Computational Efficient Algorithms for Operational Space Formulation of Branching Arms on a Space Robot

Satoko Abiko and Gerd Hirzinger
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
82334, Weßling, Germany
satoko.abiko@dlr.de

***Abstract*— This paper presents efficient computational algorithms of operational space dynamics for free-flying and for free-floating space robots. Due to the lack of the fixed base, the operational space formulation of the space robot is more complex than the fixed base robot system. By paying attention to this unique characteristic, however, the novel algorithm of the operational space dynamics for a single-serial-arm space robot is derived. Furthermore, by using the concept of the articulated-body system, recursive computation algorithms of the operational space formulation for a branching-arms on the space robot is developed. The realistic dynamic simulations are illustrated to verify the computational efficiency.**

I. Introduction

The on-orbit servicing space robots are one of the challenging fields in the robotics and space technology. Main task of the on-orbit space robots would be the tracking, the grasping and the positioning of a target. To achieve these tasks, the operational space control is required and it is convenient to use operational space formulation directly.

The operational space formulation is an approach for dynamic modeling and control of a complex branching redundant mechanism at its tasks or operational control points. Khatib proposed the formulation for ground based robot systems in [1], [2]. Chang and Khatib introduced the efficient algorithms for this formulation, especially for operational space inertia matrix in [3], [4].

In the space robot, the operational space formulation is more complex due to the lack of the fixed base. By virtue of no fixed base, however, the space robot is invertible in its modeling. By making use of this unique characteristic, this paper firstly proposes an efficient algorithm for the operational space dynamics of a single-serial-arm space robot system. Then, by using the concept of the articulated-body algorithm [5], recursive computations of the operational space formulation for a branching-arms space robot are proposed. The numerical simulations with a two-arms space robot shown in Fig. 1 are illustrated.

The paper is organized as follows. Section II describes basic dynamic equations for free-flying and free-floating space robots. Section III derives the operational space formulation for the free-flying and the free-floating space robots. Section IV introduces a spatial notation and quantities to model complex robot kinematics and dynamics, which are utilized for the derivation of the proposed algorithms. Efficient recursive algorithms for the generalized Jacobian matrix [6] are presented in section V. In Section VI, computational efficient algorithms for the operational space formulation are proposed. The simulation results are illustrated in Section VII. The conclusions are summarized in Section VIII.

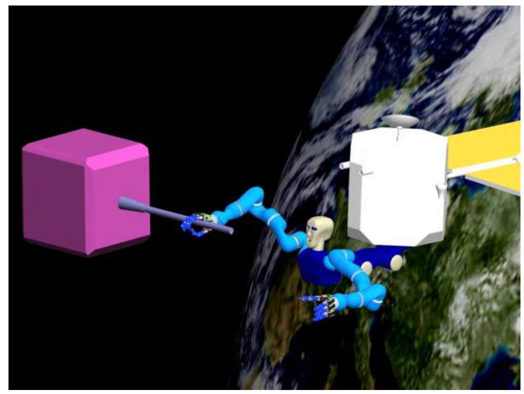

Fig. 1: Chaser-robot and target scenario

II. Basic Equations

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

A. Linear and Angular Momentum Equations

The motion of the space robot is generally governed by the momentum equations 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 :

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

where $\mathcal{L}_0 \in R^{6\times 1}$ represents the total linear and angular momentum around the base coordinate frame. In the absence of the external forces, the total momentum is conserved and the motion of the base can be determined by :

$$\dot{\boldsymbol{x}}_b = \boldsymbol{J}_b^* \dot{\boldsymbol{\phi}} + \boldsymbol{H}_b^{-1} \mathcal{L}_0 \ , \tag{2}$$

where

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

TABLE I: main notaitons

$\dot{\boldsymbol{x}}_b$	$\in R^{6\times 1}$	:	linear and angular velocity of the base.
$\dot{\boldsymbol{\phi}}$	$\in R^{n\times 1}$	:	linear and angular velocity of the arms.
$\dot{\boldsymbol{x}}_e$	$\in R^{6p\times 1}$	:	linear and angular velocity of the end-effectors.
$\boldsymbol{H}_b$	$\in R^{6\times 6}$	:	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 end-effectors.
$\boldsymbol{\tau}$	$\in R^{n\times 1}$	:	torque on the joints.
$\boldsymbol{J}_b$	$\in R^{6\times 6}$	:	Jacobian matrix for the base.
$\boldsymbol{J}_m$	$\in R^{6\times n}$	:	Jacobian matrix for the arms.
$\boldsymbol{J}_b^*$	$\in R^{6\times 6}$	:	Generalized Jacobian matrix for the base related to the i-th end-effector.
$\boldsymbol{J}_m^*$	$\in R^{6\times n}$	:	Generalized Jacobian matrix for the arms related to the i-th end-effector.

