad \boldsymbol{J}_1 \triangleq \begin{bmatrix} \boldsymbol{E}_3 & -{}^I\boldsymbol{R}_{0n} \\ 0 & \boldsymbol{E}_3 \end{bmatrix}, \quad \boldsymbol{J}_2 \triangleq \boldsymbol{J}_2^n \tag{16} $$

and where $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the geometrical Jacobian matrices that do not take account of the momentum conservation. ${}^I\boldsymbol{R}_{0n}$ in the above equation is defined by (4) for $k = n$.

Equations (10) and (15) are the fundamental equations. In (10), $\boldsymbol{H}_1 \in R^{6\times 6}$ is always nonsingular. This can be explained as follows. For $\dot{\boldsymbol{\theta}}_2 = 0$, the momentum of the system would be given as $\boldsymbol{H}_1\dot{\boldsymbol{\theta}}_1$. Now, for nonzero $\dot{\boldsymbol{\theta}}_1$, it is physically impossible for the momentum to be zero. This physically signifies that the matrix $\boldsymbol{H}_1$ has no null space and is hence invertible. Therefore, (10) is identical to

$$ \dot{\boldsymbol{\theta}}_1 = -\boldsymbol{H}_1^{-1}\boldsymbol{H}_2\dot{\boldsymbol{\theta}}_2 \triangleq \boldsymbol{H}\dot{\boldsymbol{\theta}}_2. \tag{17} $$

Substituting (17) into (15) we have

$$ \dot{\boldsymbol{x}}_E = \left(-\boldsymbol{J}_1\boldsymbol{H}_1^{-1}\boldsymbol{H}_2 + \boldsymbol{J}_2\right)\dot{\boldsymbol{\theta}}_2 = (\boldsymbol{J}_1\boldsymbol{H} + \boldsymbol{J}_2)\dot{\boldsymbol{\theta}}_2 \triangleq \tilde{\boldsymbol{J}}\dot{\boldsymbol{\theta}}_2. \tag{18} $$

Umetani and Yoshida [11] named the coefficient matrix $\tilde{\boldsymbol{J}}$ the *generalized Jacobian matrix*. In the above derivation, the momentum conservation of (10) was used as a constraint to eliminate the dependent variables $\dot{\boldsymbol{\theta}}_1$.

III. Inverse Dynamics of Space Robots

A. Framework

The goal of inverse dynamics is to compute the torques to be applied to the joints of the manipulator such that the system behaves in a desired way. It is possible to compute the joint torques when the desired $\ddot{\boldsymbol{\theta}}_2$, the current values of $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dot{\boldsymbol{\theta}}_1, \dot{\boldsymbol{\theta}}_2$, and the external forces are provided. However, in a practical situation we are often required to control the motion of the end-effector. In such a case, we must also computate the inverse kinematics to obtain $\ddot{\boldsymbol{\theta}}_1$ and $\ddot{\boldsymbol{\theta}}_2$ for a given end-effector motion. The inverse kinematics and the inverse dynamics involve a lot of common computations since the momentum constraints should be considered in both computations. In this section, we discuss the inverse dynamics problem in conjunction with resolved acceleration control [7], where both inverse kinematics and inverse dynamics are to be done simultaneously. We assume that we are provided with the trajectory of $\ddot{\boldsymbol{x}}$ and we know all the external forces that act upon the system along with their points of application.

The external forces mentioned above will also include thruster forces. Thrusters will be indispensable in the practical case where the space robot will have to interact dynamically with the environment. Thrusters would be required to remove additional momentum from the system such that the system does not drift, spin, and move away from its prescribed trajectory.

The momentum constraint in (10) was derived on the basis of the assumption that the momentum, both linear and angular, of the system is initially zero and no external forces act on the system afterward. In the presence of external forces, the momentum of the system is nonzero, and (10) is then replaced by

$$ \boldsymbol{H}_1\dot{\boldsymbol{\theta}}_1 + \boldsymbol{H}_2\dot{\boldsymbol{\theta}}_2 = \boldsymbol{b} \tag{19} $$

where $\boldsymbol{b} \in R^6$ is the vector of the linear and angular momentum of the whole system with reference to the inertial frame. Equation (19) can be differentiated to obtain the acceleration constraint as follows:

$$ \boldsymbol{H}_1\ddot{\boldsymbol{\theta}}_1 + \boldsymbol{H}_2\ddot{\boldsymbol{\theta}}_2 = \dot{\boldsymbol{b}} - \left(\dot{\boldsymbol{H}}_1\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{H}}_2\dot{\boldsymbol{\theta}}_2\right) \triangleq \boldsymbol{z}_1 \tag{20} $$

where $\dot{\boldsymbol{b}}$ physically implies the vector of the external generalized forces acting at the origin of the inertial frame. External forces and moments acting on different links are first converted into equivalent force-moment pairs acting at the origin of the inertial frame. Subsequently, $\dot{\boldsymbol{b}}$ is obtained as the algebraic sum of the equivalent force-moment pairs. In the absence of external forces, we use $\dot{\boldsymbol{b}} = 0$.

