International Journal of Automation and Computing

14(2), April 2017, 169-178
DOI: 10.1007/s11633-017-1055-1

Singularity Robust Path Planning for Real Time Base Attitude Adjustment of Free-floating Space Robot

Cheng Zhou Ming-He Jin Ye-Chao Liu Ze Zhang Yu Liu Hong Liu

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

Abstract: This paper presents a singularity robust path planning for space manipulator to achieve base (satellite) attitude adjustment and end-effector task. The base attitude adjustment by the movement of manipulator will save propellant compared with conventional attitude control system. A task-priority reaction null-space control method is applied to achieve the primary task of adjusting attitude and secondary task of accomplishing end-effector task. Furthermore, the algorithm singularity is eliminated in the proposed algorithm compared with conventional reaction null-space algorithm. And the singular value filtering decomposition is introduced to dispose the dynamic singularity, the unit quaternion is also introduced to overcome representation singularity. Hence, a singularity robust path planning algorithm of space robot for base attitude adjustment is derived. A real time simulation system of the space robot under Linux/RTAI (realtime application interface) is developed to verify and test the feasibility and reliability of the method. The experimental results demonstrate the feasibility of online base attitude adjustment of space robot by the proposed algorithm.

Keywords: Space robot, path planning, base attitude adjustment, task priority, reaction null-space.

1 Introduction

With advances in space application, space robot is essential to implement space exploration mission including assembling the space station, on-orbit servicing, space debris removal etc.[1−4] The space robot is expected to reduce the danger of space servicing jobs and provide the convenience for space exploration.

Unlike terrestrial manipulator, the free-floating space robot is always free from the influence of external force. Due to the dynamics coupling of free-floating base and the manipulator, the attitude of base is disturbed by the movement of the manipulator, however, the attitude of the satellite is so important when considering solar supplement and information communication. The task of satellite attitude adjustment is typically achieved by the attitude control system of satellite, a way in which a limited amount of fuel will be consumed. Because the energy of the manipulator is from solar power, so the base attitude adjustment task implemented by the mounted manipulator is a better choice. And the coupling character between the base and the manipulator can be utilized to adjust and maintain the attitude[5, 6]. Furthermore, the additional degree of freedom (DOF) will be used to achieve end-effector task of the manipulator. In this paper, we solve this problem by the Cartesian path planning strategy.

In literature, Dubowsky and Torres planned the trajectory of space manipulator using enhanced disturbance map (EDM) to minimize the disturbance of the base attitude[7]. And a 2-DOF manipulator was taken for example, however, it was difficult to obtain the EDM of more DOF manipulator. Vafa and Dubowsky used Virtual manipulator model to develop path planning that reduced base disturbance, which was called the self-correcting path planning algorithm[8]. The nominal path was described in self-correcting motions. In this method, the space robot system is viewed as a linear system with the assumption that the movement of joint is small enough. Nakamura and Mukherjee[9] utilized Lyapunov function to achieve the regulation of both the satellite orientation and the manipulator joint angle simultaneously, this scheme was known as bi-directional approach. However, the stability of this method was not demonstrated strictly and the planned joint angles were not smooth. In Fernandes′s study, the authors proposed near-optimal nonholonomic motion planning to achieve attitude control inspired by the fact that a falling cat can change its orientation in midair and the algorithm was derived by this analogy[10], and a 3-DOF Puma manipulator mounted on the space platform was employed to test the nonholonomic motion planning algorithm. The inconvenience of Fernandes′s algorithm was that the Jacobian matrices should be derived by symbolic manipulation software. And a path planning method in Cartesian space was proposed by Papadopoulos which can both achieve base disturbance reducing and the avoidance of dynamic singularity. The nonholonomic redundancy character of space robot was also utilized in Xu′s study[11], the joint trajectory was parameterized by polynomial function, and off the shelf

Research Article
Manuscript received October 19, 2015; accepted July 8, 2016; published online January 18, 2017
This work was supported by National Program on Key Basic Research Project (973 Program, No. 2013CB733103) and the Program for New Century Excellent Talents in University (No. NCET-10-0058)
Recommended by Associate Editor Min Cheol Lee

optimization methods were invoked to solve the unknown parameters. This method could be applied to the target berthing and base reorientation after capturing.

In addition, Nenchev et al.[12] originally utilized the notation of reaction null-space (RNS) to achieve satellite attitude control. The RNS method was the only method that can achieve the base attitude regulation and end-effector trajectory planning simultaneously. In Nenchev′s study[13], the notation of RNS was also applied to achieve a vibration suppression task, a reactionless end-point control task and combined motion control task. And then it was utilized to achieve reactionless manipulation or zero reaction maneuver[14]. It was also verified in the flight experiments of ETS-VII. However, the base attitude regulation based on RNS has the problem of intrinsic dynamic singularity, To non-redundant space manipulator, the reasonable paths were limited. And to redundant robot, the extra degrees of freedom could be used to alleviate singularity issues[15].

An extension of the task-priority planning strategy [16, 17] of terrestrial robot is employed to cope with the base attitude adjustment issue in this article. And we also focus our attention on the dynamic singularity avoidance in the base attitude adjustment algorithm[18].

2 Preliminaries

2.1 Dynamics and kinematics model of space robot

Let us consider the linear and angular velocities of the base, $\dot{x}_b = (v_b^{\mathrm{T}}, w_b^{\mathrm{T}})^{\mathrm{T}} \in \mathbf{R}^{6\times 1}$, and the angular velocities of the joints, $\dot{\phi} \in \mathbf{R}^{n\times 1}$. The motion equation of space manipulator system is described in the following form:

$$\begin{pmatrix} H_b & H_{bm} \\ H_{bm}^{\mathrm{T}} & H_m \end{pmatrix}\begin{pmatrix} \ddot{x}_b \\ \ddot{\phi} \end{pmatrix} + \begin{pmatrix} c_b \\ c_m \end{pmatrix} = \begin{pmatrix} F_b \\ \tau \end{pmatrix} \tag{1}$$

where $H_b \in \mathbf{R}^{6\times 6}$: the inertia matrix of the satellite base, $H_{bm} \in \mathbf{R}^{6\times n}$: the coupled inertia matrix of the space robot. $c_b \in \mathbf{R}^6$: the velocity-dependent nonlinear term of the base, $c_m \in \mathbf{R}^n$: the velocity-dependent non-linear term of the manipulator. F_b: the driving force actin on the base. τ: the manipulator joint torque.

The motion of the space manipulator observes the law of conservation of momentum:

$$\sum_{i=0}^{n} m_i \dot{r}_i = const \tag{2}$$

$$\sum_{i=0}^{n} (I_i w_i + m_i r_i \times \dot{r}_i) = const \tag{3}$$

where m_i is the mass of body i, r_i is the vector from the inertia coordinate system to the center of the mass, I_i is the inertia matrix of body i.