represents the generalized Jacobian matrix for the base-satellite [7]. By introducing the kinematic relationship for the end-effector, $\dot{\boldsymbol{x}}_e = \boldsymbol{J}_b\dot{\boldsymbol{x}}_b + \boldsymbol{J}_m\dot{\boldsymbol{\phi}}$, eq. (1) provides the velocity of the end-effector expressed as :

$$\dot{\boldsymbol{x}}_e = \boldsymbol{J}_m^*\dot{\boldsymbol{\phi}} + \boldsymbol{J}_b\boldsymbol{H}_b^{-1}\mathcal{L}_0 \ , \tag{4}$$

where

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

is called the generalized Jacobian matrix for a end-effector or a operational point [6]. The above two generalized Jacobian matrices, (3) and (5), are for the case of single serial arm on the base. Those matrices are simply extended to the multi-arm case. In section V, we derive recursive calculations for those two matrices.

B. Equations of Motion

The general dynamic equations for the space robot are described by the following expression [6]:

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

In the case that $\mathcal{F}_b$ is generated actively (*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.

C. Reduced Form for a Free-Floating Space Robot

The dynamic equations for the free-floating space robot can be furthermore reduced a form expressed with only joint acceleration, $\ddot{\boldsymbol{\phi}}$, by eliminating the base-satellite 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 $\boldsymbol{H}_m^* = \boldsymbol{H}_m - \boldsymbol{H}_{bm}^T\boldsymbol{H}_b^{-1}\boldsymbol{H}_{bm} \in R^{n\times n}$, $\boldsymbol{c}_m^* = \boldsymbol{c}_m - \boldsymbol{H}_{bm}^T\boldsymbol{H}_b^{-1}\boldsymbol{c}_b \in R^{n\times 1}$ denote generalized inertia matrix and generalized non-linear velocity dependent term, respectively.

III. Operational Space Formulation

The operational space formulation is convenient to model and to control the system in the operational space, which associates with the dynamics projected from the joint space dynamics into the operational space and its associated null space. In the space robots, two formulations are possible. One is for the free-flying space robot and the other is for the free-floating space robot. In this section, the general equations of motion for n-link with p operational points are discussed.

A. A Free-Flying Space Robot

The operational space formulation of the free-flying space robot can be described in the following form :

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

where

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

$\mathcal{F}_e$ denotes a $6p \times 1$ vector consisting of the 6×1 force of each of p end-effector. $\boldsymbol{J}_e$ denotes $6p \times n$ Jacobian matrix of Jacobian matrix of each end-effector as :

$$\mathcal{F}_e = \begin{bmatrix} \mathcal{F}_{e_1} \\ \vdots \\ \mathcal{F}_{e_p} \end{bmatrix} \quad \text{and} \quad \boldsymbol{J}_e = \begin{bmatrix} \boldsymbol{J}_{b1}, & \boldsymbol{J}_{m1} \\ \vdots & \vdots \\ \boldsymbol{J}_{b_p}, & \boldsymbol{J}_{m_p} \end{bmatrix} .$$

The operational space matrix for 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 :

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

B. A Free-Floating Space Robot

In the free-floating space robot, the system can be described as the reduced form in the joint space by using eq. (7). Its operational space formulation is derived as :

$$\boldsymbol{\Gamma}_e\ddot{\boldsymbol{x}}_e + \boldsymbol{\Gamma}_e\boldsymbol{\mu} = \boldsymbol{\Gamma}_e\boldsymbol{\Lambda}^{-1}\boldsymbol{J}_m^{*T+}\boldsymbol{\tau} + \mathcal{F}_e \ , \tag{11}$$

where

$$\boldsymbol{\Gamma}_e^{-1} = \boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda}_b^{-1} \in R^{6p\times 6p} \ ,$$

$$\boldsymbol{\Lambda}^{-1} = \boldsymbol{J}_m^*\boldsymbol{H}_m^{*-1}\boldsymbol{J}_m^{*T} \ , \ \boldsymbol{\Lambda}_b^{-1} = \boldsymbol{J}_b\boldsymbol{H}_b^{-1}\boldsymbol{J}_b^T \ .$$