The kinematic equation for a space robot can be obtained from (15) as

$$ \boldsymbol{J}_1\ddot{\boldsymbol{\theta}}_1 + \boldsymbol{J}_2\ddot{\boldsymbol{\theta}}_2 = \ddot{\boldsymbol{x}}_E - \left(\dot{\boldsymbol{J}}_1\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{J}}_2\dot{\boldsymbol{\theta}}_2\right) \triangleq \boldsymbol{z}_2. \tag{21} $$

Equations (20) and (21) can be summarized by the following single equation:

$$ \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{J}_1 & \boldsymbol{J}_2 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\theta}}_1 \\ \ddot{\boldsymbol{\theta}}_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \tag{22} $$

In the above equation, the coefficient matrix is of dimension $12 \times (n+6)$. Hence, if $n > 6$, we have kinematic redundancy. In that case, the solution for $\ddot{\boldsymbol{\theta}}_1$, and $\ddot{\boldsymbol{\theta}}_2$ could be obtained by using the pseudoinverse. Note that this kinematic redundancy can be treated by the generalized Jacobian matrix of (18) as well. However, one of the advantages of using (22) lies in the fact that the redundancy can be utilized to plan the motion of both $\ddot{\boldsymbol{\theta}}_1$ and $\ddot{\boldsymbol{\theta}}_2$. While using the generalized Jacobian matrix, the redundancy can be utilized to plan for the motion of $\ddot{\boldsymbol{\theta}}_2$ alone.

We now assume $n = 6$ to discuss the computational requirements and to compare with the approach of Yamada [14]. Equation (22) can thus be solved as

$$\begin{bmatrix} \ddot{\boldsymbol{\theta}}_1 \\ \ddot{\boldsymbol{\theta}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{J}_1 & \boldsymbol{J}_2 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix}. \tag{23}$$

The joint velocities $\dot{\boldsymbol{\theta}}_2$ are measured with the help of sensors. The linear and angular velocity of the vehicle $\dot{\boldsymbol{\theta}}_1$ can also be measured, or they can be computed when they cannot be measured. In such cases, we use the momentum constraint of (19) for the computation of $\dot{\boldsymbol{\theta}}_1$ as

$$\dot{\boldsymbol{\theta}}_1 = \boldsymbol{H}_1^{-1}\left(\boldsymbol{b} - \boldsymbol{H}_2\,\dot{\boldsymbol{\theta}}_2\right). \tag{24}$$

B. Steps of Computation

This section summarizes the steps of the computation required for inverse dynamics in conjunction with resolved acceleration control. The details of efficient computation necessary to execute these steps is to be explained in Section IV.

1) Measure $\boldsymbol{\theta}_2$ and $\dot{\boldsymbol{\theta}}_2$. If $\dot{\boldsymbol{\theta}}_1$ cannot be measured, compute its value with the help of (24) from the present value of $\dot{\boldsymbol{\theta}}_2$. Integrate $\dot{\boldsymbol{\theta}}_1$ numerically to obtain $\boldsymbol{\theta}_1$.
2) Compute the variables $\ddot{\boldsymbol{\theta}}_1$ and $\ddot{\boldsymbol{\theta}}_2$ using (23).
3) Compute the joint torques using the quantities computed in the above two steps. This can be done by using the well known recursive Newton–Euler algorithm proposed in [6]. First compute the link velocities and accelerations in the inertial frame of reference using equations provided in Appendix A. Next compute the generalized inertial forces of each link using equations provided in Appendix B. As a next step, compute the forces exchanged between the different links. Please refer to Appendix C for the necessary equations. Finally compute the joint torques from the forces exchanged between the links using equations provided in Appendix D.

The main difficulty of resolved acceleration control of a space robot as compared to a terrestrial robot is due to the fact that, in resolving acceleration, one has to consider the momentum constraint of (20) as well as the kinematic constraint of (21). The major contribution of this paper is to 1) provide a systematic method in computing the momentum constraint and 2) to propose an efficient algorithm that computes as a whole all three steps mentioned above.