From the rotational momentum conservation equation, the following expression can be obtained

$$I_S w_b + I_M \dot{\phi} = L_0 \tag{4}$$

where $I_S \in \mathbf{R}^{3\times 3}$ is the inertia matrix of the base, $I_M \in \mathbf{R}^{3\times n}$ is the inertia matrix of the manipulator. L_0 is initial momentum. Through (4), we can see the coupling relationship between the manipulator and the floating base.

Furthermore, denote by m the dimension of the end-effector task. We can get the following equation from the translational momentum conservation equation

$$\dot{x} = J_S w_b + J_M \dot{\phi} + \dot{x}_0 \tag{5}$$

where $\dot{x} \in \mathbf{R}^m$ is end-effector velocity vector, $\dot{x}_0 \in \mathbf{R}^m$ is initial rate vector of end-effector. $J_s \in \mathbf{R}^{m\times 3}$ and $J_M \in \mathbf{R}^{m\times n}$ are the Jacobian matrices.

From (4) we have

$$w_b = -I_S^{-1} I_M \dot{\phi} + I_S^{-1} L_0. \tag{6}$$

Substitute (6) in (5) to obtain

$$\dot{x} = (J_M - J_S I_S^{-1} I_M)\dot{\phi} + \dot{\hat{x}}_0 = J_G \dot{\phi} + \dot{\hat{x}}_0 \tag{7}$$

where $\dot{\hat{x}}_0 = \dot{x}_0 + J_S I_S^{-1} L_0$ and $J_G = J_M - J_S I_S^{-1} I_M$ is the generalized Jacobian matrix.

2.2 Reaction null-space based path planning of space robot

For redundant space robot, redundancy is determined by the DOF of the manipulator and the number of end-effector tasks combined with base task. m and l denote the number of task variables for end-effector task and base task, respectively.

From (4), the least-squares minimun-norm (LM) solution based on the Jacobian pseudoinverse for the base attitude adjustment is presented as

$$\dot{\phi}_{w\_LM} = I_M^{+} \left(L_0 - I_S w_b\right). \tag{8}$$

As $n > l$, $n - l$ DOF exists, the non-minimun-norm solutions to (4) based on the Jacobian pseudoinverse can be written in the general form

$$\dot{\phi} = \dot{\phi}_{w\_LM} + \dot{\phi}_{w\_null} = I_M^{+} \left(L_0 - I_S w_b\right) + \left(I - I_M^{+} I_M\right)\dot{\zeta} \tag{9}$$

where I denotes the identity matrix. $(\cdot)^+$ is the Moore-Penrose pseudoinverse of matrix $(\cdot)^+$, the obtained joints angle velocities $\dot{\phi}$ are the minimum-norm solution. And $\left(I - (\cdot)^+ (\cdot)\right)$: the projection onto the null space of matrix $(\cdot)$, $\dot{\zeta}$ is arbitrary vector.

In (9), the coupling inertia matrix from rotational momentum conservation equation comprises a non-trivial kernel.

And $N = \left\{\dot{\phi}_{w\_null}: \dot{\phi}_{w\_null} = \left(I - I_M^{+} I_M\right)\zeta, \forall \dot{\zeta}\right\}$ is the set of reactionless joint velocities. Because $\dot{\phi} \in N$, the manipulator joint motion is dynamically decoupled from the motion of satellite base. This reactionless kernel is well known as reaction null-space.

Substitute (9) into (5) to obtain

$$\left(J_S - J_M I_M^{+} I_S\right) w_b + J_M \left(I - I_M^{+} I_M\right)\dot{\zeta} = \dot{x} - \dot{\hat{x}}_0 \tag{10}$$

where $\dot{\tilde{x}}_0 = \dot{x}_0 + J_M I_M^+ L_0$, the arbitrary vector ζ in null space is presented as

$$\dot{\zeta}=\left[J_M\left(I-I_M^+ I_M\right)\right]^+\left[\dot{x}-\dot{\tilde{x}}_0-\left(J_S-J_M I_M^+ I_S\right)w_b\right]. \quad (11)$$

Furthermore, substitute (11) into (9) to obtain

$$\begin{aligned}\dot{\phi}=&I_M^+\left(L_0-I_S w_b\right)+\\&\left[J_M\left(I-I_M^+ I_M\right)\right]^+\left[\dot{x}-\dot{\tilde{x}}_0-\left(J_S-J_M I_M^+ I_S\right)w_b\right]\end{aligned} \quad (12)$$

where $P(J_M P)^+$ can be simplified as $(J_M P)^+$, as $P=\left(I-I_M^+ I_M\right)$ is a symmetrical and idempotent projectional matrix.

Furthermore, assume the initial system momentum L_0 and initial end-effector velocity $\dot{x}_0$ are zero, and the desired angular velocity of the satellite w_b is zero, then (9) can be written as

$$\dot{\phi}=\left[J_M\left(I-I_M^+ I_M\right)\right]^+ \dot{x}_{FAR}. \quad (13)$$

And we call the matrix $J_M\left(I-I_M^+ I_M\right)$ the fixed attitude restricted Jacobian matrix. If the joint movement follows ϕ, then there will be no disturbance to the satellite attitude and the end-effector will move in agreement with $\dot{x}_{FAR}$.

In (12), the satellite attitude control is the primary priority task, whereas, the end-effector task is a subtask with the second priority. And the lower priority task will not affect the higher priority task. The first and second priority task will have no residual error. However, there are dynamic singularity and algorithmic singularity in RNS method, while the former occurs in the case of either:

$$\rho\left(I_M\right)<m \quad \text{or } \rho\left(J_M\right)<l$$

$$\rho\left(J_M\left(I-I_M^+ I_M\right)\right)=\rho\left(\begin{bmatrix} J_M \\ I_M \end{bmatrix}\right)$$

where $\rho(\cdot)$ denotes the rank of a matrix.

2.3 Extended Jacobian based path planning of space robot

When the initial state of the space robot is still, so $L_0 = 0$ and $\dot{x}_0 = 0$. From (4) and (5) we can get the following equations respectively

$$w_b = -I_S^{-1} I_M \dot{\phi} \quad (14)$$

$$\dot{x} = \left(J_M - J_S I_S^{-1} I_M\right)\dot{\phi}. \quad (15)$$

From (14) and (15) we can get a combined motion equation as

$$\begin{pmatrix} w_b \\ \dot{x} \end{pmatrix}=\begin{bmatrix} -I_S^{-1} I_M \\ \left(J_M - J_S I_S^{-1} I_M\right) \end{bmatrix}\dot{\phi} = J_E \dot{\phi} \quad (16)$$

where J_E is the extended Jacobian.

