We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



2

Operational Space Dynamics of a Space Robot and Computational Efficient Algorithm

Satoko Abiko and Gerd Hirzinger
Institute of Robotics and Mechatronics,German Aerospace Center (DLR)
82334, Weßling, Germany

1. Introduction

On-orbit servicing space robot is one of the challenging applications in space robotic field. Main task of the on-orbit space robot involves the tracking, the grasping and the positioning of a target. The dynamics in operational space is useful to achieve such tasks in Cartesian space. The operational space dynamics is a formulation of the dynamics of a complex branching redundant mechanism in task or operational points. Khatib proposed the formulation of a serial robot manipulator system on ground in (Khatib, 1987). Russakow *et. al.* modified it for a branching manipulator system in (Russakow et al., 1995). Chang and Khatib introduced efficient algorithms for this formulation, especially for operational space inertia matrix in (Chang & Khatib, 1999; 2000).
The operational space dynamics of the space robot is more complex than that of the ground-based manipulator system since the base-satellite is inertially free. However, by virtue of no fixed-base, the space robot is invertible in its modeling and arbitrary operational points to control can be chosen in a computational efficient manner. By making use of this unique characteristic, we firstly propose an algorithm of the dynamics of a single operational point in the space robot system. Then, by using the concept of the articulated-body algorithm(Featherstone, 1987), we propose a recursive computation of the dynamics of multi-operational points in the space robot. The numerical simulations are carried out using a two-arm space robot shown in Fig. 1.
This chapter is organized as follows. Section 2 describes basic dynamic equations of free-flying and free-floating space robots. Section 3 derives the operational space formulation of both types of space robots. Section 4 briefly introduces spatial notation to represent complex robot kinematics and dynamics, which is used for the derivation of the proposed algorithms. Section 5 describes recursive algorithms of the generalized Jacobian matrix(Xu & Kanade, 1993), that is a Jacobian matrix including dynamical coupling between the base body and the robot arm. Section 6 proposes computational efficient algorithms of the operational space dynamics. Section 7 shows the simulation example of the proposed algorithms. Section 8 summarizes the conclusions.

2. Basic Equations

This section presents basic dynamic equations of the space robot. The main symbols used in this section are defined in table 1.

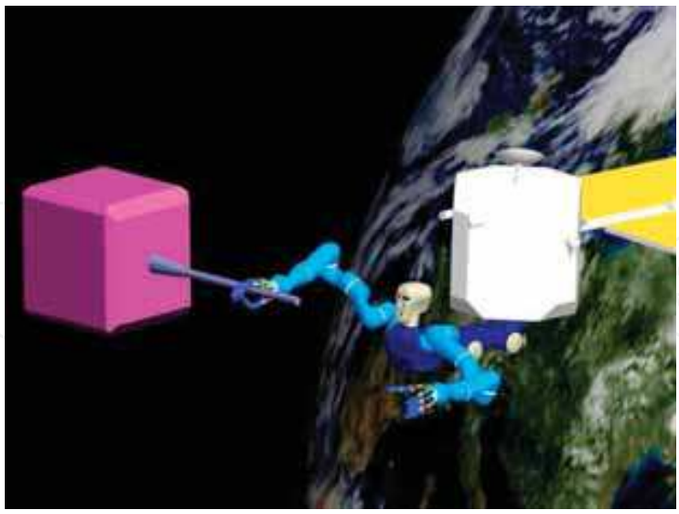


Fig. 1. Chaser-robot and target scenario

2.1 Linear and Angular Momentum Equations

The motion of the space robot is generally governed by the principle of the conservation of momentum. When the spatial velocity of the base body, $\dot{\boldsymbol{x}}_b = (\boldsymbol{v}_b^T, \boldsymbol{\omega}_b^T)^T \in R^{6\times 1}$, and the motion rate of the joints, $\dot{\boldsymbol{\phi}} \in R^{n\times 1}$, are considered as the generalized coordinates, total linear and angular momentum, $\mathcal{M}_0 \in R^{6\times 1}$, are expressed as follows:

$$\mathcal{M}_0 = \boldsymbol{H}_b \dot{\boldsymbol{x}}_b + \boldsymbol{H}_{bm} \dot{\boldsymbol{\phi}} \, . \tag{1}$$

Note that $\mathcal{M}_0$ represents the total momentum around the center of mass of the base body.
In the absence of external forces, the total momentum is conserved. From eq. (1), the motion of the base body is expressed by $\dot{\boldsymbol{\phi}}$ and $\mathcal{M}_0$ as:

$$\dot{\boldsymbol{x}}_b = \boldsymbol{J}_b^* \dot{\boldsymbol{\phi}} + \boldsymbol{H}_b^{-1} \mathcal{M}_0 \in R^{6\times 1}, \tag{2}$$

where

$$\boldsymbol{J}_b^* = -\boldsymbol{H}_b^{-1} \boldsymbol{H}_{bm} \in R^{6\times n} \tag{3}$$

represents the generalized Jacobian matrix of the base body (Yokokohji et al., 1993). By introducing the kinematic mapping of the i-th operational point, $\dot{\boldsymbol{x}}_{e_i} = \boldsymbol{J}_{b_i} \dot{\boldsymbol{x}}_b + \boldsymbol{J}_{m_i} \dot{\boldsymbol{\phi}}$, eq. (1) provides the velocity of the operational point as follows:

$$\dot{\boldsymbol{x}}_{e_i} = \boldsymbol{J}_{m_i}^* \dot{\boldsymbol{\phi}} + \boldsymbol{J}_{b_i} \boldsymbol{H}_b^{-1} \mathcal{M}_0 \in R^{6\times 1}, \tag{4}$$

where

$$\boldsymbol{J}_{m_i}^* = \boldsymbol{J}_{m_i} - \boldsymbol{J}_{b_i} \boldsymbol{H}_b^{-1} \boldsymbol{H}_{bm} \in R^{6\times n} \tag{5}$$

is called the generalized Jacobian matrix of the operational point (Umetani & Yoshida, 1989). The above generalized Jacobian matrix, (5), is for the case that a single point is selected as an operational point. This matrix is simply extended to the case of the multi-operational points