IV. Computational Details

A. Algorithm

This section presents the computational sequence and the important techniques adopted for efficient computation of the inverse dynamics as discussed in Section III. Before the sequence of computation is presented, a few equations are derived that are to be used in computation. Consider (21) and (20) from which we can obtain the following relations directly:

$$\dot{\boldsymbol{J}}_1\,\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{J}}_2\,\dot{\boldsymbol{\theta}}_2 = \ddot{\boldsymbol{x}}_E\,\big|_{\ddot{\boldsymbol{\theta}}_1=\ddot{\boldsymbol{\theta}}_2=0} = \begin{pmatrix} {}^I\ddot{\boldsymbol{r}}_n \\ {}^I\dot{\omega}_n \end{pmatrix}_{\ddot{\boldsymbol{\theta}}_1=\ddot{\boldsymbol{\theta}}_2=0} \tag{25}$$

$$\dot{\boldsymbol{H}}_1\,\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{H}}_2\,\dot{\boldsymbol{\theta}}_2 = \dot{\boldsymbol{b}}\,\big|_{\ddot{\boldsymbol{\theta}}_1=\ddot{\boldsymbol{\theta}}_2=0}\,. \tag{26}$$

The matrix $\boldsymbol{H}_1$ is always invertible as explained in Section II. The inverse can be computed from (11) using the inverse of the block matrix [4] in the following way:

$$\boldsymbol{H}_1^{-1} = \begin{bmatrix} \boldsymbol{E}_3 & -{}^I\boldsymbol{R}_{C0} \\ 0 & \boldsymbol{E}_3 \end{bmatrix} \begin{bmatrix} 1/M\boldsymbol{E}_3 & 0 \\ 0 & \boldsymbol{S}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_3 & 0 \\ -{}^I\boldsymbol{R}_C & \boldsymbol{E}_3 \end{bmatrix} \tag{27}$$

where $\boldsymbol{S}$ is given as

$$\boldsymbol{S} = \sum_{k=0}^{n}\left[{}^I\boldsymbol{A}_k\,{}^k\boldsymbol{I}_k\,{}^I\boldsymbol{A}_k^T - m_k\left({}^I\boldsymbol{R}_k\right)^2\right] + M\left({}^I\boldsymbol{R}_C\right)^2. \tag{28}$$

Note that the matrix now to be inverted is of dimension 3×3. We compute the coefficient matrix of (23) again using the inverse of the block matrix as follows:

$$\begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{J}_1 & \boldsymbol{J}_2 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{E}_6 & -\boldsymbol{H}_1^{-1}\boldsymbol{H}_2 \\ 0 & \boldsymbol{E}_6 \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1^{-1} & 0 \\ 0 & \tilde{\boldsymbol{J}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_6 & 0 \\ -\boldsymbol{J}_1\boldsymbol{H}_1^{-1} & \boldsymbol{E}_6 \end{bmatrix} \tag{29}$$

where $\tilde{\boldsymbol{J}}$ is defined by (18). It is reasonable to assume that the end-effector path that would be specified would not pass through the singularities of the generalized Jacobian $\tilde{J}$. Therefore, $\tilde{J}^{-1}$ would exist. The sequence with which the inverse dynamics computation should be performed for efficiency is summarized next. For most of the steps below, we provide in brackets the number of multiplications M and additions A that will be necessary.

