DOI: 10.3901/CJME.2010.**.***, available online at www.cjmenet.com; www.cjmenet.com.cn

Spatial Operator Algebra for Free-Floating Space Robot Modeling and Simulation

TIAN Zhixiang* and WU Hongtao

Department of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Received April 24, 2009; revised July 9, 2010; accepted July 30, 2010; published electronically **, 2010

Abstract: As the dynamic equations of space robots are highly nonlinear, strongly coupled and nonholonomic constrained, the efficiency of current dynamic modeling algorithms is difficult to meet the requirements of real-time simulation. This paper combines an efficient spatial operator algebra(SOA) algorithm for base fixed robots with the conservation of linear and angular momentum theory to establish dynamic equations for the free-floating space robot, and analyzes the influence to the base body's position and posture when the manipulator is capturing a target. The recursive Newton-Euler kinematic equations on screw form for the space robot are derived, and the techniques of the sequential filtering and smoothing methods in optimal estimation theory are used to derive an innovation factorization and inverse of the generalized mass matrix which immediately achieve high computational efficiency. The high efficient SOA algorithm is spatially recursive and has a simple math expression and a clear physical understanding, and its computational complexity grows only linearly with the number of degrees of freedom. Finally, a space robot with three degrees of freedom manipulator is simulated in Matematica 6.0. Compared with ADAMS, the simulation reveals that the SOA algorithm is much more efficient to solve the forward and inverse dynamic problems. As a result, the requirements of real-time simulation for dynamics of free-floating space robot are solved and a new analytic modeling system is established for free-floating space robot.

Key words: nonholonomic constrained, spatial operator algebra, dynamic, free-floating space robot

1 Introduction

Dynamic modeling and simulation have played a very important role in the design and control of the multibody systems such as spacecraft, robot and automobile, especially in the more complex system, the free-floating space robot. The motion control of the free-floating robot is not like the terrestrial robot, and one mainly difference is that the base body of the free-floating robot has six degrees of freedom that the control methods for terrestrial robot can not be easily applied to free-floating space robot directly[1]. The free-floating space robot is also an underactuated robot that has fewer actuators than degrees of freedom, in which the six base body degrees of freedom are passive and other degrees of freedom are active[2]. In recent years, the development of the science and technology, especially the robotics and astronavigation technology, has brought a great challenge to the dynamic modeling and simulation of systems which have a large number of degrees of freedom[2]. Therefore, it is necessary to establish a high efficient modeling algorithm for complex mechanical systems. Currently, there are three main methods for mechanical multibody dynamic modeling, one is Newton-Euler algorithm based on vector mechanics, another is Lagrange algorithm based on analytical mechanics, and the third one is Kane algorithm based on vector mechanics and analytical mechanics both. But these traditional algorithms are not efficient enough, the numbers of arithmetical operations of the Newton-Euler algorithm and Lagrangian algorithm grow as the cube of the number of degrees of freedom known as $O(n^3)$, and the numbers of arithmetical operations of the Kane algorithm is $O(n^2)$. As in highly complex and interactive space robotic systems, it is required that the algorithms should be quickly reconfigured in response to configuration changes. In this paper, the SOA algorithm[3–8] is proposed for space robots and it provides a high level architectural understanding of the mass matrix not readily apparent from detailed algorithm, and it is very propitious to develop computer programs for real-time simulation.

The remainder of this paper is presented as follows: In section 2, the recursive Newton-Euler kinematic and dynamic equations on screw form are presented. In section 3, we derive the forward dynamic equations based on spatial operator. In section 4, the factorization expression for generalized mass matrix is explained. Sections 5 and 6 describe the dynamics of free-floating space robot and

* Corresponding author. E-mail: tzxnuaa@126.com
This project is supported by National Natural Science Foundation of China (Grant No. 50375071), and Commission of Science, Technology and Industry for National Defense Pre-research Foundation of China (Grant No. C4220062501).

simulation of three degrees of freedom manipulator respectively, followed by conclusions.

2 Recursive Newton-Euler Equations

A serial manipulator with n rigid body links is considered as a robot. The links and joints in a manipulator are numbered in an increasing order from tip to base. This is different from the common numbering approach in which the numbers increase toward the tip that this order allows one to consider sequentially moving from joint 1 to joint n as going "forward" and moving form joint n to joint 1 as going "backward"[5]. The base body denotes link n, and the joint n–1 connects link n–1 to the base body. In this model all the joints of the space robot are rotational joints.

2.1 Kinematic relations between adjacent links

In order to establish dynamic equations for the multibody dynamic systems, we should first analyze the kinematic relations of the adjacent rigid bodies. As illustrated in Fig. 1, joint k connects the adjacent link k+1 and link k, and the motion of link k is defined as the motion of frame $O(k)$ with respect to frame $O(k+1)$. This scheme can be considered to be a modified Denavit-Hartenberg with reversed link numbering.