n		:	number of joints
p		:	number of operational points
$\dot{\boldsymbol{x}}_b$	$\in R^{6\times 1}$	:	linear and angular velocity of the base.
$\dot{\boldsymbol{\phi}}$	$\in R^{n\times 1}$	:	motion rate of the arms.
$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{x}}_{e_1} \\ \vdots \\ \dot{\boldsymbol{x}}_{e_p} \end{bmatrix}$	$\in R^{6p\times 1}$	:	linear and angular velocity of the operational points ($i = 1 \cdots p$).
$\boldsymbol{H}_b$	$\in R^{6\times 6}$	:	inertia matrix of the base.
$\boldsymbol{H}_m$	$\in R^{n\times n}$	:	inertia matrix of the arms.
$\boldsymbol{H}_{bm}$	$\in R^{6\times n}$	:	coupling inertia matrix between the base and the arms.
$\boldsymbol{c}_b$	$\in R^{6\times 1}$	:	non-linear velocity dependent term of the base.
$\boldsymbol{c}_m$	$\in R^{n\times 1}$	:	non-linear velocity dependent term of the arms.
$\mathcal{F}_b$	$\in R^{6\times 1}$	:	force and moment exerted on the base.
$\mathcal{F}_e$	$\in R^{6p\times 1}$	:	force and moment exerted on the operational points.
$\boldsymbol{\tau}$	$\in R^{n\times 1}$	:	torque on joints.
$\boldsymbol{J}_{b_i}$	$\in R^{6\times 6}$	:	Jacobian matrix of the base in terms of the i-th operational point.
$\boldsymbol{J}_{m_i}$	$\in R^{6\times n}$	:	Jacobian matrix of the arms in terms of the i-th operational point.
$\boldsymbol{J}_b^*$	$\in R^{6\times 6}$	:	Generalized Jacobian matrix of the base body.
$\boldsymbol{J}_{m_i}^*$	$\in R^{6\times n}$	:	Generalized Jacobian matrix of the arms in terms of the i-th operational point.

Table 1. Main Notation

by augmenting the Jacobian matrix of each operational point. In section 5, we derive recursive calculations of the matrices, (3) and (5).

2.2 Equations of Motion

The general dynamic equation of the space robot is described by the following expression (Xu & Kanade, 1993):

$$\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\phi}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}_b^T \\ \boldsymbol{J}_m^T \end{bmatrix} \mathcal{F}_e. \tag{6}$$

where $\dot{\boldsymbol{x}}_b = (\boldsymbol{v}_b^T, \boldsymbol{\omega}_b^T)^T \in R^{6\times 1}$, and the motion rate of the joints, $\dot{\boldsymbol{\phi}} \in R^{n\times 1}$ are considered as the generalized coordinates. When $\mathcal{F}_b$ is actively generated (*e.g.* jet thrusters or reaction wheels etc.), the system is called a *free-flying* robot. If no active actuators are applied on the

base, the system is termed a *free-floating* robot. The integral of the upper part of eq. (6) describes the total linear and angular momentum around the center of mass of the base body and corresponds to the equation (1).

2.3 Dynamics of a Free-Floating Space Robot

The dynamic equation of the free-floating space robot can be furthermore reduced a form expressed with only joint acceleration, $\ddot{\boldsymbol{\phi}}$, by eliminating the base body acceleration, $\ddot{\boldsymbol{x}}_b$, from eq. (6):

$$\boldsymbol{H}_m^* \ddot{\boldsymbol{\phi}} + \boldsymbol{c}_m^* = \boldsymbol{\tau} + \boldsymbol{J}_b^{*T} \mathcal{F}_b + \boldsymbol{J}_m^{*T} \mathcal{F}_e \tag{7}$$

where $\boldsymbol{H}_m^* = \boldsymbol{H}_m - \boldsymbol{H}_{bm}^T \boldsymbol{H}_b^{-1} \boldsymbol{H}_{bm} \in R^{n \times n}$ and $\boldsymbol{c}_m^* = \boldsymbol{c}_m - \boldsymbol{H}_{bm}^T \boldsymbol{H}_b^{-1} \boldsymbol{c}_b \in R^{n \times 1}$ represent generalized inertia matrix and generalized non-linear velocity dependent term, respectively.

3. Operational Space Formulation

The operational space dynamics is useful to control the system in the operational space, which represents the dynamics projected from the joint space to the operational space. The two types of space robot dynamics are described in the following subsections. One is for the free-flying space robot and the other is for the free-floating space robot. This section derives the equations of motion for the space robots consisting of n-links with p operational points.

3.1 Free-Flying Space Robot

The operational space dynamics of the free-flying space robot is described in the following form:

$$\boldsymbol{\Gamma}_e \ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e = \mathcal{F}_e^{in} + \mathcal{F}_e \,, \tag{8}$$

where

$$\begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{J}_e^T \mathcal{F}_e^{in} \,.$$

$\mathcal{F}_e \in R^{6p \times 1}$ consists of the 6×1 external force of each of p operational points. $\boldsymbol{J}_e \in R^{6p \times (6+(n-1))}$ consists of Jacobian matrix of each operational point.

$$\mathcal{F}_e = \begin{bmatrix} \mathcal{F}_{e_1} \\ \vdots \\ \mathcal{F}_{e_p} \end{bmatrix} \text{ and } \boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{J}_{b_1}, & \boldsymbol{J}_{m_1} \\ \vdots & \vdots \\ \boldsymbol{J}_{b_p}, & \boldsymbol{J}_{m_p} \end{bmatrix}.$$

The operational space inertia matrix of the free-flying space robot, $\boldsymbol{\Gamma}_e$, is an $6p \times 6p$ symmetric positive definite matrix. Its inverse matrix can be expressed as :

$$\boldsymbol{\Gamma}_e^{-1} = \boldsymbol{J}_e \boldsymbol{H}^{-1} \boldsymbol{J}_e^T \,,\; \boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \,. \tag{9}$$

The operational space centrifugal and Coriolis forces, $\boldsymbol{\mu}_e$, is expressed as :

$$\boldsymbol{\mu}_e = \boldsymbol{J}_e^{T+} \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} - \boldsymbol{\Gamma}_e \frac{d}{dt} \boldsymbol{J}_e \begin{bmatrix} \dot{\boldsymbol{x}}_b \\ \dot{\boldsymbol{\phi}} \end{bmatrix} \,, \tag{10}$$

where $\boldsymbol{J}_e^+$ is the dynamically consistent generalized inverse of the Jacobian matrix $\boldsymbol{J}_e$ for the free-flying space robot to minimize the instantaneous kinetic energy of the space robot:

$$\boldsymbol{J}_e^+ = \boldsymbol{H}^{-1} \boldsymbol{J}_e^T \boldsymbol{\Gamma}_e. \tag{11}$$