The matrix, $(\mathbf{\Lambda}^{-1}+\mathbf{\Lambda}_b^{-1})$, corresponds to the inertia matrix described in eq. (9). The vector, $\boldsymbol{\mu}$, expresses the Coriolis and centrifugal forces in operational space in the reduced form :

$$\boldsymbol{\mu}=\mathbf{\Lambda}^{-1}\boldsymbol{c}_m^*-\dot{\boldsymbol{J}}_m^*\dot{\boldsymbol{\phi}}-\frac{d}{dt}(\mathbf{\Lambda}_b^{-1})\begin{bmatrix}\mathcal{P}\\ \mathcal{L}\end{bmatrix}\in R^{6p\times 1}.$$

Comparing 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. In Section VI, we propose the efficient algorithms for the operational space formulation represented here.

IV. Spatial Notation and Quantities

The *Spatial Notation* is well-known and intuitive notation to model kinematics and dynamics of articulated robot systems, introduced by Featherstone [3], [5]. In this section, the basic spatial notation and quantities are concisely reviewed which are used in the derivation of the efficient computational algorithms in the following sections.

A. 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 :

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

where $\boldsymbol{v}_i$, $\boldsymbol{\omega}_i$, $\mathcal{F}_i$ and $\mathcal{T}_i$ denote the 3×1 linear and angular velocity, the force and moment in terms of link i in frame i, respectively.

The simple joint model, $\boldsymbol{S}_i\in R^{6\times 1}$, for prismatic and rotational joint is defined, such that 1 is assigned along the prismatic or rotational axis : *e.g.*

$$\boldsymbol{S}_i=[\,0\;0\;1\;0\;0\;0\,]^T\quad\text{and}\quad\boldsymbol{S}_i=[\,0\;0\;0\;0\;0\;1\,]^T\;.$$

More complex multi-degrees-of-freedom joint is introduced in [5], [8].

The spatial inertia matrix of link i in frame c_i, $\boldsymbol{I}_{c_i}$, is a symmetric positive definite matrix as:

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

where m_i is the mass of link i, and $\mathbf{I}_{c_i}\in R^{3\times 3}$ is the inertia matrix around the center of mass of the link i in frame c_i. $\boldsymbol{E}_3$ is 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 :

$${}^h_i\boldsymbol{X}=\begin{bmatrix}{}^h_i\boldsymbol{R} & \mathbf{0}\\ \widetilde{{}^h\boldsymbol{r}_i}\,{}^h_i\boldsymbol{R} & {}^h_i\boldsymbol{R}\end{bmatrix}\in R^{6\times 6}. \qquad (12)$$

where ${}^h_i\boldsymbol{R}\in R^{3\times 3}$ is a rotation matrix and ${}^h\boldsymbol{r}_i\in R^{3\times 1}$ 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 (see. Fig. 2). Unlike 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 :

$${}^i_h\boldsymbol{X}=\begin{bmatrix}{}^h_i\boldsymbol{R}^T & \mathbf{0}\\ \left(\widetilde{{}^h\boldsymbol{r}_i}\,{}^h_i\boldsymbol{R}\right)^T & {}^h_i\boldsymbol{R}^T\end{bmatrix}\in R^{6\times 6}. \qquad (13)$$

B. Spatial Quantities

The most of the spatial quantities including the spatial velocity, acceleration and the spatial force, can be 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 root link toward the leaves or the end-effectors and the spatial force is obtained by inward recursion from the leaves toward the root link. It is here termed the *forward chain approach*.

In the second approach, the direction of the recursive calculation is opposed to the first one, such that the kinematic quantities are computed by inward recursion and the spatial force is obtained by outward recursion. It is here termed the *inverted chain approach*.