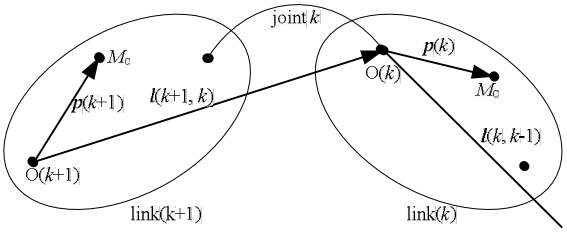

Fig. 1. Adjacent links in a manipulator

The kinematic equations of adjacent link k+1 and link k can be found in Ref. [9], and the symbol definitions used in the following paragraphs are as follows:

$O(k)$ —Fixed point on joint axis k, that can be viewed as the origin of a frame fixed in link k;
$m(k)$ —Mass of link k;
$M_c(k)$ —Mass center of link k;
$\boldsymbol{p}(k)$ —Vector from point $O(k)$ to point $M_c(k)$;
$\boldsymbol{l}(k+1,k)$ —Vector from point $O(k+1)$ to point $O(k)$;
$\boldsymbol{h}(k)$ —Unit vector along joint axis k;
$\theta(k)$ —Angle of link k with respect to link k+1 about joint axis k;
$\boldsymbol{\omega}(k)$ —Angular velocity of link k;
$\boldsymbol{v}(k)$ —The velocity of link k at point $O(k)$ of joint k;
$\boldsymbol{v}_c(k)$—Velocity of link k at point $M_c(k)$;
$\boldsymbol{F}(k)$ — Constraint force on link k at point $O(k)$;
$\boldsymbol{F}_c(k)$ — Net force at link k mass centre;
$\boldsymbol{N}(k)$ — Constraint moment on link k at joint k;
$\boldsymbol{T}(k)$ —Actuated torque at joint k;
$\boldsymbol{I}(k)$ —Inertia tensor of link k at the point $O(k)$;
$\boldsymbol{I}_c(k)$ —Inertia tensor of link k at the mass center.

2.2 Recursive Newton-Euler equations on screw form

The spatial velocity $\boldsymbol{V}(k)$ of the link k is defined as

$$\boldsymbol{V}(k)=\begin{pmatrix}\boldsymbol{\omega}(k)\\ \boldsymbol{v}(k)\end{pmatrix}, \tag{1}$$

with $\boldsymbol{\omega}(k)$ and $\boldsymbol{v}(k)$ denoting the angular and linear velocity of the link k respectively. The advantage of using spatial velocity is to decrease the computational complexity and unify the two vectors in one vector. Similarly, the spatial acceleration and spatial force are defined as

$$\boldsymbol{a}(k)=\begin{pmatrix}\dot{\boldsymbol{w}}(k)\\ \dot{\boldsymbol{v}}(k)\end{pmatrix}, \tag{2}$$

$$\boldsymbol{f}(k)=\begin{pmatrix}\boldsymbol{N}(k)\\ \boldsymbol{F}(k)\end{pmatrix}. \tag{3}$$

The spatial inertia $\boldsymbol{M}(k)$ of link k is defined as

$$\boldsymbol{M}(k)=\begin{pmatrix}\boldsymbol{I}(k) & m(k)\tilde{\boldsymbol{p}}(k)\\ -m(k)\tilde{\boldsymbol{p}}(k) & m(k)\boldsymbol{I}\end{pmatrix}\in\mathbf{R}^{6\times 6}. \tag{4}$$

The quantities defined above in the screw form are quite significant, as they imply that only the rules of ordinary matrix algebra are needed here. Also we define

$$\boldsymbol{H}^*(k)=\begin{pmatrix}\boldsymbol{h}(k)\\ 0\end{pmatrix},$$
$$\boldsymbol{\phi}(k+1,k)=\begin{pmatrix}\boldsymbol{I} & \tilde{\boldsymbol{l}}(k+1,k)\\ 0 & \boldsymbol{I}\end{pmatrix}\in\mathbf{R}^{6\times 6}. \tag{5}$$

$\boldsymbol{I}$ denotes the identity operator, and with the matrix identity $\tilde{\boldsymbol{x}}\boldsymbol{y}\equiv\boldsymbol{x}\times\boldsymbol{y}$, the kinematic equations of adjacent link k+1 and link k in Ref. [9] can be rewritten as follows.