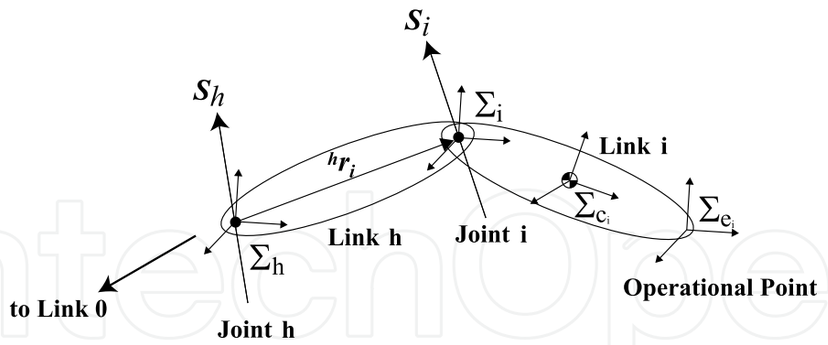


Fig. 2. Notation Representation

3.2 Free-Floating Space Robot

In the free-floating space robot, no active forces exist on the base (*e.g.* $\mathcal{F}_b = \mathbf{0}$). Then, the system can be described as the reduced form in the joint space by using eq. (7). Its operational space dynamics can be derived from eqs. (5) and (7):

$$\mathbf{\Gamma}_e \ddot{\boldsymbol{x}}_e + \mathbf{\Gamma}_e \boldsymbol{\mu} = \mathbf{\Gamma}_e \mathbf{\Lambda}^{-1} \boldsymbol{J}_m^{*T+} \boldsymbol{\tau} + \mathcal{F}_e \,, \tag{12}$$

where

$$\mathbf{\Gamma}_e^{-1} = \mathbf{\Lambda}^{-1} + \mathbf{\Lambda}_b^{-1} \in R^{6p \times 6p} \,,$$
$$\mathbf{\Lambda}^{-1} = \boldsymbol{J}_m^* \boldsymbol{H}_m^{*-1} \boldsymbol{J}_m^{*T} \,,\ \mathbf{\Lambda}_b^{-1} = \boldsymbol{J}_b \boldsymbol{H}_b^{-1} \boldsymbol{J}_b^T \,.$$

The matrix, $(\mathbf{\Lambda}^{-1} + \mathbf{\Lambda}_b^{-1})$, corresponds to the inertia matrix described in eq. (9). The vector, $\boldsymbol{\mu}$, expresses the bias acceleration vector resulting from the Coriolis and centrifugal forces as:

$$\boldsymbol{\mu} = \mathbf{\Lambda}^{-1} \boldsymbol{J}_m^{*+} \boldsymbol{c}_m^* - \dot{\boldsymbol{J}}_m^* \dot{\boldsymbol{\phi}} - \frac{d}{dt} (\boldsymbol{J}_b \boldsymbol{H}_b^{-1}) \mathcal{M}_0 \in R^{6p \times 1}.$$

where

$$\boldsymbol{J}_m^{*+} = \boldsymbol{H}_m^{*-1} \boldsymbol{J}_m^{*T} \mathbf{\Lambda} \tag{13}$$

represents the dynamically consistent generalized inverse of the Jacobian matrix $\boldsymbol{J}_m^*$ for the free-floating space robot. Compared with eq. (8), the relationship $\mathbf{\Gamma}_e \boldsymbol{\mu} = \boldsymbol{\mu}_e$ is obtained. Note that each dynamic equation described in this section is expressed in the inertial frame. Section 6 describes the efficient algorithms for the operational space dynamics represented in this section.

4. Spatial Notation

The *Spatial Notation* is well-known and intuitive notation in modeling kinematics and dynamics of articulated robot systems, introduced by Featherstone (Featherstone (1987); Chang & Khatib (1999)). This section concisely reviews the basic spatial notation. The main symbols used in the spatial notation are defined in Table 2. The symbols are expressed in the frame fixed at each link. (See. Fig. 2).

$\mathbf{v}_i$	$\in R^{6\times1}$	:	spatial velocity of link i.
$\mathbf{a}_i$	$\in R^{6\times1}$	:	spatial acceleration of link i.
$\boldsymbol{f}_i$	$\in R^{6\times1}$	:	composite force of link i.
$\boldsymbol{f}_i^*$	$\in R^{6\times1}$	:	spatial force of link i.
$\mathcal{L}_i$	$\in R^{6\times1}$	:	composite momentum of link i.
$\mathcal{L}_i^*$	$\in R^{6\times1}$	:	spatial momentum of link i.
$\boldsymbol{I}_i^C$	$\in R^{6\times1}$	:	composite inertia matrix of link i.

Table 2. Main Symbols in Spatial Notation

4.1 Spatial Notation

In the spatial notation, linear and angular components are dealt with in a unified framework and results in a concise form (*e.g.* 6×1 vector or 6×6 matrix). In this expression, a spatial velocity, $\mathbf{v}_i$, and a spatial force, $\boldsymbol{f}_i$, of link i are defined as :

$$\mathbf{v}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \text{ and } \boldsymbol{f}_i = \begin{bmatrix} \mathcal{F}_i \\ \mathcal{T}_i \end{bmatrix},$$

where $\boldsymbol{v}_i$, $\boldsymbol{\omega}_i$, $\mathcal{F}_i$, and $\mathcal{T}_i$ represent the 3×1 linear and angular velocity, the force and moment in terms of link i in frame i, respectively.
The simple joint model, $\boldsymbol{S}_i \in R^{6\times1}$, for prismatic and rotational joint is defined, so that 1 is assigned along the prismatic or rotational axis : *e.g.*

$$\boldsymbol{S}_i = [\,0\,0\,1\,0\,0\,0\,]^T \text{ for prismatic joint at z axis}$$

and

$$\boldsymbol{S}_i = [\,0\,0\,0\,0\,0\,1\,]^T \text{ for rotational joint at z axis.}$$

More complex multi-degrees-of-freedom joint is introduced in (Featherstone, 1987; Lilly, 1992).
The spatial inertia matrix of link i in frame c_i, $\boldsymbol{I}_{c_i}$, is a symmetric positive definite matrix as:

$$\boldsymbol{I}_{c_i} = \begin{bmatrix} m_i\boldsymbol{E}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{c_i} \end{bmatrix} \in R^{6\times6},$$

where m_i is the mass of link i, and $\mathbf{I}_{c_i} \in R^{3\times3}$ is the inertia matrix around the center of mass of the link i in frame c_i. $\boldsymbol{E}_3$ stands for the 3×3 identity matrix.
The 6×6 spatial transformation matrix, ${}^h_i\boldsymbol{X}$, transforms a spatial quantity from frame i to frame h as:

$${}^h_i\boldsymbol{X} = \begin{bmatrix} {}^h_i\boldsymbol{R} & \mathbf{0} \\ \widetilde{{}^h\boldsymbol{r}_i}\,{}^h_i\boldsymbol{R} & {}^h_i\boldsymbol{R} \end{bmatrix} \in R^{6\times6}. \tag{14}$$

where ${}^h_i\boldsymbol{R} \in R^{3\times3}$ is a rotation matrix and ${}^h\boldsymbol{r}_i \in R^{3\times1}$ is a position vector from the origin of frame h to that of frame i expressed in frame h. $\widetilde{\{\cdot\}}$ denotes the skew-symmetric matrix. Unlike