1) Compute the total mass of the system, $\sum_{k=0}^{n} m_k$. This has to be computed only once and can be done ahead of time. [nA.]
2) Compute ${}^I\omega_k$ for $k = 0, 1, \cdots, n$ using (9). [$3nM + 3nA$.]
3) Compute ${}^I\boldsymbol{A}_k$ for $k = 0, 1, \cdots, n$. This would require the Euler angles of vehicle orientation and $\boldsymbol{\theta}_2$. [$27nM + 18nA$.]
4) Compute ${}^I\boldsymbol{r}_k$ for $k = 0, 1, \cdots, n$. This would require all the necessary homogeneous transformation matrices evaluated in the previous step and the D-H link parameters $\boldsymbol{l}_k$ and $\boldsymbol{s}_k$ of all the links. Note that we shall be able to obtain the quantity ${}^I\overline{\boldsymbol{r}}_k$ for $k = 0, 1, \cdots, n$ in the process of this computation. ${}^I\overline{\boldsymbol{r}}_k$ would be required for computing the Jacobian $\boldsymbol{J}_2^k$ in step 9. [$18nM + (15n + 9)A$.]
5) Compute the quantity $\sum_{k=0}^{n} {}^I\boldsymbol{A}_k\,{}^k\boldsymbol{I}_k\,{}^I\boldsymbol{A}_k^T$. Each term of the sum, i.e., ${}^I\boldsymbol{A}_k\,{}^k\boldsymbol{I}_k\,{}^I\boldsymbol{A}_k^T \in R^{3\times 3}$, is a symmetric matrix, and therefore we need to evaluate only six of its nine elements. The summation is performed in the reverse order, i.e., beginning with $k = n$ and ending with $k = 0$. This is useful because we need to compute $\sum_{k=i}^{n} {}^I\boldsymbol{A}_k\,{}^k\boldsymbol{I}_k\,{}^I\boldsymbol{A}_k^T$, for $i = 1, 2, \cdots, n$, in step 9. [$45(n + 1)M + (39n + 30)A$.]
6) Compute $\dot{\boldsymbol{J}}_1\,\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{J}}_2\,\dot{\boldsymbol{\theta}}_2$ using (25). We apply the forward recursion of Newton–Euler recursive scheme to evaluate ${}^I\ddot{\boldsymbol{r}}_n$ and ${}^I\dot{\omega}_n$ subject to the constraints $\ddot{\boldsymbol{\theta}}_1 = \ddot{\boldsymbol{\theta}}_2 = 0$. Specifically, we use (A1) through (A4) of Appendix A, subject to the constraints $\ddot{\boldsymbol{\theta}}_1 = \ddot{\boldsymbol{\theta}}_2 = 0$. Though (A4) is redundant here, it is useful for computation in the next step. [$45nM + (39n - 9)A$.]
7) Compute $\dot{\boldsymbol{H}}_1\,\dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{H}}_2\,\dot{\boldsymbol{\theta}}_2$ using (26). We have from the previous step the linear acceleration of the center of mass and the angular acceleration of each link under the condition $\ddot{\boldsymbol{\theta}}_1 = \ddot{\boldsymbol{\theta}}_2 = 0$. We can therefore compute the generalized inertial forces of each link when $\ddot{\boldsymbol{\theta}}_1 = \ddot{\boldsymbol{\theta}}_2 = 0$. These inertial forces can be converted into equivalent force-moment pairs acting at the center of mass of the total system. $\dot{\boldsymbol{b}}\,\big|_{\ddot{\boldsymbol{\theta}}_1=\ddot{\boldsymbol{\theta}}_2=0}$ is the algebraic sum of these force-moment pairs. [$33(n + 1)M + (30n + 24)A$.]
8) Compute $\boldsymbol{H}_1^{-1}$ explicitly from (27) as a product of the three matrices. The major part of this computation has already been performed in step 5. [$(9n + 117)M + (3n + 72)A$.]
9) Compute $\boldsymbol{H}_2$ from (12). Clearly we are required to compute $\boldsymbol{J}_2^k$, as given by (5)–(7), for $k = 1, 2, \cdots, n$. Note that most of the computation for the matrix $\boldsymbol{P}$ given by (14) that is required for the computation of $\boldsymbol{H}_2$ has already been done in step 5. [$(7.5n^2 + 19.5n + 3)M + (7.5n^2 + 10.5n)A$.]

10) The matrices J_1 and J_2^n as given by (16) need very little additional computation. J_2^n was already evaluated in the previous step. [$3A$.]
11) Compute the values of $\ddot{\theta}_1$, and $\ddot{\theta}_2$, from (23) by using (29). Note that it is computationally less expensive to repeat the matrix-vector product three times by substituting (29) into (23) than to compute the matrix of (29) explicitly. Also note that the matrix-vector multiplications are simplified due to the presence of null and identity blocks in the three matrices of (29).
12) If $\dot{\theta}_1$ cannot be measured, then compute it using (24).
13) Compute the link velocities and accelerations using Newton–Euler recursive scheme and the computed quantities $\dot{\theta}_1, \ddot{\theta}_1, \dot{\theta}_2$, and $\ddot{\theta}_2$. Use (A1) through (A4) of Appendix A. Since we computed the forward recursion of Newton–Euler recursive scheme subject to the constraints $\ddot{\theta}_1 = \ddot{\theta}_2 = 0$ in step 6, we need to repeat the computation of (A2)–(A4) with the fact in mind that many of the terms do not change. [$15nM + (24n - 9)A$.]
14) Compute the inertial forces and inertial moments due to each link using (B1) of Appendix B. Note that the terms ${}^l\omega_k \times ({}^l I_k {}^l\omega_k)$ were already evaluated in step 7. [$12(n+1)M + 9(n+1)A$.]
15) Compute the forces exchanged between the links using (C1) from Appendix C. Finally, compute the joint torques using (D1) from Appendix D. [$15nM + 20nA$.]