For $k=n, n-1, \cdots, 1$ loop:

$$\begin{aligned}\boldsymbol{V}(k)&=\boldsymbol{\phi}^*(k+1,k)\boldsymbol{V}(k+1)+\boldsymbol{H}^*(k)\dot{\theta}(k),\\ \boldsymbol{\alpha}(k)&=\boldsymbol{\phi}^*(k+1,k)\boldsymbol{\alpha}(k+1)+\boldsymbol{H}^*(k)\ddot{\theta}(k)+\boldsymbol{a}(k).\end{aligned} \tag{6}$$

For $k=1, 2, \cdots, n$ loop

$$\begin{aligned} \boldsymbol{f}(k) &= \boldsymbol{\phi}(k,k-1)\boldsymbol{f}(k-1) + \boldsymbol{M}(k)\boldsymbol{\alpha}(k) + \boldsymbol{b}(k), \\ \boldsymbol{T}(k) &= \boldsymbol{H}(k)\boldsymbol{f}(k). \end{aligned} \tag{7}$$

Where

$$\boldsymbol{\alpha}(k) = \begin{pmatrix} \boldsymbol{\omega}(k+1)\times\boldsymbol{h}(k)\dot{\theta}(k) \\ \boldsymbol{\omega}(k+1)\times[\boldsymbol{\omega}(k+1)\times\boldsymbol{l}(k+1,k)] \end{pmatrix},$$

$$\boldsymbol{b}(k) = \begin{pmatrix} \boldsymbol{\omega}(k)\times\boldsymbol{I}(k)\cdot\boldsymbol{\omega}(k) \\ m(k)\boldsymbol{\omega}(k)\times[\boldsymbol{\omega}(k)\times\boldsymbol{p}(k)] \end{pmatrix}. \tag{8}$$

$\boldsymbol{\alpha}(k)$ denotes the Coriolis and centrifugal spatial acceleration at point $O(k)$. $\boldsymbol{b}(k)$ denotes the gyroscopic spatial force at point $O(k)$. $\boldsymbol{H}(k)$ is used to project the spatial force to the vector of the joint axis, and $\boldsymbol{H}^*(k)$ is used to project the spatial velocity and spatial acceleration to the joint axis. Similarly, $\boldsymbol{\phi}(k,k-1)$ is used for spatial force projection from joint *k*–1 to joint *k* and its transpose matrix $\boldsymbol{\phi}^*(k+1,k)$ is used for spatial velocity and spatial acceleration projection in the opposite direction. If joint *k* is a sliding joint, some quantities should be redefined as

$$\boldsymbol{H}^*(k) = \begin{pmatrix} 0 \\ \boldsymbol{h}(k) \end{pmatrix},$$

$$\boldsymbol{\alpha}(k) = \begin{pmatrix} 0 \\ \boldsymbol{\omega}(k+1)\times[\boldsymbol{\omega}(k+1)\times\boldsymbol{l}(k+1,k)+\boldsymbol{\omega}(k)] \end{pmatrix}. \tag{9}$$

Form Eqs. (6) and (7) we could find that recursive Newton-Euler equations on screw form are much more simply than the expressions in Ref. [9], and the calculation of the dynamic equations would change into the matrix calculation that would be a great advantage for matrix factorization and computer simulation.

3 Forward Dynamic Equations Based on Spatial Operators

The spatial velocity vector is defined as $\boldsymbol{V} = [\boldsymbol{V}(1),\cdots,\boldsymbol{V}(N)]^{\mathrm{T}}$. The quantity referred to as $\boldsymbol{V}$ is also quite significant that it unifies all the spatial velocities of the link in one vector which is beneficial to velocity recursion of the whole system that results in the high computation efficiency. Similarly, define quantities $\boldsymbol{\theta}$, $\boldsymbol{T}$, $\boldsymbol{\alpha}$, $\boldsymbol{f}$, $\boldsymbol{a}$, and $\boldsymbol{b}$. With the above definitions, Eqs. (6), (7) become

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{\phi}^*\boldsymbol{H}^*\dot{\boldsymbol{\theta}}, \\ \boldsymbol{\alpha} &= \boldsymbol{\phi}^*\boldsymbol{H}^*\ddot{\boldsymbol{\theta}} + \boldsymbol{\phi}^*\boldsymbol{a}, \\ \boldsymbol{f} &= \boldsymbol{\phi}(\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{b}), \\ \boldsymbol{T} &= \boldsymbol{H}\boldsymbol{f}. \end{aligned} \tag{10}$$

Where the spatial operators $\boldsymbol{\phi}$, $\boldsymbol{H}$ and $\boldsymbol{M}$ are given by

$$\boldsymbol{\phi} = (\boldsymbol{I} - \boldsymbol{\varepsilon}_{\phi})^{-1} = \begin{pmatrix} \boldsymbol{I} & 0 & \cdots & 0 \\ \boldsymbol{\phi}(2,1) & \boldsymbol{I} & \cdots & \vdots \\ \vdots & \vdots & & 0 \\ \boldsymbol{\phi}(n,1) & \boldsymbol{\phi}(n,2) & \cdots & \boldsymbol{I} \end{pmatrix},$$

$$\boldsymbol{H} = \mathrm{diag}[\boldsymbol{H}(1),\ \boldsymbol{H}(2),\cdots,\ \boldsymbol{H}(n)], \tag{11}$$

$$\boldsymbol{M} = \mathrm{diag}[\boldsymbol{M}(1),\ \boldsymbol{M}(2),\cdots,\ \boldsymbol{M}(n)].$$