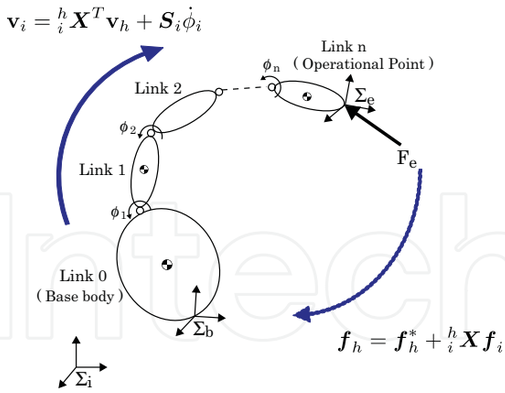


Fig. 3. Forward Chain Approach

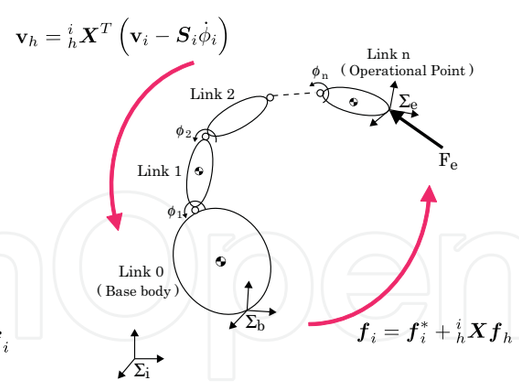


Fig. 4. Inverted Chain Approach

the conventional 3×3 rotational matrix, the spatial transformation matrix is not orthogonal. The transformation matrix from frame h to frame i, ${}^i_h\boldsymbol{X}$, is expressed as:

$$ {}^i_h\boldsymbol{X} = \begin{bmatrix} {}^h_i\boldsymbol{R}^T & \mathbf{0} \\ \left(\widetilde{{}^h\boldsymbol{r}_i}\,{}^h_i\boldsymbol{R}\right)^T & {}^h_i\boldsymbol{R}^T \end{bmatrix} \in R^{6\times 6}. \tag{15} $$

4.2 Spatial Quantity

The most of the spatial quantities including the spatial velocity, acceleration, and the spatial force are iteratively calculated. To carry out the iterative calculation, two approaches can be found, which are dependent on the direction of the iteration. In the first approach, the kinematic quantities such as the velocity and the acceleration are computed by outward recursion from the link 0 toward the operational points as shown in Fig. 3 and the spatial force is obtained by inward recursion from the operational points toward the link 0. This approach is termed *forward chain approach*. In the second approach, the direction of the recursive calculation is opposed to the first one, so that the kinematic quantities are computed by inward recursion and the spatial force is obtained by outward recursion as shown in Fig. 4. This approach is termed *inverted chain approach*.

4.2.1 Forward Chain Approach

In the forward chain approach, the spatial velocity of link i is computed by the spatial velocity of its parent link and its joint velocity as:

$$ \mathbf{v}_i = {}^h_i\boldsymbol{X}^T\mathbf{v}_h + \boldsymbol{S}_i\dot{\phi}_i, \quad (\mathbf{v}_i = \mathbf{v}_0 \text{ at } i = 0)\,. \tag{16} $$

The spatial acceleration can be calculated similarly in the following form:

$$ \mathbf{a}_i = {}^h_i\boldsymbol{X}^T\mathbf{a}_h + \mathbf{v}_i\widehat{\times}\boldsymbol{S}_i\dot{\phi}_i + \boldsymbol{S}_i\ddot{\phi}_i, \quad (\mathbf{a}_i = \mathbf{a}_0 \text{ at } i = 0)\,. \tag{17} $$

where $\widehat{\times}$ denotes the spatial cross-product operator associated with a spatial vector $\left[\boldsymbol{a}^T, \boldsymbol{b}^T\right]^T$ defined as:

$$ \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix}\widehat{\times} = \begin{bmatrix} \widetilde{\boldsymbol{b}} & \widetilde{\boldsymbol{a}} \\ \mathbf{0} & \widetilde{\boldsymbol{b}} \end{bmatrix} \in R^{6\times 6}. $$

The iterative calculation of the composite force acted on the link h is defined as:

$$\boldsymbol{f}_h = \boldsymbol{f}_h^* + {}^h_i\boldsymbol{X}\boldsymbol{f}_i \,, \;\; (\boldsymbol{f}_n = \boldsymbol{f}_n^*) \,, \tag{18}$$

where

$$\boldsymbol{f}_i^* = \boldsymbol{I}_i\mathbf{a}_i + \mathbf{v}_i \widehat{\times} \boldsymbol{I}_i\mathbf{v}_i \,. \tag{19}$$

$\boldsymbol{I}_i$ is the spatial inertia matrix of link i in frame i expressed as:

$$\boldsymbol{I}_i = {}^i_{c_i}\boldsymbol{X}\boldsymbol{I}_{c_i}{}^i_{c_i}\boldsymbol{X}^T \,,$$

where $\boldsymbol{I}_i$ is a symmetric positive definite matrix. The integral of eq. (19) represents the spatial momentum of link h and the composite momentum of link h can be derived as :

$$\mathcal{L}_h = \boldsymbol{I}_h\mathbf{v}_h + {}^h_i\boldsymbol{X}\mathcal{L}_i \,, \; (\mathcal{L}_n = \boldsymbol{I}_n\mathbf{v}_n) \,. \tag{20}$$

In the above recursive calculation of the momentum, the composite rigid-body inertia of link h can be obtained as follows, which is the summation of inertia matrices of link h and its children links (Lilly, 1992):

$$\boldsymbol{I}_h^C = \boldsymbol{I}_h + {}^h_i\boldsymbol{X}\boldsymbol{I}_i^{Ch}{}_i\boldsymbol{X}^T, \; (\boldsymbol{I}_n^C = \boldsymbol{I}_n) \,. \tag{21}$$

Note that the velocity and the acceleration of link 0 are arbitrary, not only zero but also non-zero values are acceptable since our main focus is on the free-flying or free-floating space robots. We assume that those values are measurable or can be estimated.