Furthermore, we can obtain $\dot{\phi}$ using (16)

$$\dot{\phi}=\begin{bmatrix} -I_S^{-1} I_M \\ \left(J_M - J_S I_S^{-1} I_M\right) \end{bmatrix}^*\begin{pmatrix} w_b \\ \dot{x} \end{pmatrix}=J_E^*\begin{pmatrix} w_b \\ \dot{x} \end{pmatrix} \quad (17)$$

where $J_E^* = J_E^{-1}$ when $3 + m = n$ and $J_E^* = J_E^+$, when $3 + m < n$ ($J_E^* \in \mathbf{R}^{n\times(3+m)}$). And we call this extended Jacobian (EJ) algorithm.

To define the relative priority between satellite task and end-effector task, two positive weighting coefficients λ_b and λ_x can be introduced. Then, we can obtain

$$\dot{\phi} = \begin{bmatrix} -\lambda_b I_S^{-1} I_M \\ \lambda_x\left(J_M - J_S I_S^{-1} I_M\right) \end{bmatrix}^*\begin{pmatrix} \lambda_b w_b \\ \lambda_x \dot{x} \end{pmatrix}=J_E^{\lambda *}\begin{pmatrix} \lambda_b w_b \\ \lambda_x \dot{x} \end{pmatrix}. \quad (18)$$

Let us consider the manipulation with zero disturbance to the base by setting $w_b = 0$, then the momentum conservation (4) can be revised as

$$I_M \dot{\phi} = 0. \quad (19)$$

And the above equation combined with (15) will derive zero reaction maneuver path:

$$\dot{\phi}=\begin{bmatrix} -I_M \\ \left(J_M - J_S I_S{}^{-1} I_M\right) \end{bmatrix}^+\begin{pmatrix} 0 \\ \dot{x} \end{pmatrix}. \quad (20)$$

The above solution is also belonging to reaction null-space path planning, but the other $n-3$ DOF is utilized to achieve end-effector task.

3 Singularity robust path planning algorithm for base attitude adjustment

3.1 A path planning algorithm without algorithmic singularity

The algorithmic singularity which occurs in the RNS method is artificial. And through (11), it is obvious that the algorithmic singularity is due to the selection of arbitrary vector $\dot{\zeta}$, so we can avoid the algorithmic singularity by choosing proper null-space velocity to satisfy the secondary task.

Let us consider (7) again, the equation can be rewritten as

$$\dot{x}-\dot{x}_0-J_S w_b=J_M I_M^+\left(L_0-I_S w_b\right)+J_M\left(I-I_M^+ I_M\right)\dot{\zeta}. \quad (21)$$

We can know that the above equation can be achieved if the following condition is satisfied

$$\dot{x}_C \in R\left(J_M\right) \quad (22)$$

where $\dot{x}_C=\dot{x}-\dot{x}_0-J_S w_b$, $R\left(J_M\right)$ denotes the range space of J_M. This means arbitrary vector stands for the motion in null space and it can be derived by the solution of the following

$$J_M J_M^+ \dot{x}_C=J_M I_M^+\left(L_0-I_S w_b\right)+J_M\left(I-I_M^+ I_M\right)\dot{\zeta}. \quad (23)$$

Then, we can obtain

$$J_M^+ \dot{x}_C=I_M^+\left(L_0-I_S w_b\right)+\left(I-I_M^+ I_M\right)\dot{\zeta}. \quad (24)$$

This means that we search for the null-space arbitrary vector to achieve the equivalence between the non-minimun-norm solutions of (4) and least-squares minimun-norm solution of (5). The $\dot{\zeta}$ is selected to satisfy the following equation as accurately as possible

$$\dot{\phi}_{x\_LM}=\dot{\phi}_{w\_LM}+\dot{\phi}_{w\_null} \tag{25}$$

where $\dot{\phi}_{x\_LM}=J_M^+\dot{x}_C$, is the least-squares minimun-norm solution of (5).

Furthermore, $\dot{\zeta}$ can be given

$$\dot{\zeta}=\left(I-I_M^+I_M\right)^+\left(J_M^+\dot{x}_C-I_M^+\left(L_0-I_Sw_b\right)\right). \tag{26}$$

By recalling $\left(I-I_M^+I_M\right)^+=I-I_M^+I_M$ and $I_M^+I_MI_M^+=I_M^+$ (26) can be simplified as

$$\dot{\zeta}=\left(I-I_M^+I_M\right)J_M^+\dot{x}_C. \tag{27}$$

Substituting (27) into (9) and recalling the idempotence of $\left(I-I_M^+I_M\right)$ we can obtain

$$\dot{\phi}=I_M^+\left(L_0-I_Sw_b\right)+\left(I-I_M^+I_M\right)J_M^+\dot{x}_C. \tag{28}$$

In above equation, the satellite attitude control is higher priority task, whereas, the end-effector task is a subtask with the second priority. The implementation of the above equation has no algorithmic singularity. We call it task-priority based reaction null-space algorithm (TP-RNS).

The priority level of the two tasks can be exchanged, that is to say, the end-effector task is primary while the base task is secondary. And following the analogous math deduction, we can get

$$\dot{\phi}=J_M^+\dot{x}_C+\left(I-J_M^+J_M\right)I_M^+\left(L_0-I_Sw_b\right). \tag{29}$$

Furthermore, suppose the initial system momentum L_0 and initial end-effector velocity x_0 are zero, and the desired angular velocity of the satellite w_b is zero too, then (28) can be written as

$$\dot{\phi}=\left(I-I_M^+I_M\right)J_M^+\dot{x}_{\rm FAR}. \tag{30}$$

In the above (28), the joint velocities associated with the respective desired task velocities are solved by the pesudoinverses I_M^+ and J_M^+. And the above solution is equivalent to solution (12) when the end-effector task is compatible to the satellite base related task if and only if the following equation is satisfied

$$J_MI_M^+\left(L_0-I_Sw_b\right)=0\ \ \forall\, L_0-I_Sw_b \text{ that is } J_MI_M^+=0.$$

And solution (28) is equivalent to solution of (12) when the end-effector task is not compatible to the satellite base related task for the case

$$\left(I-I_M^+I_M\right)J_M^+=0.$$

So in this case, there is no additional DOF to achieve end-effector Cartesian path tracking task, and both solutions reduce to (8).

3.2 Dynamic singularity avoidance

Equations (28) and (30) are the other algorithms to solve two priority control of space redundant manipulator but without algorithmic singularity. However, the algorithms still have dynamic singularity, and it is intrinsic character of space robot which cannot be avoided by the reconfiguration of algorithms. To better know the notation of dynamic singularity, consider the singular value decomposition (SVD) of I_M and the analysis of J_M is the same as I_M

$$I_M=U\Sigma V^{\rm T}=\sum_{i=1}^{l}\sigma_iu_iv_i^{\rm T} \tag{31}$$

where u_i,v_i are the columns of U and V. From (31) we can see there is a discontinuity on the pesudoinverse operator around the singular configuration. When coming across dynamic singularity, the discontinuity causes large joint velocities with high conditioning. The condition number of I_M can be defined as

$$\text{cond}\left(I_M\right)=\frac{\sigma_{\max}\left(I_M\right)}{\sigma_{\min}\left(I_M\right)}. \tag{32}$$

To avoid the discontinuity, some methods are proposed. The singular value filtering (SVF) aims to modify the Jacobian singular value to obtain an alternative pseudoinverse with full-rank and bounded condition number[19].

In SVF, the Jacobian matrix′s singular values are modified to get the following transformation

$$\hat{I}_M=U\hat{\Sigma}V^{\rm T}=\sum_{i=1}^{l}f_{\upsilon,\sigma_{0\_I}}\left(\sigma_i\right)u_iv_i^{\rm T} \tag{33}$$

and

$$f_{\upsilon,\sigma_{0\_I}}\left(\sigma_i\right)=\frac{\sigma_i^3+\upsilon\sigma_i^2+2\sigma_i+2\sigma_{0\_I}}{\sigma_i^2+\upsilon\sigma_i+2}$$

where σ_{0_I} is the constraint bound value for the minimum singularity value, υ is a shape factor. So the pseudoinverse of the Jacobian I_M is

$$\hat{I}_M^{\#}=\sum_{i=1}^{l}\frac{v_iu_i^{\rm T}}{f_{\upsilon,\sigma_{0\_I}}\left(\sigma\right)}=\sum_{i=1}^{l}\frac{\sigma_i^2+\upsilon\sigma_i+2}{\sigma_i^3+\upsilon\sigma_i^2+2\sigma_i+2\sigma_{0\_I}}v_iu_i^{\rm T}. \tag{34}$$

The condition number of $\widehat{I}_M$ is

$$\text{cond}\left(\hat{I}_M\right)=\frac{\left(\sigma_1^3+\upsilon\sigma_1^2+2\sigma_1+2\sigma_{0\_I}\right)\left(\sigma_l^2+\upsilon\sigma_l+2\right)}{\left(\sigma_1^2+\upsilon\sigma_1+2\right)\left(\sigma_l^3+\upsilon\sigma_l^2+2\sigma_l+2\sigma_{0\_I}\right)} \tag{35}$$

and $\sigma_i\rightarrow 0,\frac{1}{f_{\upsilon,\sigma_0}(\sigma_i)}\rightarrow\frac{1}{\sigma_0}$ and $\text{cond}\left(I_M\right)=const$ and when $\sigma_i\rightarrow\infty,\frac{1}{f_{\upsilon,\sigma_0}(\sigma_i)}\rightarrow\frac{1}{\sigma_i}$.

So the dynamic singularity can be avoided and the condition number is bounded.

3.3 Representation of singularity avoidance

In practical application, if the orientation is represented by Euler angles $\psi=[\alpha\,\beta\,\gamma]$, then the relationship between angular velocity w and rotational velocity $\dot{\psi}$ is represented as

$$w=\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}=R\dot{\psi}=\begin{bmatrix} \cos\alpha\cos\beta & -\sin\alpha & 0 \\ \sin\alpha\cos\beta & \cos\alpha & 0 \\ -\sin\beta & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}. \tag{36}$$

And w can be derived from forward kinematics or non-holonomic constraint in velocity level. So the orientation velocity is solved by the inverse operation of transformation $\mathbf{R}$. But $\mathbf{R}$ is not invertible for $\beta=\pm 90°$. We call this orientation singularity. A numerically robust solution to this problem can be formulated with the help of unit quaternion. The unit quaternion can be defined as

$$q=\begin{bmatrix} \eta & \varepsilon \end{bmatrix}=\begin{bmatrix} \eta & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}\in\mathbf{R}^4 \tag{37}$$

where η is the scalar part and ε is the vector part, and they satisfy $\eta^2+\varepsilon_1^2+\varepsilon_2^2+\varepsilon_3^2=1$.

The relationship between the time derivative of the quaternion $\dot{q}$ and angular velocity w is

$$\begin{bmatrix} \dot{\eta} \\ \dot{\varepsilon} \end{bmatrix}=\frac{1}{2}\begin{bmatrix} -\varepsilon^{\mathrm{T}} \\ \eta I-\tilde{\varepsilon} \end{bmatrix}w. \tag{38}$$

The integration of $\dot{q}$ can be utilized to calculate the Jacobian matrix in the path planning and control algorithm.

So the Euler angle is used to display attitude and the unit quaternion is utilized for numerical calculation.

3.4 Singularity robust implementation of path planning algorithm

To handle singularity, the inverse can be replaced by the singularity robust inverse, denoted by the superscript #, in this paper, the novel pseudoinverse can be calculated by the way of singular value filtering. So the singularity robust solution of (12), (18) and (28) are as follows

$$\begin{aligned}\dot{\phi}_{DLS\_RNS}=&\hat{I}_M^{\#}\left(L_0-I_S w_b\right)+\\ &\left[J_M\left(I-\hat{I}_M^{\#}I_M\right)\right]^{\#}\left[\dot{x}-\dot{\tilde{x}}_0-\left(J_S-J_M\hat{I}_M^{\#}I_S\right)w_b\right]\end{aligned} \tag{39}$$

$$\dot{\phi}_{DLS\_EJ}=\begin{bmatrix} -\lambda_b {I_S}^{-1} I_M \\ \lambda_x\left(J_M-J_S {I_S}^{-1} I_M\right) \end{bmatrix}^{\#}\begin{pmatrix} w_b \\ \dot{x} \end{pmatrix}=J_E^{\lambda\#}\begin{pmatrix} w_b \\ \dot{x} \end{pmatrix} \tag{40}$$

$$\dot{\phi}_{DLS\_TP\_RNS}=\hat{I}_M^{\#}\left(L_0-I_S w_b\right)+\left(I-\hat{I}_M^{\#}I_M\right)\hat{J}_M^{\#}\dot{x}_C. \tag{41}$$

Take (41) for example, and let us look at the residual error of the algorithm. Combining (41) and (4) we can deduce

$$\begin{aligned}R_1=&I_S w_b-L_0+I_M\dot{\phi}_{DLS\_TP\_RNS}=\\ &I_S w_b-L_0+I_M\left(\hat{I}_M^{\#}\left(L_0-I_S w_b\right)+\right.\\ &\left.\left(I-\hat{I}_M^{\#}I_M\right)\hat{J}_M^{\#}\dot{x}_C\right)\end{aligned} \tag{42}$$

which stands for the residual error of primary task. And $R_1=0$, if $\sigma_i << \sigma_{0\_I}$, $i=1\cdots n$. And the singularity occurring in the execution of secondary task will not effect the implementation of the primary task. To some degree, semi-singularity-separation can be achieved.