$\boldsymbol{\varepsilon}_{\phi}$ is a shift operator whose elements are all zero, except along its lower subdiagonal. $\boldsymbol{\phi}$ denotes the rigid recursive operator that is used to implement the force recursion form tip to base that it is the most fundamental operator in this paper, and some other operators mentioned latter all have been derived based on this operator. The transpose matrix $\boldsymbol{\phi}^*$ is used to implement the velocity and acceleration recursion. Matrix $\boldsymbol{\phi}(i,j)$ is the Jacobian which relates the spatial velocity at point *i* to spatial velocity at point *j*, and it obeys the "group properties". By using the quantities defined above, Eq. (10) can be rewritten as follows

$$\begin{aligned} \boldsymbol{T} &= \mathbf{M}\ddot{\boldsymbol{\theta}} + \boldsymbol{C}, \\ \boldsymbol{M}_{\mathrm{G}} &= \boldsymbol{H}\boldsymbol{\phi}\boldsymbol{M}\boldsymbol{\phi}^*\boldsymbol{H}^*, \\ \boldsymbol{C} &= \boldsymbol{H}\boldsymbol{\phi}(\boldsymbol{M}\boldsymbol{\phi}^*\boldsymbol{a} + \boldsymbol{b}). \end{aligned} \tag{12}$$

$\boldsymbol{M}_{\mathrm{G}}$ denotes the generalized mass matrix and is referred to as Newton-Euler factorization of the mass matrix[7], and these results may be the simplest proof that reflects the equivalence of Lagrangian and Newton-Euler manipulator dynamics[10]. Vector $\boldsymbol{C}$ contains the velocity dependent Coriolis and centrifugal hinge force. Then the forward dynamic equations are obtained in much more simply forms.

4 Factorization Expression for Generalized Mass Matrix

As the operators $\boldsymbol{H}\boldsymbol{\phi}$ and $\boldsymbol{\phi}^*\boldsymbol{H}^*$ in Eq. (12) are not square matrix, the inverse matrix of generalized mass matrix can not be expressed by operators $\boldsymbol{H}\boldsymbol{\phi}$ and $\boldsymbol{\phi}^*\boldsymbol{H}^*$. But with the sequential filtering and smoothing methods in optimal estimation theory[11–12], the generalized mass matrix can be factorized into an innovation factorization form:

$$\boldsymbol{M}_{\mathrm{G}} = (\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})\boldsymbol{D}(\boldsymbol{I} + \boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})^*. \tag{13}$$

Immediately, the inverse of generalized mass matrix can be easily obtained

$$\boldsymbol{M}_{\mathrm{G}}^{-1} = (\boldsymbol{I} - \boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K})^*\boldsymbol{D}^{-1}(\boldsymbol{I} - \boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K}). \tag{14}$$

Operators $\boldsymbol{P}$, $\boldsymbol{D}$, $\boldsymbol{K}$ and $\boldsymbol{\psi}$ can be obtained from the following iteration.

For k=1, 2, $\cdots$, n loop:

$$\begin{aligned}
&\boldsymbol{P}(k)=\boldsymbol{\psi}(k,k-1)\boldsymbol{P}(k-1)\boldsymbol{\psi}^{*}(k,k-1)+\boldsymbol{M}(k),\\
&\boldsymbol{D}(k)=\boldsymbol{H}(k)\boldsymbol{P}(k)\boldsymbol{H}^{*}(k),\\
&\boldsymbol{G}(k)=\boldsymbol{P}(k)\boldsymbol{H}^{*}(k)\boldsymbol{D}^{-1}(k),\\
&\boldsymbol{\varepsilon}_{\psi}=\boldsymbol{\varepsilon}_{\phi}-\boldsymbol{K}(k+1,k)\boldsymbol{H}(k),\\
&\boldsymbol{K}(k+1,k)=\boldsymbol{\varepsilon}_{\phi}\boldsymbol{G}(k).
\end{aligned} \tag{15}$$

Where $\boldsymbol{\psi}$ is derived from Eqs. (11), (15). We could obtain

$$\boldsymbol{\psi}=(\boldsymbol{I}-\boldsymbol{\varepsilon}_{\psi})^{-1}=\begin{pmatrix}\boldsymbol{I} & 0 & \cdots & 0\\ \boldsymbol{\psi}(2,1) & \boldsymbol{I} & \cdots & \vdots\\ \vdots & \vdots & & 0\\ \boldsymbol{\psi}(n,1) & \boldsymbol{\psi}(n,2) & \cdots & \boldsymbol{I}\end{pmatrix}. \tag{16}$$