4.2.2 Inverted Chain Approach

In the inverted chain approach, the transformation matrix, ${}^i_h\boldsymbol{X}$, is used to obtain each spatial quantity. As mentioned before, the direction to calculate each quantity is opposed to that of the forward chain approach.
The spatial velocity is expressed as:

$$\mathbf{v}_h = {}^i_h\boldsymbol{X}^T \left(\mathbf{v}_i - \boldsymbol{S}_i\dot{\boldsymbol{\phi}}_i\right) , \; (\mathbf{v}_h = \mathbf{v}_n \text{ at } h = n) \,. \tag{22}$$

The spatial acceleration is described as:

$$\mathbf{a}_h = {}^i_h\boldsymbol{X}^T \left(\mathbf{a}_i - \mathbf{v}_i \widehat{\times} \boldsymbol{S}_i\dot{\boldsymbol{\phi}}_i - \boldsymbol{S}_i\ddot{\boldsymbol{\phi}}_i\right) , \; (\mathbf{a}_h = \mathbf{a}_n \text{ at } h = n) \,. \tag{23}$$

The composite force is derived as:

$$\boldsymbol{f}_i = \boldsymbol{f}_i^* + {}^i_h\boldsymbol{X}\boldsymbol{f}_h \,, \;\; (\boldsymbol{f}_0 = \boldsymbol{f}_0^*) \,. \tag{24}$$

Under the same conditions, the results of the forward chain approach and the inverted chain approach are consistent.

5. Recursive Computation of Generalized Jacobian Matrix

This section presents efficient recursive calculations of the generalized Jacobian matrices introduced in (3) and (5). We introduce here the recursive algorithms in the framework of the spatial notation. Yokokohji proposed the recursive calculation of the generalized Jacobian matrix in (Yokokohji et al., 1993). However, by using the spatial notation, the recursive calculations can be improved to simpler and faster methods than one proposed in (Yokokohji et al., 1993).

5.1 Generalized Jacobian Matrix of Base Body (Link 0)

In the recursive expression, the total linear and angular momentum around the origin of frame 0 is expressed from eqs. (16) and (20) as follows :

$$\begin{aligned}\mathcal{L}_0 &= \sum_{k=0}^{n} {}^0_k\boldsymbol{X}\boldsymbol{I}_k\mathbf{v}_k \\ &= \boldsymbol{I}_0\mathbf{v}_0 + \sum_{k=1}^{n}\left[{}^0_k\boldsymbol{X}\boldsymbol{I}_k\left({}^0_k\boldsymbol{X}^T\mathbf{v}_0 + \boldsymbol{S}_k\dot{\phi}_k\right)\right] .\end{aligned} \tag{25}$$

where $\mathcal{M}_0 = {}^I_0 X\mathcal{L}_0$. From eq. (25), the velocity of the base can be expressed as a function of the velocity of the joint as :

$$\mathbf{v}_0 = -\left(\boldsymbol{I}_0^C\right)^{-1}\sum_{k=1}^{n} {}^0_k\boldsymbol{X}\boldsymbol{I}_k^C\boldsymbol{S}_k\dot{\phi}_k + \left(\boldsymbol{I}_0^C\right)^{-1}\mathcal{L}_0 , \tag{26}$$

where $\boldsymbol{I}_0^C$ and $\boldsymbol{I}_k^C$ denote composite rigid-body inertia matrix of the base and for link k computed by (21), respectively. The coefficient of the first term on the righthand side corresponds to the generalized Jacobian matrix of the base (link 0) in the frame 0 :

$${}^0\boldsymbol{J}_b^* = -\left(\boldsymbol{I}_0^C\right)^{-1}\sum_k {}^0_k\boldsymbol{X}^T\boldsymbol{I}_k^C\boldsymbol{S}_k . \tag{27}$$

Transformed into the inertial frame, eq. (27) equals to the expression (3) :

$$\boldsymbol{J}_b^* = {}^I_0\boldsymbol{X}\,{}^0\boldsymbol{J}_b^*. \tag{28}$$

5.2 Generalized Jacobian Matrix of Operational Point

Once ${}^0\boldsymbol{J}_b^*$ is obtained, the generalized Jacobian matrix of the operational point is straightforwardly derived. The spatial velocity of the operational point e_i can be expressed as:

$$\mathbf{v}_{e_i} = {}^n_{e_i}\boldsymbol{X}^T\mathbf{v}_n, \tag{29}$$

where $\mathbf{v}_n$ is the velocity of the link n, on which the operational point is determined, and is obtained from eq. (16). By substituting (26) into (29), the generalized Jacobian matrix of the operational point in the frame e_i can be derived as:

$${}^{e_i}\boldsymbol{J}_{m_i}^* = \sum_{k=1}^{n}\left({}^k_{e_i}\boldsymbol{X}^T\boldsymbol{S}_k - {}^0_{e_i}\boldsymbol{X}^T\left(\boldsymbol{I}_0^C\right)^{-1}\boldsymbol{I}_k^C\boldsymbol{S}_k\right) \tag{30}$$

Consequently, the generalized Jacobian matrix in the inertial frame, (5), can be obtained as follows:

$$\boldsymbol{J}_{m_i}^* = {}^I_{e_i}\boldsymbol{X}^{e_i}\boldsymbol{J}_{m_i}^*. \tag{31}$$

6. Efficient Algorithms of Operational Space Dynamics

This section describes recursive algorithms of the operational space dynamics, eq. (8). We recall here the operational space dynamics :

$$\boldsymbol{\Gamma}_e \ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e = \mathcal{F}_e \ , \quad \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{J}_e^T \mathcal{F}_e \ . \tag{32}$$

A main focus is on developing computational efficient algorithms of $\boldsymbol{\Gamma}_e$ and $\boldsymbol{\mu}_e$ of a n-link, p-operational-point branching space robot system. The derivation of $\boldsymbol{J}_e$ is omitted in this chapter since its algorithm is well-known. The algorithms of $\boldsymbol{\Gamma}_e$ and $\boldsymbol{\mu}_e$ are developed with the concept of the articulated body dynamics (Featherstone, 1987; Lilly, 1992). Firstly, the operational space formulation of a single operational point on the space robot is developed by using the inverted chain approach. Then, the algorithms of the multi-operational point system are further developed.

6.1 Single Operational Point in Space Robot

As mentioned in Section 1, a unique characteristic of the space robot is that the base satellite is inertially free and the system is invertible in its modeling unlike the ground-based robot system. Based on this characteristic, the operational space dynamics of the space robot is derived in the framework of the articulated-body dynamics. In the conventional articulated-body dynamics, the articulated-body inertia and its associated bias force are calculated inward from the operational point to the base body (link 0). When the system is inverted, the articulated-body dynamics is calculated in the opposed direction, namely outward from the base body to the operational point. This approach introduced as the *inertia propagation method* in (Lilly, 1992). We make use of the inverted chain approach for the iterative calculation of the inertia matrix, $\boldsymbol{\Gamma}_e$, and the bias force vector, $\boldsymbol{\mu}_e$ in the operational space.