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:

$$\mathbf{v}_i={}^h_i\boldsymbol{X}^T\mathbf{v}_h+\boldsymbol{S}_i\dot{\phi}_i,\quad(\mathbf{v}_0=\mathbf{v}_0)\;. \qquad (14)$$

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

$$\mathbf{a}_i={}^h_i\boldsymbol{X}^T\mathbf{a}_h+\mathbf{v}_i\hat{\times}\boldsymbol{S}_i\dot{\phi}_i+\boldsymbol{S}_i\ddot{\phi}_i,\quad(\mathbf{a}_0=\mathbf{a}_0)\;. \qquad (15)$$

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

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

The iterative calculation for the total spatial force acted on the link h is defined as :

$$\boldsymbol{f}_h=\boldsymbol{f}_h^*+{}^h_i\boldsymbol{X}\boldsymbol{f}_i\;,\quad(\boldsymbol{f}_n=\boldsymbol{f}_n^*)\;, \qquad (16)$$

where

$$\boldsymbol{f}_i^*=\boldsymbol{I}_i\mathbf{a}_i+\mathbf{v}_i\hat{\times}\boldsymbol{I}_i\mathbf{v}_i\;. \qquad (17)$$

$\boldsymbol{I}_i$ 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. (17) represents the spatial momentum of link h and the composite momentum of link h can be derived as :

$$\mathcal{L}_h=\boldsymbol{I}_h\mathbf{v}_h+{}^h_i\boldsymbol{X}\mathcal{L}_i\;,\;(\mathcal{L}_n=\boldsymbol{I}_n\mathbf{v}_n)\;. \qquad (18)$$

The composite rigid-body inertia of link h, $\boldsymbol{I}_h^C$, is the summation of inertia matrices of link h and its children links [8]:

$$\boldsymbol{I}_h^C=\boldsymbol{I}_h+{}^h_i\boldsymbol{X}\boldsymbol{I}_i^C\,{}^h_i\boldsymbol{X}^T,\;(\boldsymbol{I}_n^C=\boldsymbol{I}_n)\;. \qquad (19)$$

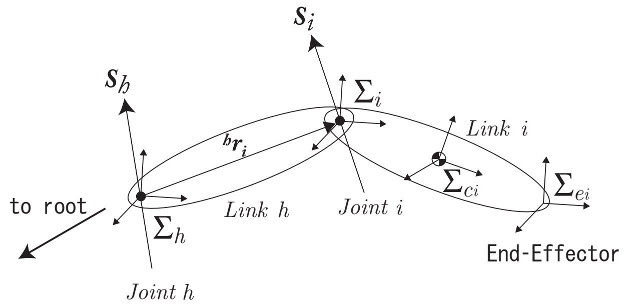

Fig. 2: Notation Representation

Note that the velocity and the acceleration of root link are non-zero values 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.

2) Inverted Chain Approach: In the inverted chain approach, the transformation matrix, ${}^i_h\boldsymbol{X}$, is used to obtain each spatial quantities.

The spatial velocity is expressed as :

$$\mathbf{v}_h = {}^i_h\boldsymbol{X}^T \left(\mathbf{v}_i - \boldsymbol{S}_i \dot{\phi}_i \right) , \ (\mathbf{v}_n = \mathbf{v}_n) \ . \tag{20}$$

The spatial acceleration is described as :

$$\mathbf{a}_h = {}^i_h\boldsymbol{X}^T \left(\mathbf{a}_i - \mathbf{v}_i \hat{\times} \boldsymbol{S}_i \dot{\phi}_i - \boldsymbol{S}_i \ddot{\phi}_i \right) , \ (\mathbf{a}_n = \mathbf{a}_n) \ . \tag{21}$$

The spatial force is derived as :

$$\boldsymbol{f}_i = \boldsymbol{f}^*_i + {}^i_h\boldsymbol{X} \boldsymbol{f}_h \ , \quad (\boldsymbol{f}_0 = \boldsymbol{f}^*_0) \ . \tag{22}$$

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

V. Recursive Computation of the Generalized Jacobian Matrix

This section presents efficient recursive calculations for 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 for the generalized Jacobian matrix in [7]. By using the spatial notation, however, the recursive calculations can be improved to simpler and faster methods than one proposed in [7].

A. Generalized Jacobian for the Base

In the recursive expression, the total linear and angular momentum around the origin of frame 0 is expressed from eqs. (14) and (18) as follows :

$$\begin{aligned} \mathcal{L}_0 &= \sum_{k=0}^{n} {}^0_k\boldsymbol{X} \boldsymbol{I}_k \mathbf{v}_k \\ &= \boldsymbol{I}_0 \mathbf{v}_0 + \sum_{k=1}^{n} \left[{}^0_k\boldsymbol{X} \boldsymbol{I}_k \left({}^0_k\boldsymbol{X}^T \mathbf{v}_0 + \boldsymbol{S}_k \dot{\phi}_k \right) \right] \ . \end{aligned} \tag{23}$$

From eq. (23), the velocity of the base can be expressed as a function of the velocity of the joint as :

$$\mathbf{v}_0 = -\left(\boldsymbol{I}^C_0 \right)^{-1} \sum_{k=1}^{n} {}^0_k\boldsymbol{X} \boldsymbol{I}^C_k \boldsymbol{S}_k \dot{\phi}_k + \left(\boldsymbol{I}^C_0 \right)^{-1} \mathcal{L}_0 \ , \tag{24}$$

where $\boldsymbol{I}^C_0$ and $\boldsymbol{I}^C_k$ denote composite rigid-body inertia matrix for the base and for link k computed by (19), respectively.

The coefficient of the first term on the righthand side corresponds to the generalized Jacobian for the base (link 0) in the frame 0 :

$${}^0\boldsymbol{J}^*_b = -\left(\boldsymbol{I}^C_0 \right)^{-1} \sum_{k} {}^0_k\boldsymbol{X}^T \boldsymbol{I}^C_k \boldsymbol{S}_k \ . \tag{25}$$

Transformed into the inertial frame, eq. (25) equals to the expression (3) :

$$\boldsymbol{J}^*_b = {}^I_0\boldsymbol{X} \ {}^0\boldsymbol{J}^*_b . \tag{26}$$

B. Generalized Jacobian for the End-Effector

Once ${}^0\boldsymbol{J}^*_b$ is obtained, the generalized Jacobian matrix for the end-effector is straightforwardly derived. The spatial velocity of the end-effector can be expressed by :

$$\mathbf{v}_e = {}^n_e\boldsymbol{X}^T \mathbf{v}_n , \tag{27}$$

where $\mathbf{v}_n$ is the velocity of the last link and is obtained from eq. (14). By substituting (24) into (27), the generalized Jacobian matrix for the end-effector in the end-effector frame e can be derived as :

$${}^e\boldsymbol{J}^*_m = \sum_{k=1}^{n} \left({}^k_e\boldsymbol{X}^T \boldsymbol{S}_k - {}^0_e\boldsymbol{X}^T \left(\boldsymbol{I}^C_0 \right)^{-1} \boldsymbol{I}^C_k \boldsymbol{S}_k \right) \tag{28}$$

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

$$\boldsymbol{J}^*_m = {}^I_e\boldsymbol{X} {}^e\boldsymbol{J}^*_m . \tag{29}$$

VI. Efficient Algorithms for Operational Space Formulation

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

$$\boldsymbol{\Gamma}_e \ddot{\boldsymbol{x}}_e + \boldsymbol{\mu}_e = \mathcal{F}_e \ , \quad \begin{bmatrix} \mathcal{F}_b \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{J}^T_e \mathcal{F}_e \ . \tag{30}$$

A main focus is to develop computational efficient algorithms for $\boldsymbol{\Gamma}_e$ and $\boldsymbol{\mu}_e$ for n-link, p-operational-point branching space robot system. The derivation of $\boldsymbol{J}_e$ is omitted in this paper since its algorithm is well-known.

The algorithms for $\boldsymbol{\Gamma}_e$ and $\boldsymbol{\mu}_e$ are developed with the concept of the articulated body dynamics [5], [8]. Firstly, the operational space formulation of a single-serial-arm space robot is developed by using the inverted chain approach. Then, the algorithms for the branching-arms space robot is developed.

A. A Single-Serial-Arm System

A unique characteristic of the space robot is that there is no fixed base and the system is invertible in its modeling. Based on this insight, the articulated-body dynamic equation is applied to derive the operational space dynamics of the space robot. In the conventional articulated-body dynamics, the articulated-body inertia and its associated bias force are calculated inward from the end-effector to the root. When the system is inverted, the articulated-body dynamics is calculated outward from the root to the end-effector. Then, one can obtain the dynamic equation for the operational space efficiently. This approach for the inertia matrix is introduced as the *inertia propagation method* in [8]. The algorithm for the inertia matrix is briefly reviewed here. Then, we make use of the inverted chain approach for the iterative calculation of the bias force vector in the operational space.

1) Operational space inertia matrix: The articulated-body inertia is calculated from the base to the end-effector with the initial condition, $\boldsymbol{I}_0^A = \boldsymbol{I}_0$:

$$\boldsymbol{I}_i^A = \boldsymbol{I}_i + {}^i\boldsymbol{I}_h^A {}_h^i\boldsymbol{L} \ , \ (i = 0 \cdots n) \ , \tag{31}$$

where

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

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

$\boldsymbol{E}_6$ denotes the 6×6 identity matrix.

Since the articulated-body inertia of end-effector corresponds to the operational space inertia matrix in the end-effector frame, the inertia matrix $\boldsymbol{\Gamma}_e$ described in the inertial frame can be derived by the spatial transformation.

$$\boldsymbol{\Gamma}_e = {}_e^I\boldsymbol{X} \boldsymbol{I}_e^A {}_e^I\boldsymbol{X}^T. \tag{32}$$

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, $\boldsymbol{p}_0^A = \boldsymbol{p}_0$:

$$\boldsymbol{p}_i^A = \boldsymbol{p}_i + {}^i\boldsymbol{p}_h^A - {}_h^i\boldsymbol{L}\boldsymbol{c}_i - \frac{{}^i\boldsymbol{I}_h^A \boldsymbol{S}_i \boldsymbol{S}_i^T {}^i\boldsymbol{p}_h^A}{\alpha_i}, \tag{33}$$

where

$$ {}^i\boldsymbol{p}_h^A = {}_h^i\boldsymbol{X}\boldsymbol{p}_h^A \ , \quad \boldsymbol{p}_i = \mathrm{v}_i \hat{\times} \boldsymbol{I}_i \mathrm{v}_i \ , \quad \boldsymbol{c}_i = \mathrm{v}_i \hat{\times} \boldsymbol{S}_i \dot{\boldsymbol{\phi}}_i.$$

Since the operational space bias force in the inertial frame, $\boldsymbol{\mu}_e$, is obtained from the spatial transformation of the bias force of the end-effector, $\boldsymbol{p}_e^A$, in the end-effector frame, the bias force $\boldsymbol{\mu}_e$ is obtained as:

$$\boldsymbol{\mu}_e = {}_e^I\boldsymbol{X}\boldsymbol{p}_e^A + \boldsymbol{\Gamma}_e \begin{bmatrix} \boldsymbol{v}_e \times \boldsymbol{\omega}_e \\ \boldsymbol{0}_3 \end{bmatrix} \ , \tag{34}$$

where the relationship between the spatial acceleration, $\boldsymbol{a}_i$, and the conventional acceleration of a point fixed in a rigid body, $\ddot{\boldsymbol{x}}_i$, is used here, namely $\ddot{\boldsymbol{x}}_i = \boldsymbol{a}_i - \left((\boldsymbol{v}_i \times \boldsymbol{\omega}_i)^T , \boldsymbol{0}_3^T \right)^T$, since the spatial acceleration, $\boldsymbol{a}_i$, differs from the conventional acceleration of a point fixed in a rigid body, $\ddot{\boldsymbol{x}}_i$ [5]. The vector $\boldsymbol{0}_3$ denotes the 3×1 zero vector.

B. A Branching-Arms System

In the branching-arms system, one obtains firstly the inverse inertia matrix and the bias acceleration in the operational space. In the space robot, due to the lack of the fixed base, the initial condition of each term is defined by the articulated-body dynamics.

A main feature of the operational space dynamics for the space robot is that one can 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 one need to control the base-satellite and one end-effector simultaneously, those two points can be determined as operational points. The proposed algorithm for the inertia matrix is an extension of the method proposed in [4] for the space robot. The algorithm for the bias force is direct recursive calculation instead of the calculation of each term in eq. (10).

Note that the articulated-body dynamics used in this section is based on the forward chain approach, namely the articulated-body inertia and its bias force are calculated from the end-effectors to the root. Its direction is opposed to the approach of the previous subsection. Therefore, the articulated-body inertia and the bias force described in this section are different from those in the previous subsection.

1) Operational space inertia matrix: The operational space inertia matrix consists of diagonal matrices, $\boldsymbol{\Lambda}_{e_i,ei}$ and off-diagonal matrices, $\boldsymbol{\Lambda}_{e_i,e_j}$. The inertial quantities $\boldsymbol{\Lambda}_{e_i,e_i}$ describes the inertia of link i if the force is applied to only i-th operational point, and inertial quantities $\boldsymbol{\Lambda}_{e_i,e_j}$ describes the cross-coupling inertia matrices, which express the dynamic influence of the i-th operational point due to the force of the j-th operational point.

1. **Inward Recursion**: Compute the articulated-body inertia of link i in the forward chain approach with initial condition, $\boldsymbol{I}_{e_i}^A = \boldsymbol{I}_{e_i}$ $(i = 1 \cdots p)$:

$$\boldsymbol{I}_h^A = \boldsymbol{I}_h + {}_i^h\boldsymbol{L}\boldsymbol{I}_i^A {}_i^h\boldsymbol{X}^T \ , \tag{35}$$

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

2. **Outward Recursion**: Compute the block diagonal matrices with initial condition, $\boldsymbol{\Lambda}_{0,0} = \left(\boldsymbol{I}_0^A \right)^{-1}$:

$$\boldsymbol{\Lambda}_{i,i} = \frac{\boldsymbol{S}_i \boldsymbol{S}_i^T}{\beta_i} + {}_i^{h_i}\boldsymbol{L}^T \boldsymbol{\Lambda}_{h_i,h_i} {}_i^{h_i}\boldsymbol{L}, \tag{36}$$

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

$$\boldsymbol{\Lambda}_{i,j} = \begin{cases} {}_i^{h_i}\boldsymbol{L}^T \boldsymbol{\Lambda}_{j,h_i}^T & if \quad j = h \\ \boldsymbol{\Lambda}_{i,h_j} {}_j^{h_j}\boldsymbol{L} & if \quad i = h \end{cases} \tag{37}$$

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

$$\boldsymbol{\Gamma}_{e_i,e_j}^{-1} = {}_{e_i}^I\boldsymbol{X} \ \boldsymbol{\Lambda}_{e_i,e_j} \ {}_{e_j}^I\boldsymbol{X}^T \tag{38}$$

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

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 the recursive manner. The multiplication of the operational space inertia matrix $\boldsymbol{\Gamma}_e$ and the derived bias acceleration provides the bias force.

1. **Inward Recursion**: Compute the bias force of link i in the forward chain approach from the operational points to the root with initial condition, $\boldsymbol{p}_{e_i}^A = \boldsymbol{p}_{e_i}$ $(i = 1 \cdots p)$:

$$\boldsymbol{p}_h^A = \boldsymbol{p}_h + \boldsymbol{X}_i^h \boldsymbol{p}_i^A + {}_i^h\boldsymbol{L}\boldsymbol{I}_i^A \boldsymbol{c}_i - \frac{{}_i^h\boldsymbol{X}\boldsymbol{I}_i^A \boldsymbol{S}_i \boldsymbol{S}_i^T \boldsymbol{p}_i^A}{\beta_i} \ , \tag{39}$$

where

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

2. **Outward Recursion**: Compute the bias acceleration of each operational point with initial condition, $\boldsymbol{a}_0 = \left(\boldsymbol{I}_0^A\right)^{-1} \boldsymbol{p}_0^A$:

$$ \boldsymbol{a}_i = {}^h_i\boldsymbol{X}^T \boldsymbol{a}_h + \boldsymbol{c}_i + \boldsymbol{S}_i \boldsymbol{b}_i \ , \tag{40} $$

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:

$$ {}^I\boldsymbol{a}_{e_i} = {}^I_{e_i}\boldsymbol{X}\boldsymbol{a}_{e_i} \tag{41} $$

4. **Matrix Multiplication**: Compute the bias force in the operational points:

$$ \boldsymbol{\mu}_e = \boldsymbol{\Gamma}_e \left(\begin{bmatrix} \vdots \\ {}^I\boldsymbol{a}_{e_i} \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ \boldsymbol{v}_{e_i} \times \boldsymbol{\omega}_{e_i} \\ \mathbf{0}_3 \\ \vdots \end{bmatrix} \right) \ , \ (i = 0 \cdots p) \ , \tag{42} $$

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

C. Free-Floating Space Robot

In the previous subsections, we derive the inertia matrix and the bias force of eq. (8). In the case of the free-floating space robot, one need to refer to eq. (11). Since the operational inertia matrix $\left(\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda}_b^{-1}\right)^{-1} = \boldsymbol{\Gamma}_e$, the bias force $\boldsymbol{\Gamma}_e\boldsymbol{\mu} = \boldsymbol{\mu}_e$ in eq. (11) and $\boldsymbol{J}_m^*$ is calculated in Section V, one need to derive only the matrix $\boldsymbol{\Lambda}$. The inertia matrix $\boldsymbol{\Lambda}$ can be easily obtained by $\boldsymbol{\Lambda} = \left(\boldsymbol{\Gamma}_e^{-1} - \boldsymbol{\Lambda}_b^{-1}\right)^{-1}$, where the inertia matrix $\boldsymbol{\Lambda}_b^{-1}$ can be derived by using the composite inertia matrix of the base-satellite $\boldsymbol{I}_0^C$.

VII. 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-satellite.

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 3 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-satellite are determined as three operational points. In the simulation, it is demonstrated that the right arm approaches

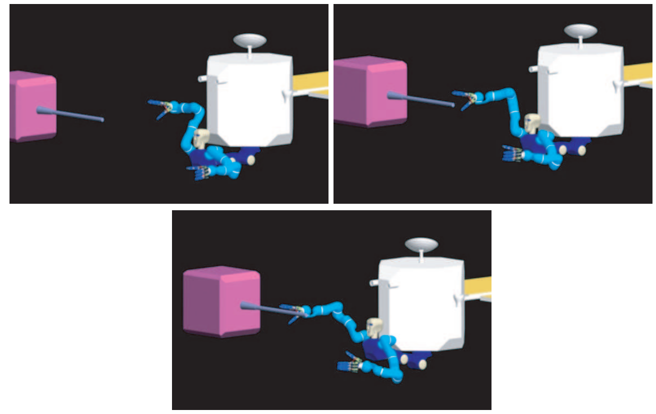

Fig. 3: Target grasping sequence, wait, approach, and grasp

to the target while the rest of active joints are operated to keep the orientation of the head and the orientation of the base constant.

VIII. Conclusions

This paper proposed efficient recursive algorithms of the operational space dynamics for the free-flying space robots and for the free-floating space robots.

In the space robot, the operational space formulation is rather complex due to the lack of the fixed base. However, by virtue of no fixed base, the space robot can be switched around. By making use of this unique characteristic, firstly the operational space formulation for a single-serial-arm space robot is developed. Then, the efficient algorithm for the operational space formulation of the branching-arms space robot is 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.

References

[1] O. Khatib, "A Unified Approach for Motion and Force control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43 – 53, 1987.

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

[3] K.-S. Chang and O. Khatib, "Efficient algorithm for extended operational space inertia matrix," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Kyongju, Korea, October 1999, pp. 350 – 355.

[4] ——, "Operational space dynamics: Efficient algorithms for modeling and control of branching mechanisms," in *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 850 – 856.

[5] R. Featherstone, *Robot Dynamics Algorithms.* Kluwer Academic Publishers, 1987.

[6] Y. Xu and T. Kanade, Eds., *Space Robotics: Dynamics and Control.* Kluwer Academic Publishers, 1993.

[7] Y. Yokokohji, T. Toyoshima, and T. Yoshikawa, "Efficient computational algorithms for trajectory control of free-flying space robots with multiple arms," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 571 – 580, October 1993.

[8] K. W. Lilly, *Efficient Dynamic Simulation of Robotic Mechanisms.* Kluwer Academic Publishers, 1992.