$\boldsymbol{\varepsilon}_{\psi}$ is similar to $\boldsymbol{\varepsilon}_{\phi}$, except that it produces articulated shifts instead of rigid shifts. $\boldsymbol{P}(k)$ denotes the sequence of spatial inertia which is the discrete Riccati equation driven by the link mass $\boldsymbol{M}_{\mathrm{G}}$. It is easy to seen that $\boldsymbol{P}(k)$ >0, that makes sure $\boldsymbol{D}^{-1}(k)$ exists. $\boldsymbol{D}(k)$ denotes the projection of the articulated body inertia, and it is memoryless and invertible. $\boldsymbol{G}(k)$ denotes the Kalman gains that is computed from the articulated body inertia and appears as a key element in the recursive Kalman filtering algorithm. $\boldsymbol{\psi}$ denotes the spatial Kalman filter transition operator, and its elements $\boldsymbol{\psi}(k+1,k)$ govern the transition of force from one link to the next that has the same properties with the operator $\boldsymbol{\phi}(k+1,k)$. Also, define these quantities in block diagonal matrix form as

$$\begin{aligned}
&\boldsymbol{P}=\mathrm{diag}[\boldsymbol{P}(1),\cdots,\boldsymbol{P}(n)],\\
&\boldsymbol{D}=\mathrm{diag}[\boldsymbol{D}(1),\cdots,\boldsymbol{D}(n)],\\
&\boldsymbol{G}=\mathrm{diag}[\boldsymbol{G}(1),\cdots,\boldsymbol{G}(n)],\\
&\boldsymbol{K}=\mathrm{diag}[\boldsymbol{K}(1),\cdots,\boldsymbol{K}(n)].
\end{aligned} \tag{17}$$

Then, Eq. (18) is

$$\begin{aligned}
&\boldsymbol{D}=\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{*},\\
&\boldsymbol{G}=\boldsymbol{P}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{D}^{-1},\\
&\boldsymbol{K}=\boldsymbol{\varepsilon}_{\phi}\boldsymbol{G}.
\end{aligned} \tag{18}$$

With the quantities and equations defined above, we have the following identities.

Identity 1 $\boldsymbol{\phi}\boldsymbol{\psi}^{-1}=\boldsymbol{I}+\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{H}$.

Proof: From Eqs. (15), (16) we have that

$$\begin{aligned}
\boldsymbol{\psi}^{-1}&=\boldsymbol{I}-\boldsymbol{\varepsilon}_{\psi}=\boldsymbol{I}-(\boldsymbol{\varepsilon}_{\phi}-\boldsymbol{K}\boldsymbol{H})\\
&=(\boldsymbol{I}-\boldsymbol{\varepsilon}_{\phi})+\boldsymbol{K}\boldsymbol{H}=\boldsymbol{\phi}^{-1}+\boldsymbol{K}\boldsymbol{H}
\end{aligned}.$$

Identity 2 $(\boldsymbol{I}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})^{-1}=\boldsymbol{I}-\boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K}$.

Proof: Using Identity 1 and a standard matrix identity

$$(\boldsymbol{I}+\boldsymbol{A}\boldsymbol{B})^{-1}=\boldsymbol{I}-\boldsymbol{A}(\boldsymbol{I}+\boldsymbol{B}\boldsymbol{A})^{-1}\boldsymbol{B},$$

we have that

$$\begin{aligned}
(\boldsymbol{I}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})^{-1}&=\boldsymbol{I}-\boldsymbol{H}(\boldsymbol{I}+\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{H})^{-1}\boldsymbol{\phi}\boldsymbol{K}\\
&=\boldsymbol{I}-\boldsymbol{H}(\boldsymbol{\phi}\boldsymbol{\psi}^{-1})^{-1}\boldsymbol{\phi}\boldsymbol{K}\\
&=\boldsymbol{I}-\boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K}
\end{aligned}.$$

Identity 3 $\boldsymbol{\phi}\boldsymbol{M}\boldsymbol{\phi}^{*}=\boldsymbol{P}+\tilde{\boldsymbol{\phi}}\boldsymbol{P}+\boldsymbol{P}\tilde{\boldsymbol{\phi}}^{*}+\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*}\boldsymbol{\phi}^{*}$.

Proof: From Eq. (15), we have that

$$\begin{aligned}
\boldsymbol{M}&=\boldsymbol{P}-\boldsymbol{\varepsilon}_{\psi}\boldsymbol{P}\boldsymbol{\varepsilon}_{\phi}^{*}\\
&=\boldsymbol{P}-\boldsymbol{\varepsilon}_{\phi}\boldsymbol{P}\boldsymbol{\varepsilon}_{\phi}^{*}+\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*}
\end{aligned}.$$

And multiplying by operator $\boldsymbol{\phi}$ from right and multiplying by operator $\boldsymbol{\phi}^{*}$ from left, we have

$$\begin{aligned}
\boldsymbol{\phi}\boldsymbol{M}\boldsymbol{\phi}^{*}&=\boldsymbol{\phi}(\boldsymbol{P}+\boldsymbol{\varepsilon}_{\phi}\boldsymbol{P}\boldsymbol{\varepsilon}_{\phi}^{*}+\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*})\boldsymbol{\phi}^{*}\\
&=\boldsymbol{P}+\tilde{\boldsymbol{\phi}}\boldsymbol{P}+\boldsymbol{P}\tilde{\boldsymbol{\phi}}^{*}+\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*}\boldsymbol{\phi}^{*}
\end{aligned},$$

where $\tilde{\boldsymbol{\phi}}$ is equal to $\boldsymbol{\phi}-\boldsymbol{I}$ and $\boldsymbol{\phi}\boldsymbol{\varepsilon}_{\phi}$.