6.1.1 Operational space inertia matrix

Operational space inertia matrix corresponds to the articulated-body inertia calculated from the base body to the operational point with the initial condition, $\boldsymbol{I}_0^A = \boldsymbol{I}_0$:

$$\boldsymbol{I}_i^A = \boldsymbol{I}_i + {}^i\boldsymbol{I}_h^A \, {}_h^i\boldsymbol{L} \ , \ (i = 0 \cdots n) \ , \tag{33}$$

where

$$ {}_h^i\boldsymbol{L} = \boldsymbol{E}_6 - \frac{\boldsymbol{S}_i \boldsymbol{S}_i^{T\,i}\boldsymbol{I}_h^A}{\alpha_i} \ ,$$

$$ {}^i\boldsymbol{I}_h^A = {}_h^i\boldsymbol{X}\boldsymbol{I}_h^A \, {}_h^i\boldsymbol{L}\boldsymbol{X}^T \ , \ \alpha_i = \boldsymbol{S}_i^{T\,i}\boldsymbol{I}_h^A\boldsymbol{S}_i .$$

$\boldsymbol{E}_6$ represents the 6×6 identity matrix. The superscript i at left side of the symbols describes the quantities of link h expressed in the frame i.
The symbols without the superscript at the left side expresses the quantities represented in their own frame. Consequently, the inertia matrix, $\boldsymbol{\Gamma}_e$, in the inertial frame is obtained by using the following spatial transformation.

$$\boldsymbol{\Gamma}_e = {}_e^I\boldsymbol{X}\boldsymbol{I}_e^A \, {}_e^I\boldsymbol{X}^T . \tag{34}$$

6.1.2 Operational space bias force vector

Likely the operational space inertia matrix, the associated bias force, $\boldsymbol{\mu}_e$, is calculated in the outward recursive manner. The bias force in the inverted chain approach is calculated as the following algorithm with the initial condition, $\boldsymbol{p}_0^A = \boldsymbol{p}_0 = \mathbf{v}_0 \widehat{\times} \boldsymbol{I}_0 \mathbf{v}_0$:

$$\boldsymbol{p}_i^A = \boldsymbol{p}_i + {}^i\boldsymbol{p}_h^A - {}^i_h\boldsymbol{L}\boldsymbol{c}_i - \frac{{}^i\boldsymbol{I}_h^A \boldsymbol{S}_i \boldsymbol{S}_i^T {}^i\boldsymbol{p}_h^A}{\alpha_i}, \tag{35}$$

where

$${}^i\boldsymbol{p}_h^A = {}^i_h\boldsymbol{X}\boldsymbol{p}_h^A \ , \ \ \boldsymbol{p}_i = \mathbf{v}_i \widehat{\times} \boldsymbol{I}_i \mathbf{v}_i \ , \ \ \boldsymbol{c}_i = \mathbf{v}_i \widehat{\times} \boldsymbol{S}_i \dot{\boldsymbol{\phi}}_i.$$

Finally, the operational space bias force in the inertial frame, $\boldsymbol{\mu}_e$, is obtained as follows:

$$\boldsymbol{\mu}_e = {}^I_e\boldsymbol{X}\boldsymbol{p}_e^A + \boldsymbol{\Gamma}_e \begin{bmatrix} \boldsymbol{v}_e \times \boldsymbol{\omega}_e \\ \mathbf{0}_3 \end{bmatrix}, \tag{36}$$

where the relationship between the spatial acceleration, $\mathbf{a}_i$, and the conventional acceleration of a point fixed in a rigid body, $\ddot{\boldsymbol{x}}_i$, is used here, *i.e.* $\ddot{\boldsymbol{x}}_i = \mathbf{a}_i - \left((\boldsymbol{v}_i \times \boldsymbol{\omega}_i)^T, \mathbf{0}_3^T \right)^T$, since the spatial acceleration, $\mathbf{a}_i$, differs from the conventional acceleration of a point fixed in a rigid body, $\ddot{\boldsymbol{x}}_i$ (Featherstone, 1987). The vector $\mathbf{0}_3$ represents the 3×1 zero vector.

6.2 Multi-Operational Points in Space Robot

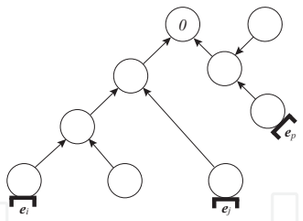
The lack of the fixed base and the dynamic coupling between each operational point lead to the computational complexity in the multi-operational points on the space robot. However, no fixed base provides an arbitrary choice of link 0 in the modeling of the kinematic connectivity. In addition, the proposed algorithms enable to formulate the dynamics of arbitrary operational points, not only the end-effectors in the real system but also the base-satellite or other controlled points in operational space. For instance, if both base-satellite and one end-effector are operated simultaneously, these two points are determined as operational points. In this subsection, the operational space inertia matrix and the bias force vector are derived based on the forward chain approach. Note that the direction of the recursive calculation is opposed to the approach in the previous subsection although the same symbols are used.

6.2.1 Operational space inertia matrix

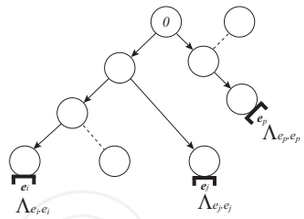
The inverse of the operational space inertia matrix, $\boldsymbol{\Gamma}_e^{-1}$, consists of diagonal matrices, $\boldsymbol{\Gamma}_{e_i,e_i}^{-1}$ and off-diagonal matrices, $\boldsymbol{\Gamma}_{e_i,e_j}^{-1}$ as expressed in the following equation:

$$\boldsymbol{\Gamma}_e^{-1} = \begin{bmatrix} \boldsymbol{\Gamma}_{e_1,e_1}^{-1} & \cdots & \boldsymbol{\Gamma}_{e_1,e_p}^{-1} \\ \vdots & \boldsymbol{\Gamma}_{e_i,e_j}^{-1} & \vdots \\ \boldsymbol{\Gamma}_{e_p,e_1}^{-1} & \cdots & \boldsymbol{\Gamma}_{e_p,e_p}^{-1} \end{bmatrix}. \tag{37}$$

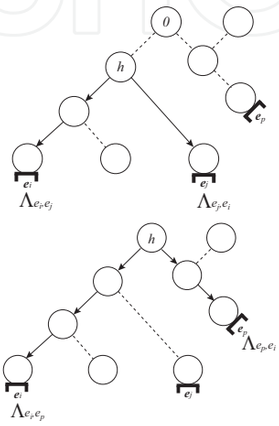
The inertial quantity $\boldsymbol{\Gamma}_{e_i,e_i}^{-1}$ describes the inertia of link i if the force is applied to only i-th operational point, and the inertial quantity $\boldsymbol{\Gamma}_{e_i,e_j}^{-1}$ describes the cross-coupling inertia matrix, which expresses the dynamic influence of the i-th operational point due to the force of the j-th operational point. Once the inverse of the operational space inertia matrix is calculated, the operational space inertia matrix is obtained as the following recursive calculation. Figure 5 shows the process of the calculation of the inertia matrix. In the figure, arrows indicate the direction of the calculation.