Combining (42), (41) and (4) we obtain

$$\begin{aligned}R_2=&\dot{x}-J_S w_b-J_M\dot{\phi}_{DLS\_TP\_RNS}-\dot{x}_0=\\ &\left(I-J_M\hat{J}_M^{\#}\right)\dot{x}_C-J_M\hat{I}_M^{\#}\left(L_0-I_S w_b-I_M\hat{J}_M^{\#}\dot{x}_C\right)\end{aligned} \tag{43}$$

which stands for the residual error of secondary task. And $R_2=0,\ if\ \sigma_j << \sigma_{0J}\ \cap\ J_M I_M^{\#}=0,\ j=1\cdots n.$

Fig. 1 illustrates the proposed algorithm implementation in the control scheme of the space robot. The control scheme consists of task planning part, central controller, joint controller, forward dynamics and forward kinematics. The proposed algorithm works in the central controller as shown in the following.

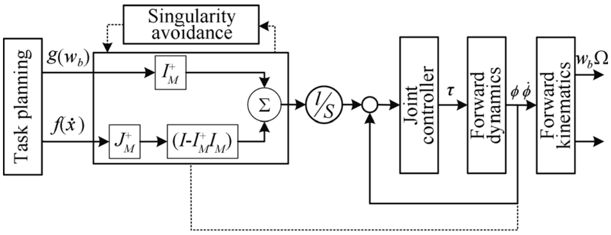


Fig. 1 Algorithm implementation in the control scheme of space robot

4 Algorithm verification in real time simulation system

4.1 Real time simulator under Linux/RTAI

To space robot, the ground hardware-based verification methods like air-bearing table[20], airplane flying or free-falling motion[21] and suspension system[22] are adopted based on the analog of no gravity. Recently, a kind of hybrid simulation method following the principle of dynamics simulation and kinematics equivalence has been presented, which is called hardware-in-the-loop simulation system[23]. The above verification systems are built costly. In this article, we build a real time simulation system of a redundant space robot under Linux/RTAI to explore the capabilities and limitations of the proposed method. This real time numerical simulator is conceived and realized as an aid to the design of the controller of a new space robot in development[24].

Taking a seven DOF space manipulator for example, the body fixed frame is shown in Fig. 2. And the kinematics and dynamics parameters are listed as Table 1.

The real-time verification system runs under the Linux operating system with the real time application interface (RTAI) extension. And RTAI is a hard real time extension of Linux. It provides task scheduling and synchronization and inter-task communication services among others. The

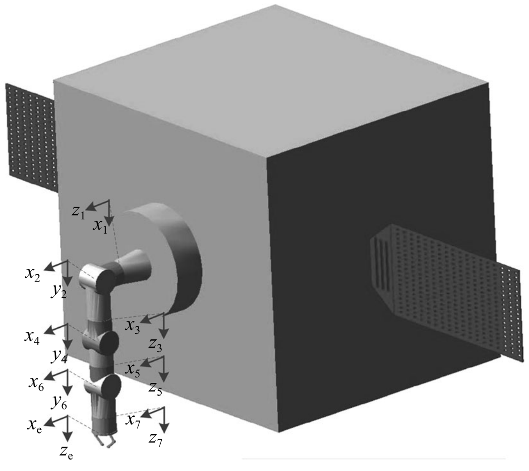


Fig. 2 Body fixed frame of space robot

Table 1 Mass properties of the space manipulator system

	Mass (kg)	Origin position (mm)	Inertia (kg·m^2)		
			I_{xx}	I_{yy}	I_{zz}
Base	2 000	[0 0 3 000]	100	100	100
Link1	5	[0 0 150]	0.15	0.15	0.08
Link2	18	[0 250 0]	0.5	0.2	0.5
Link3	10	[0 0 150]	0.35	0.35	0.15
Link4	12	[0 150 0]	0.40	0.15	0.40
Link5	8.6	[0 0 135]	0.38	0.38	0.16
Link6	16	[0 200 0]	0.40	0.16	0.40
Link7	10	[0 0 100]	0.23	0.23	0.17

functional architecture of the simulator is shown as Fig. 3. The real time simulation system consists of 4 modules:

1) Forward dynamics simulator of space robot

The dynamic model of space robot built offline in the SimMechanics and the MATLAB RTW (real-time workshop) is utilized to port the virtual prototype model into C codes for online calculation of the dynamic model.

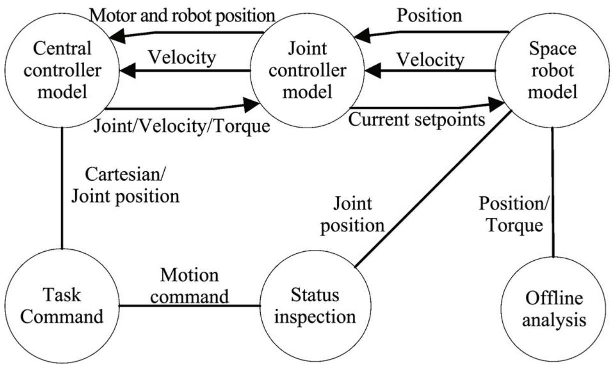


Fig. 3 Functional architecture of the real time simulator

2) Simulator of the controller

The controller of the real space robot system consists of the central controller and joint controller. The central controller is responsible for the online vision based autonomous planning, Cartesian path planning, joint space path planning, computed torque control and so on. The functional diagram of central controller simulator is shown as Fig. 4. While the simulator of joint controller is used to simulate the servo system of the space robot, and its functional diagram is shown as Fig. 5.

3) Task scheduling and synchronization

The modules mentioned above are made into three real-time tasks and run in three independent threads. Another thread for scheduling the three tasks which is called clock task is created to make them run periodically. The synchronous execution of the tasks is achieved by a blocking inter-process/thread communication semaphore.

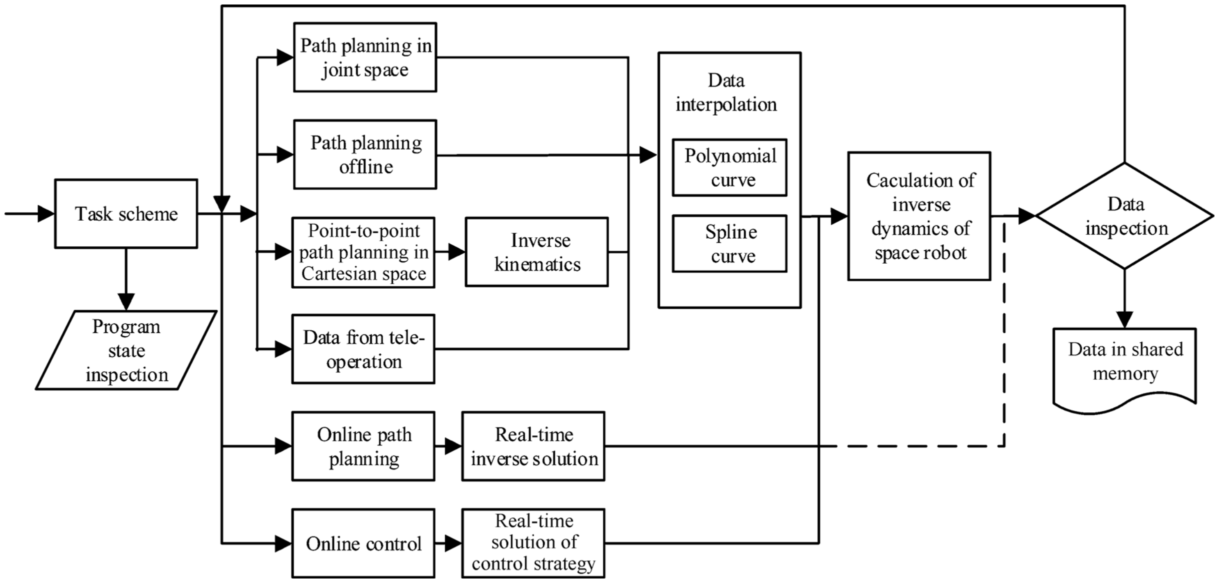


Fig. 4 Functional diagram of central controller simulator

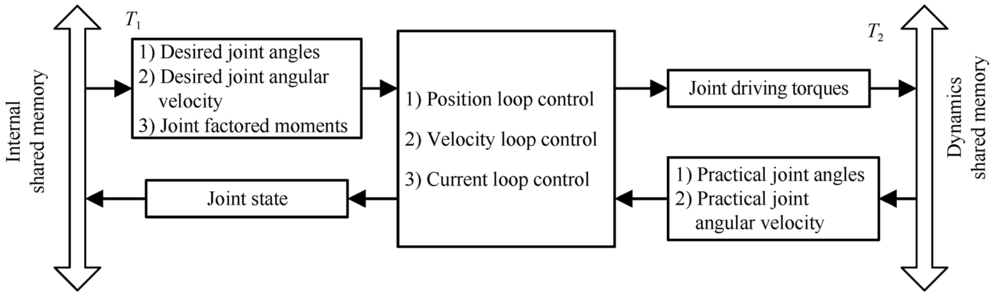


Fig. 5 Functional diagram of joint controller simulator

When one executes its period and writes data into shared memory, the semaphore value is set to one and the next is allowed to run. The shared memory in this simulator worked as a virtual bus.

4) Monitor

The monitor part runs in another Windows PC installed with Labview which can modify control parameters online and save data to offline analyze furthermore. It is connected with the main simulator by TCP/IP communication. The control parameters could be modified and sent back to the model so as to be easily adjusted.

To sum up, the real time simulator of free-floating space robot is built through the analogy with the real control system of the space robot in development. And its communication architecture of the simulator is shown as Fig. 6. This simulator can be used to verify and develop the path planning algorithm, motion control algorithm, dynamics based control algorithm and respond to offline database. And codes of different parts in the simulator can be transformed into the real controller easily.

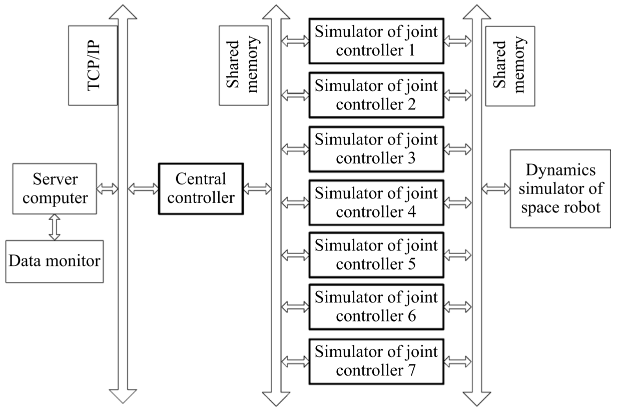

TCP/IP
Shared memory
Simulator of joint controller 1
Simulator of joint controller 2
Simulator of joint controller 3
Simulator of joint controller 4
Simulator of joint controller 5
Simulator of joint controller 6
Simulator of joint controller 7
Shared memory
Server computer
Central controller
Data monitor
Dynamics simulator of space robot

Fig. 6 Communication architecture of the real time simulator

4.2 Experiment results

The performance of TP-RNS algorithm proposed in this article, EJ algorithm and RNS algorithm are illustrated by three simulation examples. Assume the initial state of the space robot is still and the initial momentum is zero. The initial joint angles of space robot are (0 −90 0 0 0 0 0) degree and the base attitude Euler angle is (0, 0, 0) degree. Furthermore, we adopt trapezium trajectories to plan the velocity and angular velocity. And the cycle of the control system is 0.01s.

Example 1. The primary task is to execute base attitude adjustment, while the secondary is to implement the end-effector path tracking task. In this example, the desired end-effector displacement is (−35 −105 70) mm, that is from the initial position (985 0 1 650) mm to (950 105 1 720) mm , the desired base attitude is (3, 3, 0) degree. The simulation results are shown as Figs. 7 and 8.

In above, we adopt σ_{0_I}=σ_{0_J}= 0.000 000 5, υ=15. From the above results, the curves of real pose and desired pose are almost coincidental, that is to say, the 3 algorithms can perform the desired tasks. And we can see the attitude control task errors in RNS are smaller than that in EJ, while bigger than that in TP-RNS. However, the end-effector task errors in EJ algorithm are bigger than that in TP-RNS, while smaller than that in RNS. This is because, when algorithmic singularity occurs, the singularity avoidance will sacrifice accuracy in all directions, while the other two will ensure the primary task. In the meantime, the end-effector task accuracy will be sacrificed. And the semi-singularity separation will also ensure the accuracy in TP-RNS is higher than that in RNS. In this case the desired two tasks are almost compatible to some degree, and the contradiction can be eliminated by the avoidance of dynamic singularity.

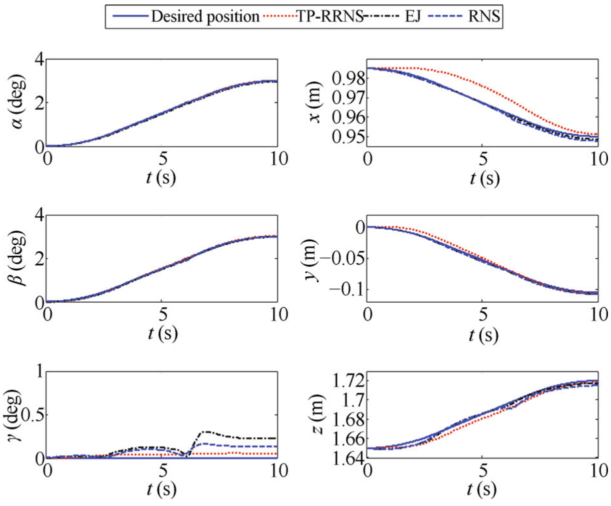


Fig. 7 Attitude of the base and the position of the end-effector

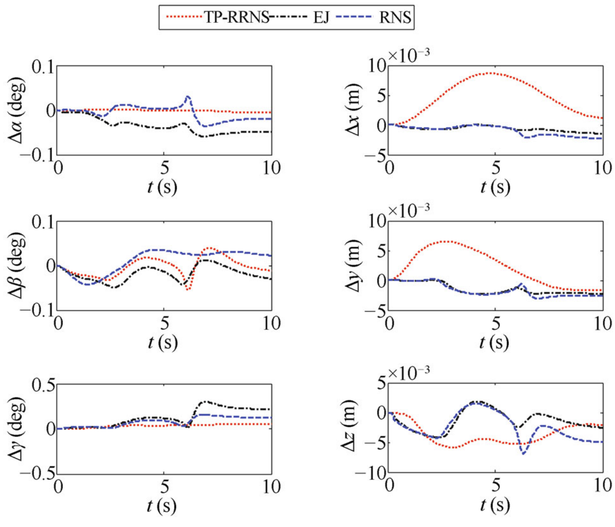


Fig. 8 Tracking errors of both tasks

Example 2. The desired end-effector displacement is (0 56 −3.5) mm, the desired attitude of the satellite is (3, 3, 0) degree. And we also adopt $\sigma_{0\_I}=\sigma_{0\_J}=0.000\,000\,5$, $\upsilon=15$ just the same as that in example one. The simulation results are shown as Fig. 9.

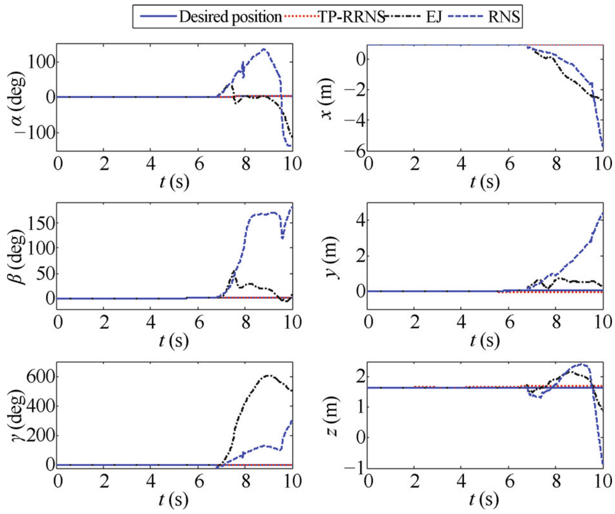


Fig. 9 Attitude of the base and the position of the end-effector

From Fig. 9, the RNS algorithm and EJ algorithm has failed to accomplish the desired task. The TP-RNS algorithm is more stable than RNS algorithm and EJ algorithm. This is because the two tasks in this example are fully contradictory, that is to say, they cannot be implemented simultaneously. While in TP-RNS algorithm proposed in this paper, we can ensure the accomplishment of the attitude control task, however, regardless of the accuracy of the lower priority task.

Example 3. The both sets of the above simulation report the comparison of the three algorithms for base attitude adjustment. Furthermore, when we set the desired angular velocity of satellite base to be zero, the described three algorithms can achieve reactionless manipulation. This base attitude maintenance manipulation is very important in the path planning and control of space robot. The desired end-effector displacement is (−21 0 −70) mm. And we adopt $\sigma_{0\_I}=\sigma_{0\_J}=0.000\,000\,5$, $\upsilon=15$ as example one. The simulation results about two tasks are shown as Fig. 10, Fig. 11 shows the errors of both tasks.

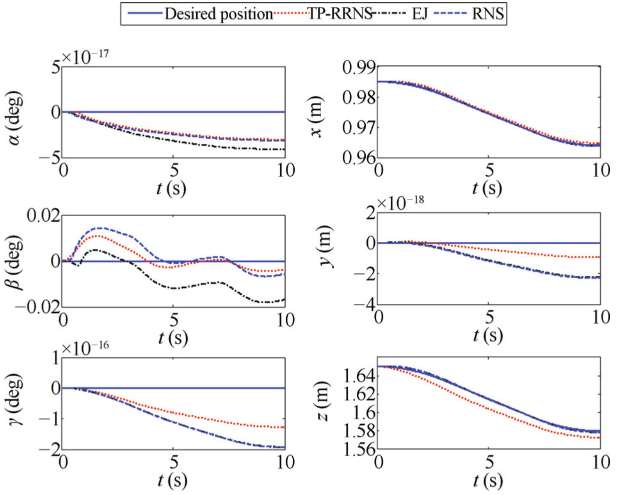


Fig. 10 Attitude of the base and the position of the end-effector

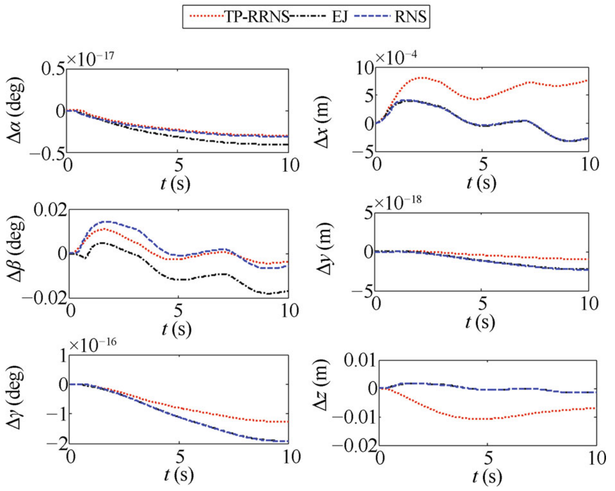


Fig. 11 Tracking errors of both tasks

The above results show that the described methods can achieve reactionless manipulation. And the errors in attitude in TP-RNS are smaller than those in other two methods.

5 Conclusions

In this paper, we proposed a task priority based reaction null-space algorithm to deal with the base attitude adjustment of space robot. Based on the characteristics of task priority, the Cartesian path planning based on TP-RNS will ensure no algorithmic singularity and the inaccuracy of lower priority task will not affect the accomplishment of the higher priority task. The proposed method is computationally less demanding and more robust to occurrence of algorithmic singularities caused by the contradictory task situations. And the singular value filtering decomposition is introduced to dispose the dynamic singularity.

In addition, a real time simulator under Linux/RTAI of free-floating space robot is built in this paper to verify the online calculation of the proposed method. Furthermore, the related comparisons about the three methods are made. Thus, real time base attitude adjustment task is implemented in the simulator.

References

[1] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[2] W. Xu, B. Liang, Y. Xu. Survey of modeling, planning, and ground verification of space robotic systems. *Acta Astronautica*, vol. 68, no. 11–12, pp. 1629–1649, 2011.

[3] S. A. A. Moosavian, E. Papadopoulos. Free-flying robots in space: An overview of dynamics modeling, planning and control. *Robotica*, vol. 25, no. 5, pp. 537–547, 2007.

[4] F. C. Liu, L H. Liang, J. J. Gao. Fuzzy PID control of space manipulator for both ground alignment and space applications. *International Journal of Automation and Computing*, vol. 11, no. 4, pp. 353–360, 2014.

[5] S. Dubowsky, E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 531–543, 1993.

[6] C. Fernandes, L. Gurvits, Z. X. Li. Attitude control of space platform/manipulator system using internal motion. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Nice, France, vol. 1, pp. 893–8981, 1992.

[7] S. Dubowsky, M. A. Torres. Path planning for space manipulators to minimize spacecraft attitude disturbances. In *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, Sacramento, USA, vol. 3, pp. 2522–2528, 1991.

[8] Z. Vafa, S. Dubowsky. On the dynamics of space manipulators using the Virtual Manipulator, with applications to path planning. *Journal of the Astronautical Sciences*, vol. 38, no. 4, pp. 441–472, 1990.

[9] Y. Nakamura, R. Mukherjee. Nonholonomic Motion Planning of free-flying space robots via a bi-directional approach. *Space Robotics: Dynamics and Control*, New York, USA: Springer, 1993.

[10] C. Fernandes, L. Gurvits, Z. X. Li. Near-optimal nonholonomic motion planning for a system of coupled rigid bodies. *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 450–463, 1994.

[11] W. F. Xu, C. Li, B. Liang, Y. S. Xu, Y. Li. Liu, W. Y. Qiang. Target berthing and base reorientation of free-floating space robotic system after capturing. *Acta Astronautica*, vol. 64, no. 2–3, pp. 109–126, 2009.

[12] D. Nenchev, Y. Umetani, K. Yoshida. Analysis of a redundant free-flying spacecraft/manipulator system. *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 1–6, 1992.

[13] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, M. Uchiyama. Reaction null-space control of flexible structure mounted manipulator systems. *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 1011–1023, 1999.

[14] D. N. Nenchev. Reaction null space of a multibody system with applications in robotics. *Mechanical Sciences*, vol. 4, no. 1, pp. 97–112, 2013.

[15] T. C. Nguyen-Huynh, I. Sharf. Adaptive Reactionless motion with joint limit avoidance for robotic capture of unknown target in space. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Vilamoura, Ecuador, pp. 1155–1160, 2012.

[16] Y. Nakamura, H. Hanafusa, T. Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[17] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[18] M. H. Jin, C. Zhou, Y. C. Liu, H. Liu. Cartesian path planning for base attitude adjustment of space robot. In *Proceedings of 2015 IEEE International Conference on Mechatronics and Automation*, IEEE, Beijing, China, pp. 582–587, 2015.

[19] A. Colome, C. Torras. Redundant inverse kinematics: Experimental comparative review and two enhancements. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Vilamoura, Ecuador, pp. 5333–5340, 2012.

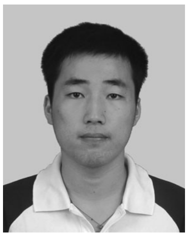
[20] Y. Umetani, K. Yoshida. Experimental study on two-dimensional free-flying robot satellite model. In *Proceedings of the NASA Conference on Space Telerobotics*, California Institute of Technology, Tokyo, Japan, vol. 5, pp. 215–224, 1989.

[21] C. Menon, A. Aboudan, S. Cocuzza, A. Bulgarelli, F. Angrilli. Free-flying robot tested on parabolic flights: Kinematic control. *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 623–630, 2005.

[22] H. Fujii, K. Uchiyama, H. Yoneoka, T. Maruyama. Ground-based simulation of space manipulators using test bed with suspension system. *Journal of guidance, control, and dynamics*, vol. 19, no. 5, pp. 985–991, 1996.

[23] R. Krenn, B. Schaefer. Limitations of hardware-in-the-loop simulations of space robotics dynamics using industrial robots. In *Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, ESA Publication Division, Noordwijk, The Netherland, pp. 681–686, 1999.

[24] G. Ferretti, G. Magnani, P. Porrati, G. Rizzi, P. Rocco, A. Rusconi. Real-time simulation of a space robotic arm. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.



Cheng Zhou received the B. Sc. and M. Sc. degrees in mechanical engineering from the Harbin Institute of Technology, China in 2012 and 2014 respectively. He is currently a Ph. D. degree candidate in the School of Mechatronic Engineering at Harbin Institute of Technology, China. He received the Best Paper Finalist in *2015 IEEE International Conference on Mechanics and Automation*.

His research interests include dynamics modeling, path planning and control of space robot.

E-mail: zhoucheng0818@sina.com (Corresponding author)

ORCID iD: 0000-0003-3412-854X

Ming-He Jin received the B. Sc., M. Sc. and Ph. D. degrees in mechanical engineering from the Harbin Institute of Technology, China in 1993, 1996 and 2000, respectively. He is currently with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. Currently, he is a professor in the School of Mechatronic Engineering at Harbin Institute of Technology, China.

His research interests include dexterous hand and space robot.

E-mail: mhjin@hit.edu.cn

Ye-Chao Liu received the B. Sc., M. Sc. and Ph. D. degrees in mechanical engineering from the Harbin Institute of Technology, China in 2002, 2004 and 2009, respectively. He is currently with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. He is currently an associate professor in the School of Mechatronic Engineering at Harbin Institute of Technology, China.

His research interests include impedance control of manipulator and space robot.

E-mail: yechao_hitdlr@aliyun.com

Ze Zhang received the B. Sc. degree in mechanical engineering from the Harbin Institute of Technology, China in 2014. He is currently a master student in the School of Mechatronic Engineering at Harbin Institute of Technology, China.

His research interests include control system development of manipulator.

E-mail: yanheyan123@sina.com

Yu Liu received the M. Sc. and Ph. D. degrees in mechanical engineering from the Harbin Institute of Technology, China in 2000 and 2003, respectively. He is currently with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, China. He is currently an associate professor in the School of Mechatronic Engineering at Harbin Institute of Technology, China.

His research interests include path planning and calibration of the robot.

E-mail: lyu11@hit.edu.cn

Hong Liu received the Ph. D. degree in mechanical engineering from the Harbin Institute of Technology, China in 1993. He was a jointly trained Ph. D. scholar with the Institute of Robotics and System Dynamics, German Aerospace Research Establishment, Wessling, Germany, from 1991 to 1993, where he has been a research fellow since 1993. He is currently with the State Key Laboratory of Robotics and System, Harbin Institute of Technology. He is currently a professor in the School of Mechatronic Engineering at Harbin Institute of Technology, China.

His research interests include dexterous hand and space robot.

E-mail: hong.liu@hit.edu.cn