With Identity 3 above, the generalized mass matrix can be rewritten as

$$\begin{aligned}
\boldsymbol{M}_{\mathrm{G}}&=\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{M}\boldsymbol{\phi}^{*}\boldsymbol{H}^{*}\\
&=\boldsymbol{H}(\boldsymbol{P}+\tilde{\boldsymbol{\phi}}\boldsymbol{P}+\boldsymbol{P}\tilde{\boldsymbol{\phi}}^{*}+\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*}\boldsymbol{\phi}^{*})\boldsymbol{H}^{*}\\
&=\boldsymbol{D}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{D}+\boldsymbol{D}\boldsymbol{K}^{*}\boldsymbol{\phi}^{*}\boldsymbol{H}^{*}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K}\boldsymbol{D}\boldsymbol{K}^{*}\boldsymbol{\phi}^{*}\boldsymbol{H}^{*}\\
&=(\boldsymbol{I}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})\boldsymbol{D}(\boldsymbol{I}+\boldsymbol{H}\boldsymbol{\phi}\boldsymbol{K})^{*}
\end{aligned}. \tag{19}$$

This is a new operator factorization of the generalized mass matrix, and easily with Identity 2, the inverse of generalized mass matrix can be derived:

$$\boldsymbol{M}_{\mathrm{G}}^{-1}=(\boldsymbol{I}-\boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K})^{*}\boldsymbol{D}^{-1}(\boldsymbol{I}-\boldsymbol{H}\boldsymbol{\psi}\boldsymbol{K}). \tag{20}$$

This leads to relatively easy recursive solutions to inverse dynamics problem that implements the $O(n)$ high efficient computational complexity.

5 Dynamics of Free-Floating Space Robot

The model of free-floating space robot can be considered as a number of links connecting by the hinge joints attached to a base body, and it is very similar to the terrestrial

manipulator, but the free-floating space robot has a few features that the terrestrial manipulators do not have. The base body of the terrestrial manipulator is immobile, and the position and posture of the base body would change when the free-floating space robot is moving its manipulator to the position to capture a target. As illustrated in Fig. 2, the base body is considered as link n, and an assumptive six degrees of freedom joint is used to attach the base body to the inertial reference frame. The manipulator of the free-floating space robot is considered as link 1 to link n–1 connecting by rotational joints.

The whole system follows the momentum conservation theory when no external force acts on the spacecraft, so the system's mass center will not change during the free-floating robot capturing a target. In the space multibody dynamics the system's mass centre is often taken as the origin of inertial reference frame. For the purpose of analyzing the coupling problem of the free-floating space robot with its base body, the SOA algorithm is used to establish dynamic equations of the free-floating space robot that is relative to the base body. Then the action and reaction theory and momentum conservation theory are used to analyze the influence of the position and posture of the base body. The velocity and acceleration of the base body[13] are given by

$$\begin{aligned}
\boldsymbol{v}(n) &= \boldsymbol{M}_1^{-1}\boldsymbol{H}_{\mathrm{v}},\\
\boldsymbol{\omega}(n) &= \boldsymbol{M}_2^{-1}\boldsymbol{H}_{\omega},\\
\dot{\boldsymbol{v}}(n) &= \boldsymbol{M}_1^{-1}\boldsymbol{F}_{\mathrm{a}},\\
\dot{\boldsymbol{\omega}}(n) &= \boldsymbol{M}_2^{-1}\boldsymbol{F}_{\mathrm{b}},
\end{aligned} \tag{21}$$

where

$$\begin{aligned}
\boldsymbol{M}_1 &= \mathrm{diag}(m_{\mathrm{c}}, m_{\mathrm{c}}, m_{\mathrm{c}}),\\
\boldsymbol{M}_2 &= \sum_{i=1,j=1}^{n}(\boldsymbol{R}_j\boldsymbol{I}_{\mathrm{c}}(i)\boldsymbol{R}_j^{\mathrm{T}} - m_i\tilde{\boldsymbol{r}}_i^2),\\
\boldsymbol{H}_{\mathrm{v}} &= -\sum_{i=1,j=1}^{n-1}\boldsymbol{R}_j m_i\dot{\boldsymbol{p}}(k),\\
\boldsymbol{H}_{\omega} &= -\sum_{i=1,j=1}^{n-1}[\boldsymbol{R}_j\boldsymbol{I}_{\mathrm{c}}\omega(i) + \boldsymbol{r}_i\times(\boldsymbol{R}_j m_i\dot{\boldsymbol{r}}_i^j)],\\
\boldsymbol{F}_{\mathrm{a}} &= -\sum_{i=1,j=1}^{n-1}\boldsymbol{R}_j\boldsymbol{F}_{\mathrm{c}}(i),\\
\boldsymbol{F}_{\mathrm{b}} &= -\sum_{i=1,j=1}^{n-1}[\boldsymbol{R}_j\boldsymbol{N}_{\mathrm{c}}(i) + \boldsymbol{r}_i\times(\boldsymbol{R}_j\boldsymbol{F}_{\mathrm{c}}(i))].
\end{aligned} \tag{22}$$

$\boldsymbol{r}_i$ denotes the vector of the link i in inertia reference frame, $\boldsymbol{r}_i^j$ the vector of the link i in reference frame j, m_c the total mass of the system, $\boldsymbol{R}_j$ the transformation matrix from reference frame j to inertia frame. With Eqs. (21), (22) the velocity and acceleration of the base body could be obtained, so the position and posture of the base body would be known in time.

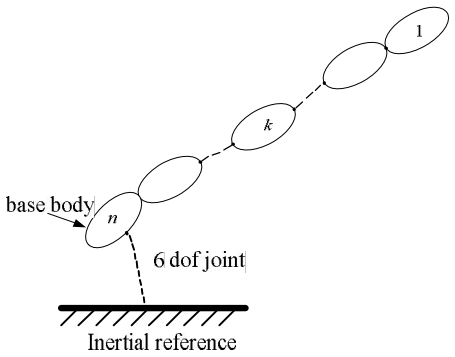

Fig. 2. Spacecraft with a manipulator

6 Simulation

In order to verify the algorithm above to be correct, the motions of the free-floating space robot and the influence to the base body were simulated by computer program in Mathematica 6.0, and the results were compared with ADAMS results. A manipulator with three degrees of freedom is taken, and Fig. 3 is the illustration of the physical model of simulation.

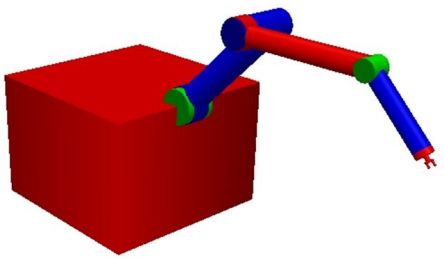
Fig. 3. Physical model of simulation

In the simulation, the integral step takes 0.001 s, and the actuated torques $T(1)$, $T(2)$ and $T(3)$ take –0.5 N·m, –0.3 N·m and –0.1 N·m respectively. The quantities of the model are as illustrated in Table.

Table. Quantities of the physical model

Link number	0	1	2	3
Inertia tensor I_{xx}/(kg·m^2)	8.33	0.84	0.84	0.11
Inertia tensor I_{yy}/(kg·m^2)	8.33	0.84	0.84	0.11
Inertia tensor I_{zz}/(kg·m^2)	8.33	0.013	0.013	0.006 3
Mass m/kg	50.0	10.0	10.0	5.0
Initial angle θ/rad	0	π/2	0	π/2

Fig. 4 is the displacement in x direction, and Fig. 5 is the displacement in y direction. Fig. 6 is the angular velocity of link. From the figures, we can find that the simulation

results are consistent with the ADAMS results, so the SOA algorithm is obvious correct. The simulation time in Mathematical 6.0 and ADAMS for 5 s actually need real time 20.2 s and 31.5 s respectively, indicating that the SOA algorithm is much more efficiency than ADAMS algorithm.

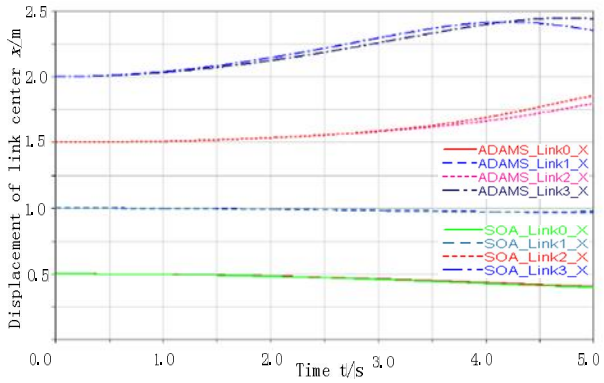

Fig. 4. Displacement in *x* direction

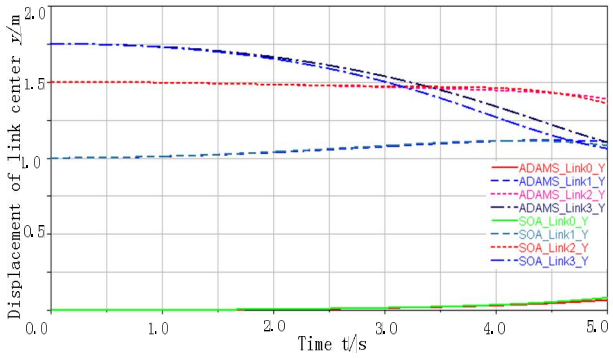

Fig. 5. Displacement in *y* direction

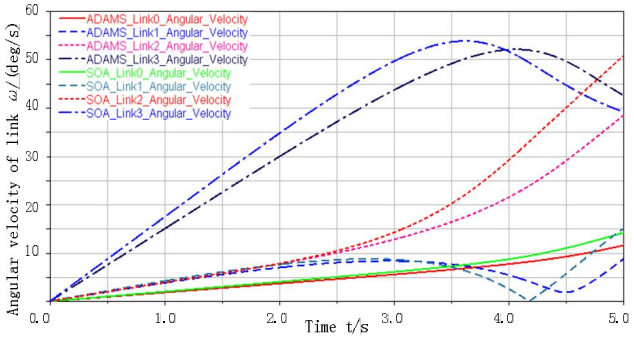

Fig. 6. Angular velocity of link

7 Conclusions

(1) In order to control the free-floating space robot well, it is very important to establish dynamic equations with high computational efficiency to analyze the dynamic characteristics. The high efficient SOA algorithm with the conservation of linear and angular momentum theory is combined to establish dynamic equations for the free-floating space robot.

(2) The universal dynamic equations with the SOA algorithm are derived for a serial manipulator with *n* rigid body links. Recursive Newton-Euler equations on screw form and the spatial operators are used to develop the factorizations and the inverse of the generalized mass matrix which results in the high computational efficiency.

(3) A manipulator with three degrees of freedom is taken into simulation. The simulation results reveal that the SOA algorithm is more efficiency than ADAMS. In the simulation the base body's posture and position are analyzed when the manipulator is moving.

(4) The free-floating space robot is a kind of underactuated system which must have nonholonomic characteristics. The nonholonomic characteristics bring a great challenge to control system that the nonlinear control must be applied. Techniques of the sequential filtering and smoothing methods in optimal estimation theory are used in the SOA algorithm and that will be easy to design the control system.

References

[1] YOJI UMETANI, KAZUYA YOSHIDA. Resolved motion rate control of space manipulators with generalized Jacobian matrix[J]. *IEEE Transactions on Robotics and Automation*, 1989, 5(3): 303–314.

[2] HUANG Panfeng, XU Yangsheng, LIANG Bin. Tracking trajectory planning of space manipulator for capturing operation[J]. *International Journal of Advanced Robotic Systems*, 2006, 3(3): 211–218.

[3] RODRIGUEZ G, JAIN A, KREUTZ-DELGADO K. Spatial operator algebra for multibody system dynamics[J]. *The Journal of the Astronautical Sciences*, 1992, 40: 27–50.

[4] RODRIGUEZ G, JAIN A K. Kreutz-Delgado. Spatial operator factorization and inversion of the manipulator mass matrix[J]. *IEEE Transactions on Robotics and Automation*, 1992, 8(1): 65–76.

[5] JAIN A, RODRIGUEZ G. Recursive flexible multibody system dynamics using spatial operators[J]. *Journal of Guidance, Control, and Dynamics*, November-December 1992, 15(6): 1 453–1 466.

[6] JAIN A. Unified formulation of dynamics for serial rigid multibody systems[J]. *Journal of Guidance, Control and Dynamics*, May-June 1991, 4(3): 531–542.

[7] JAIN A, RODRIGUEZ G. Recursive dynamics for geared robot manipulators[J]. *Proceedings of the 29*th *Conference on Decision and Control*, TA-16-2, 1990.

[8] RODRIGUEZ G. Statistical mechanics models for motion and force planning[J]. *Proc. SPIE Conf. Intelligent Control Adaptive Syst.* (Philadelphia, PA), Nov. 1989.

[9] HONG Jiazheng. *Computational dynamics of multibody systems*[M]. Beijing: Higher Education Press, 1999. (in Chinese)

[10] SILVER D B. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators[J]. *The International Journal of Robotics Research*, 1982, 1(2): 118–128

[11] GUILLERMO R. Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics[J]. *IEEE Journal of Robotics and Automation,* 1987, 3(6) : 624–639.

[12] BRYSON A, FRAZIER M. Smoothing for linear and nonlinear dynamical systems[J]. *Proc. Optimum Sys. Synthesis Conf. U.S. Air Force Tech. Rept*. ASD-TDR-63-119, Feb 1963.

[13] HONG Bingrong, LIU Chang'an, LI Huazhong. Torque control algorithm for free flying space robots capturing a moving target and its simulation[J]. *Journal of Astronautics*, Oct. 2000, 21(4): 64–70. (in Chinese)

Biographical notes

TIAN Zhixiang, born in 1984, is currently a PhD candidate in Department of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, China. His research interests include dynamic modeling and control of space robots.
Tel: +86-025-84891051; E-mail: tzxnuaa@126.com

WU Hongtao, born in 1962, is currently a professor in Department of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, China. His research interests include robotics, parallel mechanism, and multibody dynamics.
Tel: +86-025-84892503; E-mail: mehtwu@126.com