(a) step 1: Inward Recursion for $\boldsymbol{I}_h^A$



(b) step 2: Outward Recursion for $\boldsymbol{\Lambda}_{i,i}$



(c) step 3: Outward Recursion for $\boldsymbol{\Lambda}_{i,j}$

Fig. 5. Recursive process of the inertia matrix

1. **Inward Recursion**: Compute the articulated-body inertia of link i in the forward chain approach:

$$\boldsymbol{I}_h^A = \boldsymbol{I}_h + {}_i^h\boldsymbol{L}\boldsymbol{I}_i^A {}_i^h\boldsymbol{X}^T \ , \quad (\boldsymbol{I}_n^A = \boldsymbol{I}_n) \tag{38}$$

$${}_i^h\boldsymbol{L} = {}_i^h\boldsymbol{X}\left(\boldsymbol{E}_6 - \frac{\boldsymbol{I}_i^A\boldsymbol{S}_i\boldsymbol{S}_i^T}{\beta_i}\right) \ , \ \beta_i = \boldsymbol{S}_i^T\boldsymbol{I}_i^A\boldsymbol{S}_i.$$

2. **Outward Recursion**: Compute the block diagonal matrices:

$$\boldsymbol{\Lambda}_{i,i} = \frac{\boldsymbol{S}_i\boldsymbol{S}_i^T}{\beta_i} + {}_i^{h_i}\boldsymbol{L}^T\boldsymbol{\Lambda}_{h_i,h_i}{}_i^{h_i}\boldsymbol{L} \ , \quad \left(\boldsymbol{\Lambda}_{0,0} = \left(\boldsymbol{I}_0^A\right)^{-1}\right) \tag{39}$$

3. **Outward Recursion**: Compute the cross-coupling inertia matrices:

$$\boldsymbol{\Lambda}_{i,j} = \begin{cases} {}_i^{h_i}\boldsymbol{L}^T\boldsymbol{\Lambda}_{j,h_i}^T & if \ \ j = h \\ \boldsymbol{\Lambda}_{i,h_j}{}_j^{h_j}\boldsymbol{L} & if \ \ i = h \end{cases} \tag{40}$$

4. **Spatial Transformation**: Compute the inverse inertia matrices in the inertial frame:

$$\boldsymbol{\Gamma}_{e_i,e_j}^{-1} = {}_{e_i}^I\boldsymbol{X}\,\boldsymbol{\Lambda}_{e_i,e_j}\,{}_{e_j}^I\boldsymbol{X}^T \tag{41}$$

5. **Matrix Inversion**: Compute the operational space inertia matrix, $\mathbf{\Gamma}_e$, for the space robot by inverting $\mathbf{\Gamma}_e^{-1}$.

6.2.2 Operational space bias force vector

To derive the bias force vector in operational space, the bias acceleration vector of each operational point is firstly derived in a recursive manner. The multiplication of the operational space inertia matrix $\mathbf{\Gamma}_e$ and the bias acceleration provides the bias force. The recursive algorithm of the bias force vector is shown in the following.

1. **Inward Recursion**: Compute the bias force of link i in the forward chain approach from the operational points to the link 0:

$$\boldsymbol{p}_h^A = \boldsymbol{p}_h + \boldsymbol{X}_i^h \boldsymbol{p}_i^A + {}_i^h\boldsymbol{L}\boldsymbol{I}_i^A \boldsymbol{c}_i - \frac{{}_i^h\boldsymbol{X}\boldsymbol{I}_i^A \boldsymbol{S}_i \boldsymbol{S}_i^T \boldsymbol{p}_i^A}{\beta_i}\ , \quad (\boldsymbol{p}_n^A = \boldsymbol{p}_n) \tag{42}$$

where

$${}^h\boldsymbol{p}_i^A = {}_i^h\boldsymbol{X}\boldsymbol{p}_i^A\ ,\ \ \boldsymbol{p}_i = \mathbf{v}_i \widehat{\times} \boldsymbol{I}_i \mathbf{v}_i\ ,\ \ \boldsymbol{c}_i = \mathbf{v}_i \widehat{\times} \boldsymbol{S}_i \dot{\boldsymbol{\phi}}_i.$$

2. **Outward Recursion**: Compute the bias acceleration of each operational point:

$$\boldsymbol{a}_i = {}_i^h\boldsymbol{X}^T \boldsymbol{a}_h + \boldsymbol{c}_i + \boldsymbol{S}_i \boldsymbol{b}_i\ , \quad \left(\boldsymbol{a}_0 = \left(\boldsymbol{I}_0^A\right)^{-1} \boldsymbol{p}_0^A\right) \tag{43}$$

where

$$\boldsymbol{b}_i = -\frac{\boldsymbol{S}_i^T \left[\boldsymbol{I}_i^A \left(\boldsymbol{X}_i^h \boldsymbol{a}_h + \boldsymbol{c}_i\right) + \boldsymbol{p}_i\right]}{\beta_i}.$$

3. **Spatial Transformation**: Compute the bias acceleration of each operational point with respect to the inertial frame:

$${}^I\boldsymbol{a}_{e_i} = {}_{e_i}^I\boldsymbol{X}\boldsymbol{a}_{e_i} \tag{44}$$

4. **Matrix Multiplication**: Compute the bias force of the operational points with respect to the inertial frame:

$$\boldsymbol{\mu}_e = \mathbf{\Gamma}_e \left(\begin{bmatrix} \vdots \\ {}^I\boldsymbol{a}_{e_i} \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ \boldsymbol{v}_{e_i} \times \boldsymbol{\omega}_{e_i} \\ \mathbf{0}_3 \\ \vdots \end{bmatrix} \right)\ , \ (i = 1 \cdots p)\ , \tag{45}$$

where the relationship between the conventional acceleration and the spatial acceleration, $\ddot{\boldsymbol{x}}_i = \boldsymbol{a}_i - \left((\boldsymbol{v}_i \times \boldsymbol{\omega}_i)^T, \mathbf{0}_3^T\right)^T$, is used.

