On the Nature of Control Algorithms for Free-Floating Space Manipulators

Evangelos Papadopoulos, *Member, IEEE,* and Steven Dubowsky, *Member, IEEE*

***Abstract*—This paper strongly suggests that *nearly any* control algorithm that can be used for fixed-based manipulators also can be employed in the control of free-floating space manipulator systems, with the additional conditions of estimating or measuring a spacecraft's orientation and of avoiding dynamic singularities. This result is based on the structural similarities between the kinematic and dynamic equations of a free-floating space system and the same equations for the same manipulator but with a fixed base. Barycenters are used to formulate the kinematic and dynamic equations of free-floating space manipulators. A control algorithm for a space manipulator system is designed to demonstrate the value of the analysis. The results obtained should encourage the development of a wide variety of control algorithms for free-floating space manipulator systems.**

I. Introduction

The planning and control of the robotic manipulators expected to play important roles in future space missions, such as the Flight Telerobotic Servicer shown in Fig. 1, pose additional problems beyond those found in fixed-based manipulators due to the dynamic coupling between space manipulators and their spacecraft. A number of control techniques for such systems have been proposed. These schemes can be classified into three categories. In the first, reaction jets control spacecraft position and attitude, compensating for any manipulator dynamic forces exerted on the spacecraft. Control laws for earth-bound manipulators can be used in this case, but their utility will be limited if the manipulator motions can saturate the reaction jet system. Reaction jets also may consume relatively large amounts of attitude control fuel, limiting the useful life of the system [1], [2]. In the second category, reaction wheels or jets control a spacecraft's attitude but not its translation [3], [4]. The control of these systems is somewhat more complicated than for the first category, although a technique called the virtual manipulator (VM) method can be used to simplify the problem [4]–[6]. In the third category, free-floating systems have been proposed in order to conserve fuel or electrical power [5]–[9]. These permit the manipulator's spacecraft to move freely in response to the manipulator motions. Since the spacecraft's attitude control system does not operate during this mode of space manipulation, this mode becomes feasible when no external forces and torques act on the system and when its total momentum is negligible. In practice, momentum dump maneuvers are employed to remove any momentum that may accumulate [12]. Free-floating manipulators also can be modeled using the VM approach [5], [6]. Past control algorithms for free-floating systems have been proposed and their validity demonstrated on a case by case basis, [7]–[10]. Algorithms for such systems that ignore the kinematics or dynamics of the spacecraft in their formulation have been found to have problems [9], [10]. These problems may be attributed to *dynamic singularities* that are not found in earthbound manipulators [11], [12].

Manuscript received January 2, 1990; revised March 20, 1991. This work was supported by the NASA Langley Research Center, Automation Branch, under Grant NAG-1-801. Portions of this paper were presented at the 1990 IEEE Robotics and Automation Conference, Cincinnati, OH.

E. Papadopoulos was with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139. He is now with McGill University, Montreal, Quebec, Canada H3A 2B4.

S. Dubowsky is with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9101599.

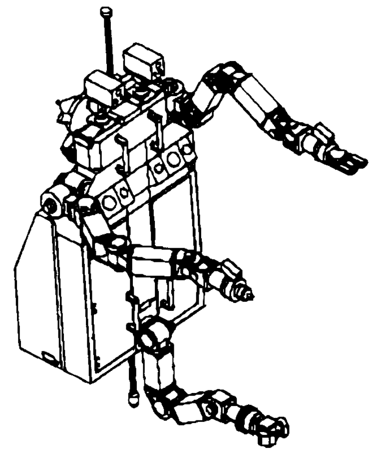


Fig. 1. The Flight Telerobotic Servicer (FTS).

This paper takes a more fundamental approach to the question: "What control algorithms can be applied to the motion control of free-floating space manipulators?" The results obtained show that nearly any algorithm that can be applied to conventional fixed-based manipulators can be directly applied to free-floating manipulators, with a few weak additional conditions. These include the measurement or estimation of a spacecraft's orientation and the avoidance of dynamic singularities. These results should encourage the development of a wide range of control algorithms for free-floating space manipulator systems.

II. Dynamic Modeling of Free-Floating Systems

This section develops the dynamic equations of a rigid free-floating space manipulator system, see Fig. 2, using a Lagrangian approach. The body 0 in Fig. 2 represents the

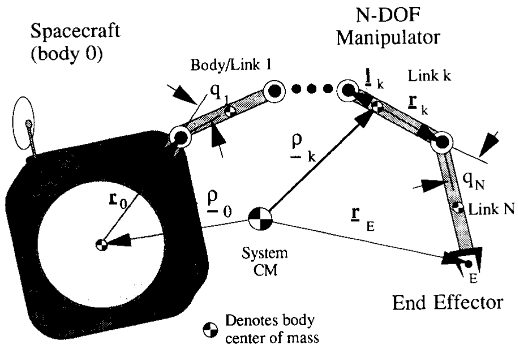


Fig. 2. A spatial free-floating space manipulator system.

spacecraft; the bodies $k(k = 1, \ldots, N)$ represent the N-manipulator links. The manipulator joint angles and velocities are represented by the $N \times 1$ column vectors $\mathbf{q}$ and $\dot{\mathbf{q}}$. The spacecraft's attitude control system is turned off when the system is operating in a free-floating mode and hence the spacecraft can translate and rotate in response to manipulator movements.

For simplicity, it will be assumed that the system has a single manipulator with revolute joints and is in an open-chain kinematic configuration so that, in a system with an N-degree-of-freedom (DOF) manipulator, there will be $6 + N$ DOF. The method of analysis that follows, as well as its results, hold equally well if these assumptions are removed.

First, the system kinetic energy is expressed as a function of the generalized coordinates and their velocities. Under the assumptions of free-floating operation, i.e., the absence of external forces and of zero initial momentum, the system center of mass (CM) remains fixed in inertial space and the inertial origin, O, can be chosen to be the CM. The system kinetic energy T can be written as

$$T = \frac{1}{2} \sum_{k=0}^{N} \{\underline{\omega}_k \cdot \underline{\mathbf{I}}_k \cdot \underline{\omega}_k + m_k \dot{\underline{\rho}}_k \cdot \dot{\underline{\rho}}_k\} \tag{1}$$

where m_k is the mass of the kth body $\underline{\mathbf{I}}_k$ is its inertia dyadic with respect to its center of mass, and $\dot{\underline{\rho}}_k$ and $\underline{\omega}_k$ are its linear and angular inertial velocities, respectively. It can be shown that T can be written in a more compact form as a function of the $N + 1$ angular velocities as [11], [12]:

$$T = \frac{1}{2} \sum_{j=0}^{N} \sum_{i=0}^{N} \underline{\omega}_i \cdot \underline{\mathbf{D}}_{ij} \cdot \underline{\omega}_j \tag{2}$$

where the $\underline{\mathbf{D}}_{ij}$ terms are inertia dyadics that are functions of the mass and inertia distribution of the space manipulator system, and are given by [11], [12]:

$$\underline{\mathbf{D}}_{ij} \equiv \begin{cases} -M\{(\underline{\mathbf{l}}_j^* \cdot \underline{\mathbf{r}}_i^*)\mathbf{1} - \underline{\mathbf{l}}_j^*\underline{\mathbf{r}}_i^*\}, & i < j \\ \underline{\mathbf{I}}_i + \sum_{k=0}^{N} m_k\{(\underline{v}_{ik} \cdot \underline{v}_{ik})\mathbf{1} - \underline{v}_{ik}\underline{v}_{ik}\}, & i = j. \\ -M\{(\underline{\mathbf{r}}_j^* \cdot \underline{\mathbf{l}}_i^*)\mathbf{1} - \underline{\mathbf{r}}_j^*\underline{\mathbf{l}}_i^*\}, & i > j \end{cases} \tag{3}$$

In (3), M is the total system mass, $\underline{\mathbf{1}}$ is the unit dyadic. The vectors $\underline{v}_{ik}$ $(i,\ k = 0, \ldots, N)$, $\underline{\mathbf{r}}_i^*$ and $\underline{\mathbf{l}}_i^*(i = 0, \ldots, N)$ are defined by the barycenters [13], [14] of the ith body. First, the body fixed vector $\underline{\mathbf{c}}_i$ is defined as referring to the location of the ith body's barycenter with respect to the body's CM. It can be shown that $\underline{\mathbf{c}}_i$ is equal to:

$$\underline{\mathbf{c}}_i = \underline{\mathbf{l}}_i\mu_i + \underline{\mathbf{r}}_i(1 - \mu_{i+1}), \qquad i = 0, \ldots, N \tag{4}$$

where μ_i represents the mass distribution given by

$$\mu_i \equiv \begin{cases} 0, & i = 0 \\ \sum_{j=0}^{i-1} \frac{m_i}{M}, & i = 1, \ldots, N. \\ 1, & i = N + 1 \end{cases} \tag{5}$$

It might be noted that the barycenter of the ith body can be found equivalently by adding a point mass equal to $M\mu_i$ to joint i, and a point mass equal to $M(1 - \mu_{i+1})$ to joint $i + 1$, forming an *augmented body* [13], [14]. The barycenter is then the center of mass of the augmented body (see Fig. 3). Fig. 3 also shows the body-fixed vectors $\underline{\mathbf{r}}_i^*$ and $\underline{\mathbf{l}}_i^*$ required by (3) and the vector $\underline{\mathbf{c}}_i^*$, which can be written as

$$\underline{\mathbf{c}}_i^* = -\underline{\mathbf{c}}_i \tag{6a}$$

$$\underline{\mathbf{r}}_i^* = \underline{\mathbf{r}}_i - \underline{\mathbf{c}}_i \tag{6b}$$

$$\underline{\mathbf{l}}_i^* = \underline{\mathbf{l}}_i - \underline{\mathbf{c}}_i. \tag{6c}$$

Finally, the vectors v_{ik} in (3) are defined by

$$\underline{v}_{ik} \equiv \begin{cases} \underline{\mathbf{r}}_i^*, & i < k \\ \underline{\mathbf{c}}_i^*, & k = i. \\ \underline{\mathbf{l}}_i^*, & i > k \end{cases} \tag{7}$$

Equation (2) is a compact representation of the system's kinetic energy, but it is convenient to work with a scalar (matrix) form of the equation. To this end, we use the following notation. Bold lowercase symbols represent column vectors; bold uppercase symbols represent matrices. Right superscripts are interpreted as "with respect to," left superscripts as "expressed in frame." A missing left superscript implies a column vector expressed in the inertial frame. In addition, we introduce $N + 1$ reference frames, each one attached to the CM of each body, with axes parallel to the body's principal axes. Hence, the body inertia matrix expressed in this frame is diagonal.

The system kinetic energy is written in matrix form as follows. The inertial angular velocity of body j, expressed by the vector $\boldsymbol{\omega}_j$, can be written as the sum of the inertial angular velocity of the spacecraft, $\boldsymbol{\omega}_0$, and the inertial angular velocity of body j relative to the spacecraft $\boldsymbol{\omega}_j^0$:

$$\boldsymbol{\omega}_j = \boldsymbol{\omega}_0 + \boldsymbol{\omega}_j^0 \qquad j = 1, \ldots, N. \tag{8}$$

The angular velocity $\boldsymbol{\omega}_j^0$ can be expressed as a function of the joint angles $\mathbf{q}$. This is accomplished by defining a 3×3 transformation matrix ${}^{i-1}\mathbf{A}_i(q_i)(i = 1, \ldots, N)$, which is a function of q_i (the ith relative joint angle), and which transforms a column vector expressed in frame i to a column vector expressed in frame $i - 1$. An additional transforma-

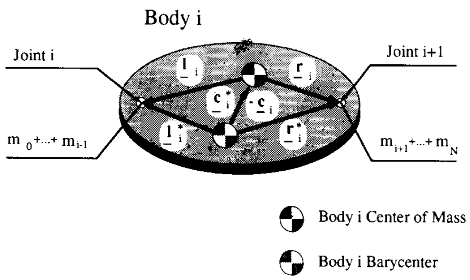


Fig. 3. Definition of vectors $\mathbf{r}_i^*$, $\mathbf{l}_i^*$, $\mathbf{c}_i^*$.

tion matrix can be defined as

$$ {}^0\mathbf{T}_i(q_1, \ldots, q_i) = {}^0\mathbf{A}_1(q_1) \ldots {}^{i-1}\mathbf{A}_i(q_i), \qquad i = 1, \ldots, N. \tag{9} $$

${}^0\mathbf{T}_i$ transforms a column vector expressed in frame i to a column vector in frame 0. Finally, $\mathbf{T}_0$ is the transformation matrix from the spacecraft frame to the inertial frame and is a function of the spacecraft's attitude, expressed by the Euler parameters ϵ and η, $\mathbf{T}_0 = \mathbf{T}_0(\epsilon, \eta)$, see [15]. Using these transformation matrices, the inertial velocity ω_j^0 of link j relative to the spacecraft can be written as

$$ \omega_j^0 = \mathbf{T}_0 \sum_{i=1}^{j} {}^0\mathbf{T}_i {}^i\mathbf{u}_i \dot{q}_i = \mathbf{T}_0 {}^0\mathbf{F}_j \dot{\mathbf{q}}, \qquad j = 1, \ldots, N \tag{10} $$

where ${}^i\mathbf{u}_i$ is the unit column vector in frame i parallel to the axis of revolution through joint i, and ${}^0\mathbf{F}_j$ is a $3 \times N$ matrix, function of the joint angles $\mathbf{q}$ only and given by

$$ {}^0\mathbf{F}_j(\mathbf{q}) \equiv \left[{}^0\mathbf{T}_1 {}^1\mathbf{u}_1, {}^0\mathbf{T}_2 {}^2\mathbf{u}_2, \ldots, {}^0\mathbf{T}_j {}^j\mathbf{u}_j, \mathbf{0}\right], \qquad j = 1, \ldots, N \tag{11} $$

where $\mathbf{0}$ is a $3 \times (N - j)$ zero element matrix. It is easy to show that the inertia matrices $\mathbf{D}_{ij}$ that correspond to the inertia dyadics given by (3) can be expressed with respect to the spacecraft frame of reference as

$$ \mathbf{D}_{ij} = \mathbf{T}_0 {}^0\mathbf{D}_{ij} \mathbf{T}_0^T, \qquad i, j = 1, \ldots, N. \tag{12} $$

${}^0\mathbf{D}_{ij}$ is formed according to (3) with all vectors expressed in the base frame and is a function of the configuration $\mathbf{q}$ only. Also, due to (3), ${}^0\mathbf{D}_{ij} = {}^0\mathbf{D}_{ji}^T$. For convenience, define

$$ {}^0\mathbf{D}_j \equiv \sum_{i=0}^{N} {}^0\mathbf{D}_{ij} \qquad j = 0, \ldots, N \qquad {}^0\mathbf{D} \equiv \sum_{j=0}^{N} {}^0\mathbf{D}_j \tag{13a} $$

$$ {}^0\mathbf{D}_\mathbf{q} \equiv \sum_{j=1}^{N} {}^0\mathbf{D}_j {}^0\mathbf{F}_j \qquad {}^0\mathbf{D}_{\mathbf{qq}} \equiv \sum_{j=1}^{N} \sum_{i=1}^{N} {}^0\mathbf{F}_i^T {}^0\mathbf{D}_{ij} {}^0\mathbf{F}_j \tag{13b} $$

where all the above are functions of $\mathbf{q}$ only. The terms in (13b) depend on the manipulator mass and inertia properties, whereas the terms in (13a), in addition, depend on the spacecraft inertia.

Using (8)–(13), the matrix form of (2) can be written as

$$ T = \frac{1}{2} {}^0\omega_0^{T0}\mathbf{D}^0\omega_0 + \frac{1}{2} {}^0\omega_0^{T0}\mathbf{D}_\mathbf{q}\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^{T0}\mathbf{D}_\mathbf{q}^{T0}\omega_0 + \frac{1}{2}\dot{\mathbf{q}}^{T0}\mathbf{D}_{\mathbf{qq}}\dot{\mathbf{q}} \tag{14} $$

where ${}^0\omega_0$ is the spacecraft angular velocity expressed in its frame. Note that T is a function of ${}^0\omega_0$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ only. This observation suggests that if ${}^0\omega_0$ can be expressed as a function of $\mathbf{q}$ and $\dot{\mathbf{q}}$, then the spacecraft attitude coordinates are *ignorable,* that is they do not appear in the expression for T, see [16]. It turns out that this can be done under the assumptions of free-floating operation, i.e., the absence of external forces and torques and of zero initial momentum.

Indeed, the system angular momentum $\mathbf{h}_{CM}$ can be written as [11], [12]:

$$ \mathbf{h}_{CM} = \mathbf{T}_0 \sum_{j=0}^{N} \sum_{i=0}^{N} {}^0\mathbf{D}_{ij} {}^0\omega_j = \mathbf{T}_0({}^0\mathbf{D}^0\omega_0 + {}^0\mathbf{D}_\mathbf{q}\dot{\mathbf{q}}). \tag{15} $$

In the absence of external torques, the system angular momentum is constant. We further assume that during free-floating operation, the system momentum is zero. If momentum accumulates, the system may be able to continue operating for a limited period of time. In practice, the spacecraft's attitude control system would be turned on and perform a momentum dump maneuver in order to eliminate any accumulated momentum [12]. Note that the first two terms in (14) vanish if the system angular momentum is zero. By setting $\mathbf{h}_{CM} = 0$ in (15), the spacecraft angular velocity ${}^0\omega_0$ is written as a function of the joint rates $\dot{\mathbf{q}}$:

$$ {}^0\omega_0 = -{}^0\mathbf{D}^{-1\,0}\mathbf{D}_\mathbf{q}\dot{\mathbf{q}}. \tag{16} $$

Note that inversion of ${}^0\mathbf{D}$ is always possible because it is a symmetric positive definite matrix that represents the inertia of the free-floating system about its CM. Substituting ${}^0\omega_0$ in (9) and after some algebraic manipulation, T results in

$$ T = \tfrac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}^*(\mathbf{q})\dot{\mathbf{q}} \tag{17} $$

where $\mathbf{H}^*(\mathbf{q})$ is the system inertia matrix, given by

$$ \mathbf{H}^*(\mathbf{q}) \equiv {}^0\mathbf{D}_{\mathbf{qq}} - {}^0\mathbf{D}_\mathbf{q}^{T0}\mathbf{D}^{-1\,0}\mathbf{D}_\mathbf{q}. \tag{18} $$

Again, the fact that ${}^0\mathbf{D}_{ij} = {}^0\mathbf{D}_{ji}^T$ was used. The above equations are important because they show how to construct the system inertia matrix $\mathbf{H}^*$. The steps needed to accomplish this task are first, compute all the ${}^0v_{ik}$ vectors according to (4)–(7) and (9). Second, compute the ${}^0\mathbf{D}_{ij}$ inertia matrices, according to (3), using the ${}^0v_{ik}$ in the place of the v_{ik}. Third, find the ${}^0\mathbf{F}_i$ matrices according to (11) and, finally, find the inertia matrix $\mathbf{H}^*$.

It is easy to show that the system inertia matrix $\mathbf{H}^*$ is an $N \times N$ positive definite symmetric inertia matrix that depends on $\mathbf{q}$ and the system properties. All elements of $\mathbf{H}^*$ are functions of the manipulator joint angles $q_i (i = 1, \ldots, N)$ only, since the total system inertia with respect to its CM,

${}^0\mathbf{D}$, and ${}^0\mathbf{D}_j$ are functions of only the q_i's and not of the spacecraft attitude. Hence, the system inertia matrix $\mathbf{H}^*$ has the same structural properties as the inertia matrices that correspond to fixed-based manipulators.

Equation (17) shows that T is a function of $(\mathbf{q}, \dot{\mathbf{q}})$, the manipulator joint angles and velocities. The expression for T given by (17) is identically equal to the system Routhian, see [16], and is thus the appropriate function to be used in Lagrange's equations. In the absence of gravity, the potential energy of a rigid system is zero, and hence the system's dynamic equations are given by

$$\frac{d}{dt}\left\{\frac{\partial T}{\partial \dot{\mathbf{q}}}\right\} - \frac{\partial T}{\partial \mathbf{q}} = \mathbf{Q} \tag{19}$$

where $\mathbf{Q}$ is the vector of generalized forces. It is easy to see that in this case $\mathbf{Q}$ is equal to the torque vector $\tau = [\tau_1, \tau_2, \ldots, \tau_N]^T$. Applying (19) to the kinetic energy given by (17) results in a set of N dynamic equations of the form

$$\mathbf{H}^*(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}^*(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \tau \tag{20}$$

where $\mathbf{H}^*(\mathbf{q})$ is the system inertia matrix defined by (18) and $\mathbf{C}^*(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ contains the nonlinear Coriolis and centrifugal terms. Note that in these N equations of motion, the spacecraft attitude or position variables do not enter. This results from the fact that the system kinetic energy does not depend on spacecraft attitude nor on its angular or linear velocity when the initial angular momentum is zero and the system is free of external forces and torques. The spacecraft's contribution to the system's kinetic energy T enters in through the inertia matrices ${}^0\mathbf{D}_{0i}(i = 0, \ldots, N)$, which depend on its mass m_0 and inertia $\boldsymbol{I}_0$.

III. The Nature of Control Algorithms for Free-Floating Systems

It is well known that generally one needs three basic elements (or some combination of them) in order to control a fixed-based manipulator. First, there is an invertible representation of manipulator kinematics, which can be in the form of a closure equation or of a Jacobian. Most control algorithms use the latter. Second, one needs a set of dynamic equations that describe the response of manipulator joint angles to actuator torques or forces. Third, a control algorithm must use sensory information and calculate the required torques or forces to achieve a desired task.

It is clear that if a free-floating space manipulator and a fixed-based manipulator have the same dynamic equation and Jacobian structures, then a control law that can be used for that fixed-based manipulator is suitable for the space manipulator. By structure we mean that the matrices of the dynamic equations and the Jacobian of the two manipulators have the same order and symmetry and depend on the same variables. Furthermore, the inertia matrices of the two systems have the same positive definite character. Of course, the numerical values of the elements of the matrices of the free-floating space system will have different values. For example, the elements of the dynamic matrices $\mathbf{H}^*$ and $\mathbf{C}^*$, will be different from those of the similar matrices of the fixed-based manipulator, $\mathbf{H}$ and $\mathbf{C}$, since $\mathbf{H}^*$ and $\mathbf{C}^*$ depend in part on a spacecraft's mass properties. As a result, the same torque vector τ will produce different joint accelerations in the two systems. However, we are interested here in the structure of the dynamic equations and not in numerical values of the inertia matrix elements. Also, since the applicability of fixed-based controllers does not depend on the existence of gravity, it can be neglected for the purposes of this comparison.

We will compare the structures of the dynamic and kinematic equations of free-floating manipulators to the ones for fixed-based manipulators and show that, based on the above argument, it should be possible to develop a free-floating space manipulator control algorithm based on nearly any algorithm used for fixed-based manipulators, provided that some weak conditions hold. Two types of motion control will be considered. The first—spacecraft-referenced end-point motion control—is the form of control in which the manipulator end point is commanded to move to a location fixed to its own spacecraft, or when a simple joint motion is commanded, such as when the manipulator is to be driven at its stowed position. The second—inertially referenced end-point motion control—is the form of control where the manipulator end-point is commanded to move with respect to inertial space.

A. Spacecraft-Referenced End-Point Motion Control

The comparison between this form of control for a free-floating manipulator and a fixed-based manipulator is rather straightforward. Equation (20) showed that the minimum number of equations describing the dynamics of the $N + 6$ DOF space system is N for an N-DOF manipulator, the same as for a fixed-based N-DOF manipulator. As discussed above, the space system inertia matrix $\mathbf{H}^*$ depends only on the manipulator's joint variables $\mathbf{q}$ and is a symmetric positive definite matrix of size $N \times N$. These are also the properties of the inertia matrix for the fixed-based system. Finally, since $\mathbf{C}^*$ is derived from $\mathbf{H}^*$, it will have the same form as the fixed-based $\mathbf{C}$, which is derived from $\mathbf{H}$. Hence the dynamic equations of both systems have the same structure as defined above.

If the spacecraft becomes very large, m_0 and I_0 approach infinity, and $\mathbf{H}^*$ and $\mathbf{C}^*$ converge to $\mathbf{H}$ and $\mathbf{C}$. This should be expected, because a very large spacecraft will not react to the manipulator's motions and the system will behave essentially as a fixed-based system. Also, the order of the system remains fixed, equal to N, regardless of the size of m_0 and $\mathbf{I}_0$. Finally, since the motion of the space manipulator is controlled with respect to its own base, the Jacobian relating its joint rates to its end-effector velocities is identical to that of the fixed-based manipulator, called $\mathbf{J}$. The above observations hold equally for the simple joint control problem where a $\mathbf{J}$ is not required.

Thus we conclude that nearly any control algorithm that can be used for fixed-based manipulators can also be used for space manipulator systems under spacecraft end-point or joint control. Of course, since the system matrices are different, the control gain matrices may be different in the two cases.

B. Inertially Referenced End-Point Motion Control

The inertial position and orientation of the end effector of a space manipulator is a function of the position and orientation of its spacecraft and of its manipulator joint angles $\mathbf{q}$. It can be shown that the dependency of the end-effector coordinates on the position of the spacecraft can be eliminated by writing the system kinematics with respect to the system CM and by integrating the linear momentum equation. The dependency on the spacecraft orientation cannot be eliminated because the angular momentum given by (16) cannot be integrated analytically to yield a spacecraft's orientation as a function of the manipulator's joint angles [11], [12]. However, these references further show that it is still possible to construct a Jacobian that relates joint motions $\dot{\mathbf{q}}$ to end-effector velocities $\dot{\mathbf{x}}$ in the form:

$$\dot{\mathbf{x}} = \left[\dot{\mathbf{r}}_E, \boldsymbol{\omega}_E\right]^T = \mathbf{J}^*\dot{\mathbf{q}} \tag{21}$$

$$\mathbf{J}^* = \operatorname{diag}\left(\mathbf{T}_0, \mathbf{T}_0\right)^0\mathbf{J}^*(\mathbf{q}) \tag{22}$$

where $\dot{\mathbf{r}}_E$ is the end-point inertial velocity, $\boldsymbol{\omega}_E$ is the end-point inertial angular velocity, and ${}^0\mathbf{J}^*(\mathbf{q})$ is a $6 \times N$ Jacobian that is a function of both the manipulator configuration $\mathbf{q}$ and of the system mass and inertia properties. If $N = 6$, then ${}^0\mathbf{J}^*(\mathbf{q})$ becomes a square matrix. $\mathbf{T}_0$ depends on the spacecraft attitude, which can be measured or estimated as shown in [11]. Clearly, this is a difference between fixed-based and free-floating manipulators. It is unnecessary to use spacecraft attitude where the inertial motion is measured with respect to the spacecraft frame, such as in [7] and [8]. In that case, the Jacobian required in (21) is simply ${}^0\mathbf{J}^*(\mathbf{q})$.

It is well known that the Jacobian $\mathbf{J}$ of a fixed-based manipulator is a $6 \times N$ matrix that depends on $\mathbf{q}$ and the link lengths of the manipulator. $\mathbf{J}^*$ or ${}^0\mathbf{J}^*(\mathbf{q})$ has the same dimensions as $\mathbf{J}$ and also depends on $\mathbf{q}$ as well as on the ${}^0\boldsymbol{v}_{ik}$ vectors, scaled by the ${}^0\mathbf{D}_{ij}(i, j = 0, \ldots, N)$ inertia matrices. This means that the free-floating system differential kinematics, although complicated, have the same structure as the kinematics of the same manipulator with a fixed base, as defined above. Indeed, if a spacecraft's mass and inertia, m_0 and $\mathbf{I}_0$, are large, $\mathbf{T}_0$ approaches a constant matrix and $\operatorname{diag}(\mathbf{T}_0, \mathbf{T}_0)^0\mathbf{J}^*(\mathbf{q})$ results in the normal fixed-based manipulator Jacobian. Mass and inertia dependencies vanish.

However, one important difference is that the workspace of a free-floating system can be divided in two regions, the path-independent workspace (PIW) and the path-dependent workspace (PDW) [11], [12]. If the end-effector path has points in the PDW, the manipulator may become *dynamically singular,* i.e., its Jacobian $\mathbf{J}^*$ or ${}^0\mathbf{J}^*(\mathbf{q})$ becomes singular, although it may not be *kinematically singular,* meaning alignment of axes or points at the workspace boundaries. At a dynamic singularity, the end-effector cannot move in some direction and this represents a physical limitation. A workspace location may induce a singularity or not depending on the *path* or *history* of the motion. This is explained as follows: In general, a workspace location can be reached by an infinite number of sets of configurations $\mathbf{q}$ and of spacecraft orientations. The particular set in which the system will reach some workspace location will depend on the path taken to reach it. This property is due to the nonintegrability of the angular momentum equation. If the configuration in which a workspace location is reached is singular, then this workspace location induces a singularity; otherwise it does not. If the end-point path belongs entirely in the PIW, no dynamic singularities are induced. This leads to the additional condition that a controller must be able to overcome or avoid these singularities.

From the above discussion we conclude that the structure of the kinematics of a free-floating manipulator under inertially referenced end-point control are the same to the fixed-based manipulator case, with the additional conditions that the system's Jacobian depends on the spacecraft's attitude and that dynamic singularities may occur. Furthermore, since the dynamics for this case are identical to those discussed above for spacecraft-referenced end-point control, they have the same structure as a fixed-based system. Thus it follows that nearly any control algorithm that can be used for fixed-based manipulators can also be used for free-floating space manipulators under inertially referenced end-point control, provided, of course, that the appropriate matrices are used. For example, laws like resolved rate, resolved acceleration, impedance control, or computed torque can be used in space if one uses the appropriate Jacobian and inertia matrix. If these matrices are exactly known, then, as in the fixed-based manipulator case, there is *no* need for end-point sensing control. The controller can rely entirely on information provided internally by the system. However, end-point sensing may be needed for space manipulator systems when the uncertainty in the system parameters is so large that the resulting errors are unacceptable. This is also true for fixed-based systems.

C. Differences Between Free-Floating and Fixed-Based Manipulators

So far we have focused our analysis on the similarities between fixed-based and free-floating systems and have shown that it is possible to develop space control algorithms based on nearly any algorithm used for fixed-based manipulators. Now we discuss some of the practical implementation points of free-floating space manipulator control.

1) Terrestrial fixed-based manipulator Jacobians depend on the joint angles $\mathbf{q}$ only. In space, the system Jacobians also depend on spacecraft orientation, see (22). This orientation can be calculated, as shown in [11], [12], or measured on line by additional sensors. No such procedure is needed for fixed-based systems.

2) Singularities are functions of the kinematic structure of the terrestrial fixed-based manipulator only. In space, dynamic singularities exist that depend on the mass and inertia distribution [11], [12]. A point in the space system workspace can be singular or not depending on the path taken to reach it. These singularities represent physical limitations and must be avoided.

3) In general, the knowledge of kinematic parameters, such as link lengths, may be enough for fixed-based manipulator control purposes. Since Jacobians of free-floating space systems depend on a system's dynamic properties, the sys-

tem's kinematic properties are not enough for control purposes. In addition, system dynamics are more complicated and depend on products of inertias that can increase the errors in obtaining inertia matrices. External sensing or on or off-line parameter identification can be very important for space systems.

4) It is not possible to map desired Cartesian workspace points to a unique set of desired joint angles **q** for free-floating manipulators in space, as can be done for fixed-based manipulators, because infinite sets of joint angles correspond, in general, to some workspace point. Which of these sets of joint angles will actually result when the end point reaches the desired workspace point depends on the path taken to reach this point. This characteristic of space systems excludes one early manipulator control algorithm, the "point-to-point" control [17]. A restriction may apply also to adaptive algorithms which assume that dynamic parameters appear in the equations of motion linearly [12].

The above analysis confirms that, with some weak conditions, nearly any control algorithm that can be used in fixed-based systems can also be used in free-floating systems. These conditions include the estimation or measurement of a spacecraft's orientation and the avoidance of dynamic singularities. Knowledge of a system's dynamic properties is helpful, as it is in fixed-based systems. If these properties are not known with sufficient accuracy, end-point sensing can be used. This is demonstrated below by applying a control algorithm developed for fixed-based manipulators to a space system under inertially referenced end-point control.

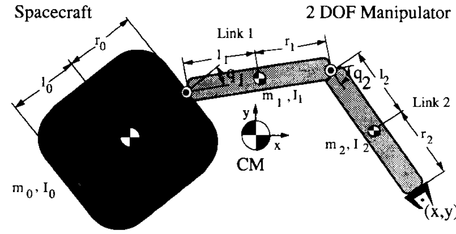


Fig. 4. A planar free-floating manipulator system.

TABLE I
THE SYSTEM PARAMETERS

Body	l_i (m)	r_i (m)	m_i (kg)	I_i (kg m^2)
0	0.5	0.5	40	6.667
1	0.5	0.5	4	0.333
2	0.5	0.5	3	0.250

than their fixed-based counterparts. This Jacobian $\mathbf{J}^*$ should be compared to the fixed-based manipulator Jacobian $\mathbf{J}$ given by

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -(l_1 + r_1)s_1 - (l_2 + r_2)s_{12} & -(l_2 + r_2)s_{12} \\ (l_1 + r_1)c_1 + (l_2 + r_2)c_{12} & (l_2 + r_2)c_{12} \end{bmatrix}. \tag{24}$$

It can be seen that $\mathbf{J}^*$ and $\mathbf{J}$ have the same structure. The system inertia matrix is found according to the analysis presented above (see the Appendix for details). The result is

$$\mathbf{H}^*(\mathbf{q}) = \begin{bmatrix} {}^0d_{11} + 2\,{}^0d_{12} + {}^0d_{22} - \dfrac{(D_1 + D_2)^2}{D} & {}^0d_{12} + {}^0d_{22} - \dfrac{D_2(D_1 + D_2)}{D} \\ {}^0d_{12} + {}^0d_{22} - \dfrac{D_2(D_1 + D_2)}{D} & {}^0d_{22} - \dfrac{D_2^2}{D} \end{bmatrix}. \tag{25}$$

IV. A Planar Example

Here, the relatively simple, fixed-based algorithm, called the transpose Jacobian control by Craig [18], is applied to the simple, planar, free-floating space manipulator system shown in Fig. 4, whose parameters are given in Table I. As shown in [13], the system Jacobian in (22) is given by

$$\mathbf{J}^* = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} {}^0\mathbf{J}^*(\mathbf{q}) \tag{23a}$$

where

$${}^0\mathbf{J}^*(\mathbf{q}) = \frac{1}{D}\begin{bmatrix} -(\beta s_1 + \gamma s_{12})D_0 & \beta s_1 D_2 - \gamma s_{12}(D_0 + D_1) \\ -\alpha(D_1 + D_2) + (\beta c_1 + \gamma c_{12})D_0 & -(\alpha + \beta c_1)D_2 + \gamma c_{12}(D_0 + D_1) \end{bmatrix} \tag{23b}$$

and $s_1 \equiv \sin(q_2)$, $c_{12} \equiv \cos(q_1 + q_2)$, etc. The inertia scalar sums D, D_0, D_1, and D_2 are defined in the Appendix by (A8) and $\alpha \equiv {}^0r_0^*$, $\beta \equiv {}^1r_1^*$, and $\gamma \equiv {}^2c_2^* + r_2$. Lengths α, β, and γ are manipulator link lengths scaled by the mass ratios (m_i/M). Since each D_i and D are functions of **q**, the Jacobian elements are more complicated functions of the **q**

The system inertia matrix $\mathbf{H}^*$, is a 2×2 symmetric matrix whose elements are functions of the joint angles q_1 and q_2. Note that D represents the inertia of the whole system with respect to its CM and thus is always a positive number. The above matrix can be seen to have the same structure as the fixed-based inertia matrix $\mathbf{H}$, whose elements are given by

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} h_{11} & h_{12} \\ h_{12} & h_{22} \end{bmatrix} \tag{26a}$$

$$\begin{aligned} h_{11} &= I_1 + m_1 l_1^2 + m_2(l_1 + r_1)^2 + 2m_2 l_2(l_1 + r_1) \\ &\quad \cdot \cos(q_2) + I_2 + m_2 l_2^2 \\ h_{12} &= h_{21} = m_2 l_2(l_1 + r_1)\cos(q_2) + I_2 + m_2 l_2^2 \\ h_{22} &= I_2 + m_2 l_2^2. \end{aligned} \tag{26b}$$

At the limit, when both m_0 and I_0 approach infinity, it is easy to see that $\beta \rightarrow l_1 + r_1$, $\gamma \rightarrow l_2 + r_2$, i.e., they approach the manipulator link lengths, while $m_0/M \rightarrow 1$, $m_1/M \rightarrow 0$, $m_2/M \rightarrow 0$, $D_0/D \rightarrow 1$, $D_1/D \rightarrow 0$, and $D_2/D \rightarrow 0$; $\mathbf{T}_0$ becomes a constant transformation from the manipulator base frame to the inertial frame, usually the unit matrix; and finally, $\mathbf{J}^* \rightarrow \mathbf{J}$, the fixed-based Jacobian, and $\mathbf{H}^* \rightarrow \mathbf{H}$, the fixed-based inertia matrix, as given by (24) and (26), respectively.

One can select any control algorithm that can be used for fixed-based manipulators, using the two matrices $\mathbf{H}^*$ and $\mathbf{J}^*$, depending on the manipulator task. Here, the transposed Jacobian control is used, augmented by a velocity feedback term for increased stability margins. The end-point position and velocity $\mathbf{x} = [x, y]^T$ and $\dot{\mathbf{x}} = [\dot{x}, \dot{y}]^T$ can be calculated or measured directly. Assuming we measure $\mathbf{x}$ and $\dot{\mathbf{x}}$, the control law is

$$\tau = \mathbf{J}^{*T}\{\mathbf{K}_p(\mathbf{x}_{\text{des}} - \mathbf{x}) - \mathbf{K}_d\dot{\mathbf{x}}\} \tag{27}$$

where $\mathbf{x}_{\text{des}}$ is the inertial desired point location. $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite diagonal matrices. Note that this algorithm drives the end point to the desired location but does not specify a path. If the control gains are large enough, then the motion of the end-point will be a straight line. The torque vector is nonzero till the $(\mathbf{x}_{\text{des}} - \mathbf{x})$ and $\dot{\mathbf{x}}$ are zero, or until the vector in the brackets in (27) is in the null space of $\mathbf{J}^{*T}$.

Fig. 5 depicts the reachable workspace, the PIW, and the PDW for this example. The boundaries of all three are circles with their center at the system CM. The PDW is found by noting that the distance of the end effector from the system CM is a function of the system configuration q only. All singular configurations q, obtainable by solving the $\det(\mathbf{J}^*) = 0$, can be mapped to circles with their centers at the system CM. The union of all these circles gives the PDW. Subtracting the PDW from the reachable workspace results in the PIW. For a more general exposition of this subject, the reader is referred to [11] and [12].

First, the end-point path will be restricted to the PIW part of the workspace, and hence dynamic singularities will be avoided. Fig. 6 shows the motion of the end-point from the initial location (1, 0) to the final (0.8, 0.8). This path is shown in Fig. 5 as path A. The control gain matrices are $\mathbf{K}_p = \text{diag}(5, 5)$ and $\mathbf{K}_d = \text{diag}(15, 15)$. The end-point path, shown with a heavy line, is almost a straight line and converges to the desired location. Shown also is the end-point path that results when the control law, given by (27), uses the fixed-based Jacobian given by (24). In this case, the end-point diverges from the straight line because it does not resolve the error term correctly. Depending on the situation, the use of the fixed-based Jacobian can create stability problems [9]. Fig. 7 shows the spacecraft attitude θ and the joint angles as a function of time, during the end-point motion depicted in Fig. 6, when $\mathbf{J}^*$ is used. Note that although the spacecraft attitude changes for about 35°, the manipulator end-point converges to the desired inertial location as it would do if its base were fixed.

Next, the end-effector is commanded to follow a straight line path from (2, 0) to (1.5, 1.5), shown as path B in Fig. 5.

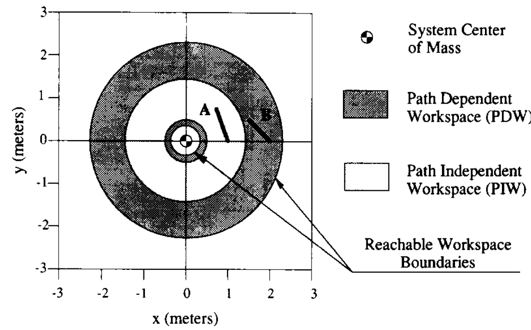


Fig. 5. The reachable, PIW and PDW, for the system shown in Fig. 4.

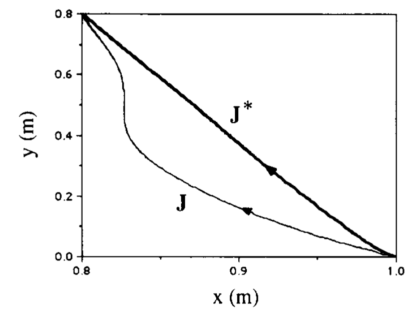


Fig. 6. End-point paths in the PDW using $\mathbf{J}^*$ and $\mathbf{J}$.

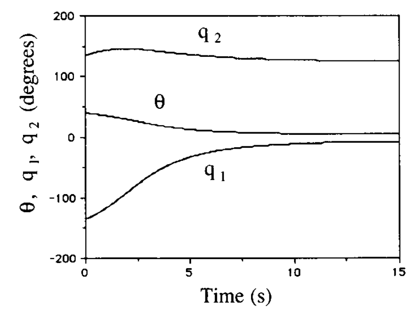


Fig. 7. The spacecraft orientation θ and the joint angles q_1 and q_2 along the path shown in Fig. 6.

Fig. 8 shows the actual end-effector path. Note that initially the end-effector moves along a straight line until it reaches point B, where a dynamic singularity occurs for the first time. At this point the end-effector diverges from the desired path and the joint angles start oscillating around singular configurations, as shown in Fig. 9. Finally, the end-effector converges to point C, which is an equilibrium point. Note that if an inverse Jacobian algorithm had been used, it would have failed numerically at point B.

In this example, we have assumed that the end-point position and velocity are measured. However, if we know the system parameters exactly, they can both be calculated. To do this, one has to measure the joint angles $\mathbf{q}$ and calculate the spacecraft attitude integrating numerically (16). Then the manipulator would rely entirely on internally provided information and would not need end-point sensing. However, it may be very difficult to obtain the correct values for all the

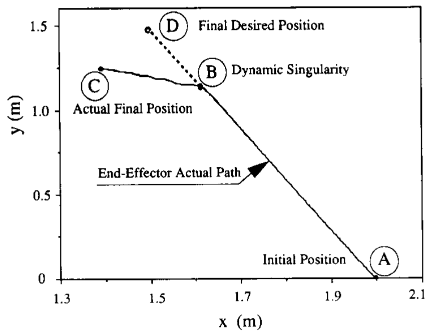


Fig. 8. An end-point path in the PDW where dynamic singularities occur.

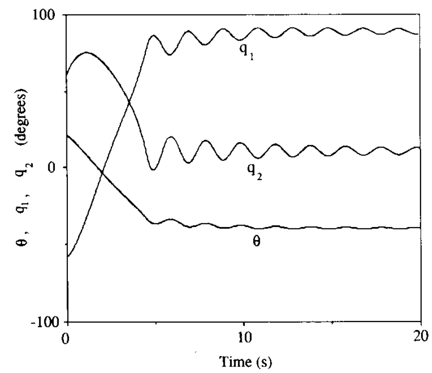


Fig. 9. The spacecraft attitute θ and the manipulator joint angles q_1 and q_2 along the path shown in Fig. 8.

system dynamic parameters and any errors will be magnified due to the products involved in the computations. Thus, it is not the physics of the problem that forces the use of end-point control, but rather the incomplete information about the dynamics of the plant. This fact is also true for fixed-based manipulators, although in this case the uncertainty is less severe because the system matrices depend on fewer parameters.

Following the above procedure, any control algorithm that employs the system $\mathbf{H}^*$ and $\mathbf{J}^*$ can be designed. However, control methods that depend on the cancellation of terms, like the computed torque methods, require the exact system inertia matrix $\mathbf{H}^*$, and thus emphasis must be placed in its computation.

V. Conclusions

A fundamental study has been performed of the characteristics of control algorithms, which may be applied to the motion control of free-floating space manipulators. The results obtained show that nearly any control algorithm that can be applied to conventional terrestrial fixed-based manipulators, with a few additional conditions, can be directly applied to free-floating space manipulators. We hope that the results encourage the development of more effective control algorithms for free-floating space manipulator systems.

Appendix

For the planar two-link system, shown in Fig. 4, the two coordinates of the end-effector x and y are assumed to be controlled by the two manipulator joint angles q_1 and q_2. The end-effector orientation is not controlled for this 2-DOF system ($N = 2$), hence (21) for this system is simply

$$\dot{\mathbf{x}} = \dot{\mathbf{r}}_E = \frac{d}{dt}[x, y]^T = \mathbf{J}^*\dot{\mathbf{q}} \tag{A1}$$

where $\mathbf{J}^*$ is given by (23).

In the following the construction of the system interia matrix will be demonstrated. First, express all v_{ik} in (7) is the frame of the ith body according to (4), (5), and (6). For the sake of simplicity assume that all r_i and $\mathbf{l}_i$ are parallel to the x axis of the ith frame. Hence, only the x component of the ${}^iv_{ik}$ is nonzero and is given by

$$\begin{aligned}
{}^0r_0^* &= \frac{1}{M} r_0 m_0 \\
{}^0c_0^* &= -\frac{1}{M} r_0(m_1 + m_2) \\
{}^0l_0^* &= -\frac{1}{M} r_0(m_1 + m_2) - l_0 \\
{}^1r_1^* &= \frac{1}{M}\{r_1(m_0 + m_1) + l_1 m_0\} \\
{}^1c_1^* &= \frac{1}{M}(l_1 m_0 - r_1 m_2) \\
{}^1l_1^* &= -\frac{1}{M}\{l_1(m_1 + m_2) + r_1 m_2\} \\
{}^2r_2^* &= \frac{1}{M} l_2(m_0 + m_1) + r_2 \\
{}^2c_2^* &= \frac{1}{M} l_2(m_0 + m_1) \\
{}^2l_2^* &= -\frac{1}{M} l_2 m_2
\end{aligned} \tag{A2}$$

where the total mass of the system M is given by

$$M = m_0 + m_1 + m_2. \tag{A3}$$

For this example, the transformation matrix from the spacecraft frame to the inertial frame $\mathbf{T}_0$ is given by

$$\mathbf{T}_0(\theta) = \mathrm{Rot}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{A4}$$

where θ denotes the spacecraft attitude, as shown in Fig. 4. Only the planar subpart of the transformation matrices is used for simplicity. The transformation matrices ${}^0\mathbf{T}_i$ are found using (9):

$${}^0\mathbf{T}_1 = \mathrm{Rot}(q_1) \qquad {}^0\mathbf{T}_2 = \mathrm{Rot}(q_1)\,\mathrm{Rot}(q_2). \tag{A5}$$

For the planar case, the inertia matrices ${}^0\mathbf{D}_{ij}$, which correspond to (3) and (12), reduce to the scalars ${}^0d_{ij}$, given by

$$
\begin{aligned}
{}^0d_{00} &= I_0 + \frac{m_0(m_1+m_2)}{M} r_0^2 \\
{}^0d_{10} &= \frac{m_0 r_0}{M}\{l_1(m_1+m_2) + r_1 m_2\}\cos(q_1) = {}^0d_{01} \\
{}^0d_{20} &= \frac{m_0 m_2}{M} r_0 l_2 \cos(q_1+q_2) = {}^0d_{02} \\
{}^0d_{11} &= I_1 + \frac{m_0 m_1}{M} l_1^2 + \frac{m_1 m_2}{M} r_1^2 + \frac{m_0 m_2}{M}(l_1+r_1)^2 \\
{}^0d_{21} &= \left\{\frac{m_1 m_2}{M} r_1 l_2 + \frac{m_0 m_2}{M} l_2(l_1+r_1)\right\}\cos(q_2) = {}^0d_{12} \\
{}^0d_{22} &= I_2 + \frac{m_2(m_0+m_1)}{M} l_2^2. \qquad \text{(A6)}
\end{aligned}
$$

Both ${}^i\mathbf{u}_i (i = 1, 2)$ in (11) are equal to $[0\ 0\ 1]^T$; the ${}^0\mathbf{F}_i$ matrices reduce to

$$
\begin{aligned}
{}^0\mathbf{F}_1 &= [1 \quad 0] \\
{}^0\mathbf{F}_2 &= [1 \quad 1]. \qquad \text{(A7)}
\end{aligned}
$$

For simplicity, drop the left superscripts from ${}^0\mathbf{D}_j$ and set

$$
\begin{aligned}
D_j &\equiv \sum_{i=0}^{2} {}^0d_{ij}, \qquad j = 0, 1, 2 \\
D &\equiv D_0 + D_1 + D_2. \qquad \text{(A8)}
\end{aligned}
$$

Using (13) and (18), the inertia matrix $\mathbf{H}^*$ is written as

$$\mathbf{H}^*(\mathbf{q}) = \sum_{i=1}^{2}\sum_{j=1}^{2} {}^0\mathbf{F}_i^T\left({}^0d_{ij} - \frac{D_i D_j}{D}\right){}^0\mathbf{F}_j. \qquad \text{(A9)}$$

Substituting into (A6), (A7), and (A8) yields the explicit form of $\mathbf{H}^*$ given as (25).

References

[1] S. Dubowsky, E. Vance, and M. Torres, "The control of space manipulators subject to spacecraft attitude control saturation limits," in *Proc. NASA Conf. Space Telerobotics* (JPL, Pasadena, CA, Jan. 31-Feb. 2, 1989).

[2] M. Torres and S. Dubowsky, "Path planning for space manipulators to minimize spacecraft attitude disturbance," in *Proc. 1991 IEEE Int. Conf. Robotics Automat.* (Sacramento, CA), Apr. 1991, vol. 3, pp. 2522-2528.

[3] R. Longman, R. Lindberg, and M. Zedd, "Satellite-mounted robot manipulators—new kinematics and reaction moment compensation," *Int. J. Robotics Res.*, vol. 6, no. 3, pp. 87-103, Fall 1987.

[4] 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), Mar. 1987.

[5] —, "On the dynamics of space manipulators using the virtual manipulator, with applications to path planning," *J. Astr. Sci.*, vol. 38, no. 4, pp. 441-472, Oct.-Dec. 1990.

[6] —, "The kinematics and dynamics of space manipulators: The virtual manipulator approach," *Int. J. Robotics Res.*, vol. 9, no. 4, pp. 3-21, Aug. 1990.

[7] H. Alexander and R. Cannon, "Experiments on the control of a satellite manipulator," in *Proc. 1987 Amer. Contr. Conf.* (Seattle, WA), June 1987.

[8] Y. Umetani and K. Yoshida, "Experimental study on two dimensional free-flying robot satellite model," in *Proc. NASA Conf. Space Telerobotics* (Pasadena, CA), Jan. 1989.

[9] Y. Masutani, F. Miyazaki, and S. Arimoto, "Sensory feedback control for space manipulators," in *Proc. IEEE Int. Conf. Robotics Automat.* (Scottsdale, AZ), May 1989.

[10] Y. Nakamura and R. Mukherjee, "Nonholonomic path planning of space robots," in *Proc. IEEE Int. Conf. Robotics Automat.* (Scottsdale, AZ), May 1989.

[11] E. Papadopoulos and D. Dubowsky, "On the dynamic singularities in the control of free-floating space manipulators," in *Proc. of ASME Winter Ann. M.* (San Francisco), Dec. 1989, ASME DSC-vol. 15.

[12] E. Papadopoulos, "On the dynamics and control fo space manipulators," Ph.D. dissertation, Dept. of Mechanical Engineering, Massachusetts Institute of Technology Cambridge, MA, Oct. 1990.

[13] W. Hooker and G. Margulies, "The dynamical attitude equations for an n-body satellite," *J. Ast. Sci.*, vol XII, no. 4, Winter 1965.

[14] J. Wittenburg, *Dynamics of Rigid Bodies.* Stuttgart: B. G. Teubner, 1977.

[15] P. Hughes, *Spacecraft Attitude Dynamics.* New York: Wiley, 1986.

[16] H. Goldstein, *Classical Mechanics.* Reading, MA: Addison-Wesley, 1980.

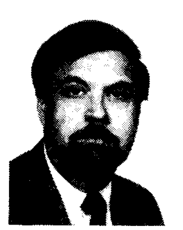
[17] A., Critchlow, *Introduction to Robotics.* New York: Macmillan, 1985.

[18] J. Craig, *Introduction to Robotics, Mechanics and Control.* Reading, MA: Addison-Wesley, 1986.

Evangelos Papadopoulos (S'82-M'91) received a Diploma in mechanical engineering from the National Technical University of Athens, Athens, Greece, in 1981, and the M.S. and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, in 1983 and 1990, respectively.

He is an Assistant Professor of Mechanical Engineering at McGill University and a member of the McGill Research Centre for Intelligent Machines. In 1991, he was a Lecturer with the Department of Mechanical Engineering at MIT. From 1985 to 1987 he was a researcher at the Greek Navy Research and Technology Office. His main areas of interest are robotics, space robotics, modeling, and control of dynamic systems, and control theory.

Dr. Papadopoulos is a member of the ASME and Sigma Xi.



Steven Dubowsky (M'72) received the B.S. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1963, and the M.S. and Sc.D. degrees from Columbia University in 1964 and 1971, respectively.

He is currently a Professor of Mechanical Engineering at the Massachusetts Institute of Technology, Cambridge. He has been a Professor of Engineering and Applied Science at the University of California, Los Angeles; a Visiting Professor at Cambridge University, Cambridge, England; and a Visiting Professor at the California Institute of Technology, Pasadena, CA. From 1963 to 1971, he was employed by the Perkin-Elmer Corporation, the General Dynamics Corporation, and the American Electric Power Service Corporation. His research has included the development of modeling techniques for manipulator flexibility and the development of optimal and self-learning adaptive control procedures for rigid and flexible robotic manipulators. He has authored or coauthored nearly 100 papers in the area of the dynamics, control, and design of high-performance mechanical and electromechanical systems.

Prof. Dubowsky is a registered Professional Engineer in the State of California and has served as an advisor to the National Science Foundation, the National Academy of Science/Engineering, the Department of Energy, and the U.S. Army. He has been elected a Fellow of the ASME and is a member of Sigma Xi and Tau Beta Pi.