Consider once again the computation required in step 11, where we invert the coefficient matrix of (22). As already mentioned in Section III, for $n \neq 6$, we need to use the pseudoinverse to solve for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ from (22), and we cannot use (23). Therefore, the number of computations required cannot be expressed conveniently as a function of n. For a comparison with the method proposed by Yamada [14], we evaluate the number of computations for $n = 6$. We need 330 multiplications and 330 additions. This number, however, does not include the computation necessary to invert the generalized Jacobian matrix $\tilde{J}$ as in (29). We excluded this computation simply because it was not included in the existing approach by Yamada [14]. For the same reason, the computation required for step 12 also has not been considered.

B. Computational Burden

The amount of computation necessary for solving the inverse dynamics problem using our algorithm is given as $(7.5n^2 + 241.5n + 210)M$ and $(7.5n^2 + 211.5n + 128)A$, where M and A refer to multiplications and additions, respectively. Additionally, for $n = 6$, we need $330M + 330A$. Therefore, for a 6-DOF manipulator, our approach requires a total of 2259 multiplications and 1992 additions.

C. Comparison of the Two Approaches

Yamada [14] developed an efficient scheme to compute the inverse dynamics along with the resolved acceleration control for space robots in the absence of external forces. The computational burden for his scheme was $(328n + 84)M$ and $(309n + 51)A$. His scheme was efficient because he did not explicitly compute the momentum constraints. He developed an effective algorithm to compute the generalized Jacobian and used it to readily obtain the joint trajectories necessary to track a given end-effector trajectory. The Newton–Euler recursive scheme was then applied only to compute the inertial forces of each link (not to compute the joint torques) by using sensor readings for the linear and angular velocity of the vehicle and using zero values for the linear and angular acceleration of the vehicle. It was argued that the actual value of the vehicle acceleration could be computed from the inertial force of the vehicle obtained at the end of the recursion. After knowing the true values of the vehicle acceleration, the Newton–Euler recursive scheme was repeated to compute the joint torques this time. In the second recursion, duplication of computation done during the first recursion was carefully avoided.

We computed the momentum constraints, but reduced the computational amount by considering the common terms. The salient feature of our method is that we can compute the dependent as well as the independent variables explicitly. For a 6-DOF manipulator, Yamada's method requires $2052M$ and $1905A$ compared to $2259M$ and $1992A$ as in our case, which is 9.16% fewer multiplications and 4.36% fewer additions. Although Yamada's method is slightly more efficient than ours, our approach has four major advantages:

First, when external forces are present, we need to consider them while summing up forces on individual links; this is also true in the momentum constraint equations. Yamada uses the generalized Jacobian matrix, which assumes momentum conservation with zero initial momentum. His approach, therefore, requires modification with a fairly large additional computation cost in order to consider external forces. Our approach can take care of external forces.

Second, in our approach, $\dot{\theta}_1$ can be computed from $\dot{\theta}_2$ and the current momentum of the system. This will be particularly useful when $\dot{\theta}_1$ cannot be measured due to the absence of sensors or when sensors develop technical problems. When $\dot{\theta}_1$ is measured by sensors, our approach can be utilized to increase the reliability of the measurement of $\dot{\theta}_1$. Yamada's approach however, cannot compute $\dot{\theta}_1$. It relies entirely on sensor information.

Third, when the manipulator DOF is greater than six, we have kinematic redundancy. In that case, the solution for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ can be obtained from (22) by using the pseudoinverse. The redundancy, therefore, can be used to plan the motion of both $\ddot{\theta}_1$ and $\ddot{\theta}_2$. Yamada's approach uses the generalized Jacobian matrix as in (18), which eliminates dependent variables. Therefore, his approach can utilize kinematic redundancy to plan the motion of $\ddot{\theta}_2$ alone.

Last, parallel recursion provides the possibility of a substantial further reduction in computational time for our approach. Parallel recursion for space robots is discussed next.

It is true that the computational burden of our approach is quadratic in n whereas the computational burden of Yamada's approach is proportional to n. However, the derivatives of the computational burden with respect to n, for $n = 6$, for the two approaches are almost the same. The figures are 331.5 for multiplications and 301.5 for additions for our approach, as compared to 328 for multiplications and 309 for additions for Yamada's approach.

D. Parallel Recursion in Space Robots

In the case of terrestrial robots, we usually compute the link velocities and accelerations by forward recursion, i.e., by starting at the base and proceeding toward the end-effector. This is because we know that the base is fixed, and therefore it has zero velocity and acceleration. If we are provided with the velocity and acceleration of the end-effector, as in resolved acceleration control, we can compute the link velocities and accelerations by backward recursion, i.e., by starting at the end-effector and proceeding toward the base. For space robots, we can compute the vehicle motion and also the end-effector motion from the motion of the joints of the manipulator using constraint and kinematic equations. Subsequently we can compute the kinematics of a space robot by beginning computation either at the vehicle or the end-effector. The kinematics for terrestrial and space robots alike, therefore, can be computed by forward or backward recursion.

Once the link velocities and accelerations are obtained in the inertial frame of reference, the inertial forces and moments of each link—for terrestrial robots as well as space robots—can be obtained straightforwardly using (B1) of Appendix B. After evaluation of the inertial forces, it is necessary to obtain the forces exchanged between the links. In the case of terrestrial robots, this can be obtained only by backward recursion. This is because the first link is fixed to the ground, and therefore the forces and moments it exchanges with the ground are completely unknown initially. Clearly, it is necessary to start computation of the generalized joint forces from the end that is free to move, i.e., the end-effector. For a space robot, both the vehicle and the end-effector are free to move. The computation of the generalized joint forces for space robots, therefore, can be performed by forward or backward recursion.

From the above discussion it is clear that, for space robots, the kinematics and the dynamics can be computed simultaneously. Moreover, they can be computed by forward as well as by backward recursion. In fact, we can use four different processors, two each for kinematics and dynamics, to compute from the two different ends, i.e., the vehicle and the end-effector. Going back to our algorithm in Section IV, parallel recursion is to be used for steps 6, 7, 13, 14, and 15. The amount of computation required in these steps is of the order of $(120n + 45)M$ and $(122n + 15)A$, which is close to 50% of the net computation. Parallel recursion, therefore, can be used to reduce this computation to almost 25%.

V. Concluding Remarks

An inverse dynamics algorithm was discussed in this paper in the framework of resolved acceleration control. We assumed that we are provided with the trajectory of the end-effector $\ddot{\boldsymbol{x}}$ and we also know all the external forces, including the thruster forces, that act upon the system along with their points of application. We wish to emphasize that the momentum constraints play an important role in determining the motion of the system and that the efficient computation is very important for resolved motion rate control.

In a different situation, we may be required to compute the thruster forces before solving for the inverse dynamics. In such a situation, all other external forces along with their points of application would be known. We would also be provided with the trajectory of the end-effector $\ddot{\boldsymbol{x}}_E$ and the motion of the space vehicle $\ddot{\boldsymbol{\theta}}_1$. A practical example of this situation is illustrated in Fig. 2. In this figure, the two-arm space robot is moving from one point of the space station to another. In this situation, the external force on the system at the end-effector, which interacts with the space station, is provided as its desired value. The thruster forces that will act on the space vehicle will have to be computed. Clearly, knowledge of only the trajectory of the active end-effector $\ddot{\boldsymbol{x}}_E$ (which is stationary in this case) is insufficient; we also need to know $\ddot{\boldsymbol{\theta}}_1$. Using purely kinematic relations, we would first compute all the joint motions $\ddot{\boldsymbol{\theta}}_2$ from $\ddot{\boldsymbol{x}}_E$ and $\ddot{\boldsymbol{\theta}}_1$ using (21). As the next step, we may compute the thruster forces using the momentum constraints given by (20) before computing the joint torques. Alternatively, we may compute the joint torques before computing the thruster forces by using the algorithm of [6]. In either procedure, we first compute the link velocities and accelerations from known values of $\ddot{\boldsymbol{\theta}}_1$ and $\ddot{\boldsymbol{\theta}}_2$ using the forward recursion of the Newton–Euler recursive scheme. Following this, we compute the generalized inertial forces of each link. If we are to compute the unknown thruster forces before computing the joint torques, the inertial forces are converted into equivalent force-moment pairs acting at the origin of the inertial frame. $\dot{\boldsymbol{b}}$ is obtained as the algebraic sum of these force-moment pairs. Since $\dot{\boldsymbol{b}}$ is also the vector sum of the all the generalized external forces acting at the origin of the inertial frame, the thruster forces can be easily computed from $\dot{\boldsymbol{b}}$ and knowledge of all other external forces. The joint torques can be computed subsequently. There is one clear advantage of computing the thruster forces first. With the complete knowledge of all external forces (including thruster forces) and the motion of the system, the joint torques can be computed using both forward and backward recursion (parallel recursion). If the thruster forces are not computed first, we can compute the joint torques by using only the backward recursion of the Newton–Euler recursive scheme [6]. Therefore, our algorithm can exploit parallel recursion to reduce the computation time. In the absence of parallel recursion, the computations required for the two methods are almost similar.

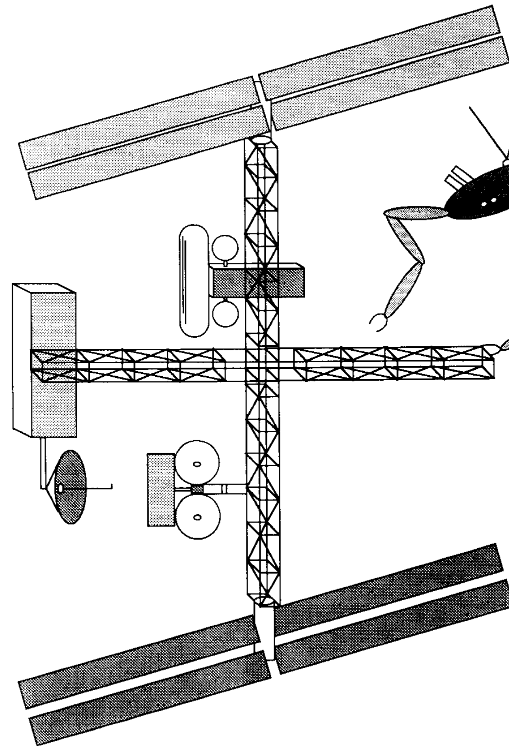

Fig. 2. A two-arm space robot moving from one point of a space station to another.

In the inverse dynamics algorithm discussed in this paper, momentum constraints were used to compute the motion of the system. In the example discussed above, momentum constraints were shown to be useful in computing the external generalized forces. There would exist various different scenarios in addition to these two for motion control and force control of space robots. The momentum constraints will appear in all of them. We believe that the efficient computation algorithm proposed in this paper will serve as a fundamental tool in any scenario for space robot applications.

To summarize, this paper provides a solution to the inverse dynamics problem for space robots. Inverse kinematics and inverse dynamics were considered together as a special approach toward solving the problem. The inverse dynamics algorithm emphasizes the explicit evaluation of the momentum constraints. An efficient computational scheme is proposed to support the algorithm. The existing approach by Yamada [14] requires 9.16% fewer multiplications and 4.36% fewer additions. However, our approach has major advantages. One such advantage is that our approach solves the inverse dynamics taking external forces into account whereas the existing one requires modification and additional computation. The possibility of reduction in computational time by using parallel recursion was also discussed as a feature of our algorithm.

Appendix A
Computation of Link Velocities and Accelerations

The commonly used equations for the computation of velocities and accelerations of the $(k+1)$th D-H link frame, measured in the inertial frame, are

$$ {}^I\omega_{k+1} = {}^I\omega_k + {}^I\boldsymbol{A}_k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_{k+1} \tag{A1} $$

$$ {}^I\dot{\omega}_{k+1} = {}^I\dot{\omega}_k + {}^I\boldsymbol{A}_k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ddot{q}_{k+1} + {}^I\omega_k \times {}^I\boldsymbol{A}_k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{q}_{k+1} \tag{A2} $$

$$ \begin{aligned} {}^I\ddot{\bar{\boldsymbol{r}}}_{k+1} &= {}^I\ddot{\bar{\boldsymbol{r}}}_k + {}^I\dot{\omega}_{k+1} \times \left({}^I\boldsymbol{A}_{k+1} l_{k+1}\right) \\ &\quad + {}^I\omega_{k+1} \times \left({}^I\omega_{k+1} \times \left({}^I\boldsymbol{A}_{k+1} l_{k+1}\right)\right) \end{aligned} \tag{A3} $$

where k varies from $0, 1, \cdots, n-1$. The acceleration of the center of mass of a link that is required for calculation of inertial forces is obtained as

$$ \begin{aligned} {}^I\ddot{\boldsymbol{r}}_{k+1} &= {}^I\ddot{\bar{\boldsymbol{r}}}_{k+1} + {}^I\dot{\omega}_{k+1} \times \left({}^I\boldsymbol{A}_{k+1} s_{k+1}\right) \\ &\quad + {}^I\omega_{k+1} \times \left({}^I\omega_{k+1} \times \left({}^I\boldsymbol{A}_{k+1} \boldsymbol{s}_{k+1}\right)\right) \end{aligned} \tag{A4} $$

where k varies from $0, 1, \cdots, n$ and ${}^I\omega_0$, ${}^I\dot{\omega}_0$ and ${}^I\ddot{\boldsymbol{r}}_0$ are obtained from the trajectories of $\dot{\boldsymbol{\theta}}_1$ and $\ddot{\boldsymbol{\theta}}_1$. The trajectories of $\dot{\boldsymbol{\theta}}_2$ and $\ddot{\boldsymbol{\theta}}_2$ enable us to evaluate the values of $\dot{q}_{k+1}$ and $\ddot{q}_{k+1}$ for k varying from $0, 1, \cdots, n-1$.

Appendix B
Computation of Inertial Forces

Once the link velocities and accelerations are obtained in the inertial frame of reference, the inertial forces and moments acting on each link can be obtained straightforwardly using the following well known equations:

$$ \begin{aligned} {}^I\boldsymbol{F}_k^* &= m_k {}^I\ddot{\boldsymbol{r}}_k \\ {}^I\boldsymbol{N}_k^* &= {}^I\boldsymbol{I}_k {}^I\dot{\omega}_k + {}^I\omega_k \times \left({}^I\boldsymbol{I}_k {}^I\omega_k\right) \end{aligned} \tag{B1} $$

where ${}^I\boldsymbol{F}_k^*$ and ${}^I\boldsymbol{N}_k^*$ are the inertial force and inertial moment of the kth link, respectively.

Appendix C
Computation of Forces Exchanged Between Links

The following well known equations are to be used for computation of the reaction forces:

$$ \begin{aligned} {}^I\boldsymbol{F}_{k-1} &= {}^I\boldsymbol{F}_k - {}^I\boldsymbol{F}_k^* \\ {}^I\boldsymbol{N}_{k-1} &= {}^I\boldsymbol{N}_k - {}^I\boldsymbol{N}_k^* - \left({}^I\boldsymbol{r}_k - {}^I\bar{\boldsymbol{r}}_k\right) \times {}^I\boldsymbol{F}_k \\ &\quad + \left({}^I\boldsymbol{r}_k - {}^I\bar{\boldsymbol{r}}_{k-1}\right) \times {}^I\boldsymbol{F}_{k-1}. \end{aligned} \tag{C1} $$

Appendix D
Computation of Joint Torques

Once the reaction forces exchanged between the links are known, the joint torques can be computed by using this simple relation:

$$ \tau_k = -\left({}^I\boldsymbol{N}_{k-1}\right)^T {}^I z_{k-1} \tag{D1} $$

where ${}^I\boldsymbol{N}_{k-1}$ is the reaction moment exchanged between the $(k-1)$th link and the kth link. Although all the joints of the space robot were assumed to be revolute in our case, it would be easy to extend the results to include prismatic joints.

References

[1] H. L. Alexander and R. H. Cannon, "Experiments on the control of a satellite manipulator," in *Proc. Amer. Control Conf.* (Seattle), 1987.

[2] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower pair mechanisms based on matrices," *J. Appl. Mechanics*, vol. 22, 1955.

[3] R. Featherstone, *Robot Dynamics Algorithm.* Norwell, CT: Kluwer, 1987.

[4] S. Kodama and N. Suda, *Matrix Theory for Systems Control.* Tokyo: Soc. Instruments Control Eng., 1978, pp. 38–43 (in Japanese).

[5] R. W. Longman, R. E. Lindberg, and M. F. Zedd, "Satellite-mounted robot manipulators: New kinematics and reaction moment compensation," *Int. J. Robotics Res.*, vol. 6, no. 3, pp. 87–103, 1987.

[6] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "On-line computational scheme for mechanical manipulators," *J. Dynam. Syst., Measurement, Control*, vol. 102, no 2, pp. 69–76, 1980.

[7] ——, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.

[8] F. Miyazaki, Y. Masutani, S. and Arimoto, "Sensor feedback using approximate Jacobian," in *Proc. USA-Japan Symp. Flexible Automat.* (Minneapolis), 1988, pp. 139–145.

[9] Y. Nakamura and R. Mukherjee, "Bi-directional approach for nonholonomic path planning of space robots," in *Proc. 5th Int. Symp. Robotics Res.* (Tokyo), 1989, pp. 405–416.

[10] ——, "Nonholonomic path planning of space robots via bidirectional approach" *IEEE Trans. Robotics Automat.*, vol. 7, no. 4, pp. 500–514, 1991.

[11] Y. Umetani and K. Yoshida, "Continuous path control of space manipulators mounted on OMV," *Acta Astronaut.*, vol. 15, no. 12, pp. 981–986, 1987.

[12] Z. Vafa, "The kinematics, dynamics and control of space manipulators: The virtual manipulator concept," Ph.D. dissertation, Dept. Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[13] Z. Vafa and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach," in *Proc. IEEE Int. Conf. Robotics Automat.* (Raleigh, NC), 1987.

[14] K. Yamada, "Formulation of space multi-body systems and its application to control," Ph.D. dissertation, Dept. of Industrial Machinery Eng., Univ. of Tokyo, Tokyo, 1989 (in Japanese).