6.3 Free-Floating Space Robot

In the previous subsections, we derive the inertia matrix and the bias force of eq. (8). In the free-floating space robot, one needs to refer to eq. (12). Since the operational inertia matrix $\left(\mathbf{\Lambda}^{-1} + \mathbf{\Lambda}_b^{-1}\right)^{-1} = \mathbf{\Gamma}_e$, the bias force $\mathbf{\Gamma}_e \boldsymbol{\mu} = \boldsymbol{\mu}_e$ in eq. (12) and $\boldsymbol{J}_m^*$ is calculated in Section 5, one need to derive only the matrix $\mathbf{\Lambda}$. The inertia matrix $\mathbf{\Lambda}$ can be easily obtained by $\mathbf{\Lambda} = \left(\mathbf{\Gamma}_e^{-1} - \mathbf{\Lambda}_b^{-1}\right)^{-1}$, where the inertia matrix $\mathbf{\Lambda}_b$ corresponds to the composite inertia matrix of the base $\mathbf{\Lambda}_b = {}_0^I\boldsymbol{X}\boldsymbol{I}_0^C$.

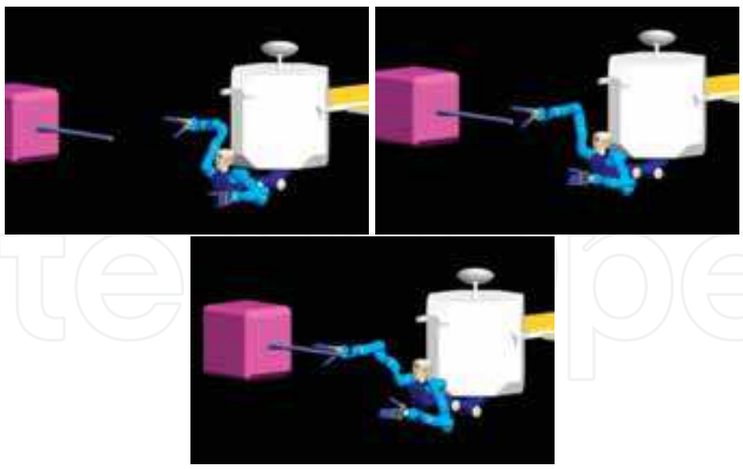


Fig. 6. Target grasping sequence, wait, approach, and grasp

7. Operational Space Task

The operational space task is illustrated by using the proposed algorithms. The robotic model considered here has total 24 degrees-of-freedom including two 7-degrees-of-freedom arms, a 4-degrees-of-freedom torso, and a 6-degrees-of-freedom base body. In practice, the redundancy of the system provides the null-space motion in the joint space. To obtain the null space motion, the classical joint space inverse dynamics is simply applied.The null space motion performs the self-motion in the joint space such as self-collision avoidance, posture behavior while the operational space motion is carried out.

Figure 6 shows the motion sequence for target grasping by a chaser-robot, consisting of waiting phase, approaching phase, and grasping phase. In those processes, the end-effector of the right hand, the head of the robot and the base body are determined as three operational points. In the simulation, it is demonstrated that the right arm approaches to the target while the rest of active joints are operated to keep the orientation of the head and the orientation of the base-satellite constant.

8. Conclusions

This chapter proposed efficient recursive algorithms of the operational space dynamics for the free-flying and the free-floating space robots. In the space robot, the operational space formulation is more complex than that of the ground-based robot due to the lack of the fixed base. However, by virtue of no fixed base, the space robot can be switched around in its modeling. By making use of this unique characteristic, firstly the operational space formulation of a single-serial-arm space robot has been developed. Then, the efficient algorithm of the operational space formulation of the branching-arms space robot has been proposed by using the concept of the articulated-body system. The realistic simulation with 24 DOF space robot system was illustrated to verify the efficiency of the proposed algorithms.

9. References

Chang, K.-S. & Khatib, O. (1999). Efficient algorithm for extended operational space inertia matrix, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, Kyongju, Korea, pp. 350 – 355.

Chang, K.-S. & Khatib, O. (2000). Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms, *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 850 – 856.

Featherstone, R. (1987). *Robot Dynamics Algorithms*, Kluwer Academic Publishers.

Khatib, O. (1987). A Unified Approach for Motion and Force control of Robot Manipulators: The Operational Space Formulation, *IEEE Journal of Robotics and Automation* **RA-3**(1): 43 – 53.

Lilly, K. W. (1992). *Efficient Dynamic Simulation of Robotic Mechanisms*, Kluwer Academic Publishers.

Russakow, J., Khatib, O. & Rock, S. M. (1995). Extended operational space formulation for serial-to-parallel chain (branching) manipulators, *Proc. of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, pp. 1056 – 1061.

Umetani, Y. & Yoshida, K. (1989). Resolved motion rate control of space manipulators with generalized jacobian matrix, *IEEE Transactions on Robotics and Automation* **5**(3): 303 – 314.

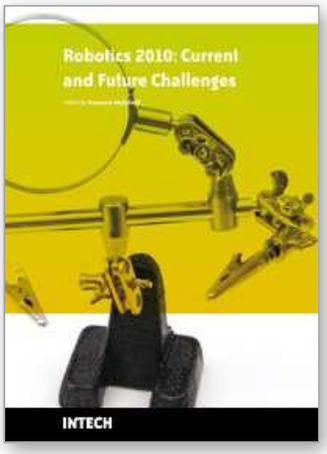
Xu, Y. & Kanade, T. (eds) (1993). *Space Robotics: Dynamics and Control*, Kluwer Academic Publishers.

Yokokohji, Y., Toyoshima, T. & Yoshikawa, T. (1993). Efficient computational algorithms for trajectory control of free-flying space robots with multiple arms, *IEEE Transactions on Robotics and Automation* **9**(5): 571 – 580.

IntechOpen

IntechOpen

IntechOpen



Robotics 2010 Current and Future Challenges

Edited by Houssem Abdellatif

ISBN 978-953-7619-78-7
Hard cover, 494 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Without a doubt, robotics has made an incredible progress over the last decades. The vision of developing, designing and creating technical systems that help humans to achieve hard and complex tasks, has intelligently led to an incredible variety of solutions. There are barely technical fields that could exhibit more interdisciplinary interconnections like robotics. This fact is generated by highly complex challenges imposed by robotic systems, especially the requirement on intelligent and autonomous operation. This book tries to give an insight into the evolutionary process that takes place in robotics. It provides articles covering a wide range of this exciting area. The progress of technical challenges and concepts may illuminate the relationship between developments that seem to be completely different at first sight. The robotics remains an exciting scientific and engineering field. The community looks optimistically ahead and also looks forward for the future challenges and new development.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Satoko Abiko and Gerd Hirzinger (2010). Operational Space Dynamics of a Space Robot and Computational Efficient Algorithm, Robotics 2010 Current and Future Challenges, Houssem Abdellatif (Ed.), ISBN: 978-953-7619-78-7, InTech, Available from: http://www.intechopen.com/books/robotics-2010-current-and-future-challenges/operational-space-dynamics-of-a-space-robot-and-computational-efficient-algorithm

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen