Front. Mech. Eng. 2022, 17(1): 2
https://doi.org/10.1007/s11465-021-0658-y

RESEARCH ARTICLE

Trajectory planning and base attitude restoration of dual-arm free-floating space robot by enhanced bidirectional approach

Zongwu XIE[a], Xiaoyu ZHAO[a], Zainan JIANG (✉)[a], Haitao YANG[b], Chongyang LI[a]

[a] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
[b] Robotic System Department, Jiangsu Jitri-Hust Intelligent Equipment Technology, Wuxi 214000, China
✉ Corresponding author. E-mail: jiangzainan@hit.edu.cn (Zainan JIANG)

ABSTRACT When free-floating space robots perform space tasks, the satellite base attitude is disturbed by the dynamic coupling. The disturbance of the base orientation may affect the communication between the space robot and the control center on earth. In this paper, the enhanced bidirectional approach is proposed to plan the manipulator trajectory and eliminate the final base attitude variation. A novel acceleration level state equation for the nonholonomic problem is proposed, and a new intermediate variable-based Lyapunov function is derived and solved for smooth joint trajectory and restorable base trajectories. In the method, the state equation is first proposed for dual-arm robots with and without end constraints, and the system stability is analyzed to obtain the system input. The input modification further increases the system stability and simplifies the calculation complexity. Simulations are carried out in the end, and the proposed method is validated in minimizing final base attitude change and trajectory smoothness. Moreover, the minute internal force during the coordinated operation and the considerable computing efficiency increases the feasibility of the method during space tasks.

KEYWORDS free-floating space robot, dual arm, coordinated operation, base attitude restoration, bidirectional approach

1 Introduction

Extra vehicular activity plays an important role in space exploration and has been studied by many researchers. It mainly includes large space structure construction, satellite refurbishment and refueling, and space debris removal [1,2]. These tasks are dangerous and laborious for astronauts. As an alternative solution, the space robot attracts attention as a feasible, safe way for on-orbit tasks [3,4].

Among different varieties of space robots, the free-floating space robot (FFSR) stands out for the advantage of fuel saving [5,6]. During space tasks, the FFSR suffers from dynamic coupling that may result in satellite pose disturbance. The disturbance affects robot end positioning accuracy and may interfere with ground communication [7,8]. The end positioning problem can be well solved by the generalized Jacobian matrix-based control. Attentions are focused on minimizing satellite attitude disturbance because ground communication is remarkably influenced by satellite angular motion rather than translation motion [9].

To eliminate satellite base disturbance during trajectory planning, the reactionless null space method [10,11] and the null reaction force method [12,13] were exploited. The former plans trajectories with zero base angular velocity, whereas the latter minimizes the force exerted on the satellite base. Both methods are effective but can only be implemented for manipulators with sufficient redundant degrees of freedom (DOFs). The disturbance map method was proposed to plan low-disturbance paths [14], and the zero-reaction manipulator was designed to decouple the motion between the satellite and the manipulator [8]. However, these two methods cannot be expanded to high-DOF manipulators. The attitude control method was developed to control the base pose by adjusting the manipulators, but the method does not consider end tasks [15]. Free-falling cat was studied, and

Received May 21, 2021; accepted August 18, 2021

the near-optimal control approach was applied to find a minimum disturbance trajectory in joint space, but the method suffers from the velocity discontinuity and low computing efficiency [16]. Nakamura and Mukherjee [17] proposed the bidirectional approach (BA) to solve the nonholonomic problem. The method employs a virtual space robot as an assistant in trajectory planning and can reduce base attitude error while ensuring accurate joint positions. However, jumping joint velocities decrease the reliability and the feasibility.

In the aforementioned studies, researchers mainly focus on single-arm space robots that are fit for simple tasks. For complex tasks in space, multiarm space robots show better extensibility [18]. Multiarm space robots can accomplish tasks in more dexterous ways, such as coordinated operation, parallel task, and target clamping [1]. Studies on multiarm FFSR are also being carried out these years, including capturing tool designs [19], capturing strategies during coordinate operations [20], and satellite–manipulator decoupling strategies of free-flying and free-floating robots [1,21]. To eliminate base disturbance with the decoupling strategy, the rapidly exploring random trees-based method was applied [22]. However, this method needs additional DOFs and does not consider end tasks. Abdul Hafez et al. [1] optimized the satellite base movement of dual-arm robot with the quadratic optimization method. However, the velocity jumps conflict with the real manipulator control. The reactionless null space method was also applied to multiarm space robot to eliminate satellite base attitude disturbance [2,23]. However, additional DOFs or manipulators are needed to balance satellite base disturbance. The same problem was also met by Xie et al. [24], who tried to compensate the base disturbance with balance manipulator. These works mainly pay attention to dual-arm parallel tasks where the robot ends the movement in free space. Scenarios of coordinated operation with dual-arm FFSR are more complicated. Extra constraints are imposed to robot ends in coordinated operation, and the available DOFs are further reduced. As a result, more challenges are encountered for satellite base disturbance minimization.

The studies above show two ideas of solving the base–manipulator coupling problem: The first one tries to keep the base attitude stable in the whole dynamic process, which is mainly used in online trajectory planning, and the second one just focuses on the unity of the initial and final base attitude, which is mainly used in offline trajectory planning. Based on the second idea, this paper proposes a method named enhanced bidirectional approach (EBA) to solve the nonholonomic problem of dual-arm FFSR with limited redundant DOFs. The method aims to keep the satellite attitude the same at the initial and final moment when the dual-arm FFSR works in free space, and minimize the final attitude disturbance when the robot works with additional constraints. As an extension of BA, the method proposed in this paper aims to overcome velocity jumps in BA and ensure a velocity-level continuous trajectory. In the EBA, the trajectory is planned in two situations: free end dual-arm robot and dual-arm robot with end constraints. The model of robot motion in free space is analyzed first, and a general state equation is then determined. A stable control law is introduced based on the analysis of the state equation, and the minimum disturbance trajectory is finally obtained. For tasks with end constraints, the constraint equations are integrated into the robot motion model, and the stable control law can be obtained directly in analogy with the situation of free end control. The EBA aims to decrease base disturbance effectively and ensures smooth joint trajectories. The smooth trajectory can avoid additional base attitude error and internal force in robot control.

The contents of this paper are organized as follows. The tasks and the FFSR analyzed are described in Section 2. The derivations of the EBA with and without end constraints are shown in Section 3. Simulations to validate the reliability and feasibility of the theory are carried out in Section 4. Finally, the paper is concluded in Section 5.

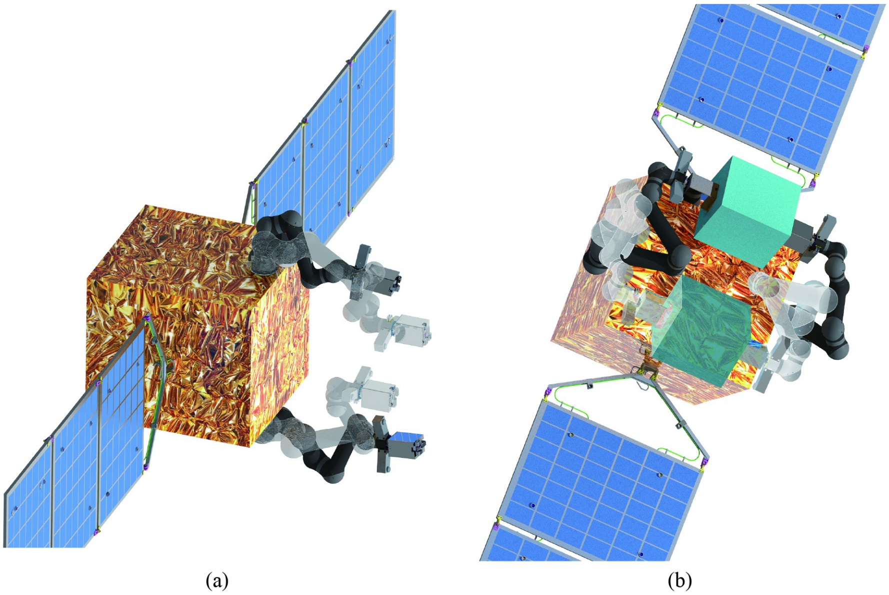
2 Problem description and robot model

2.1 Problem description

The dual-arm space robot is mainly composed of two space manipulators and a satellite base. When the space robot carries out on-orbit tasks, it usually runs into two scenarios, as shown in Fig. 1. In both figures, the opaque robot stays at the initial robot state, and the transparent pose is the final state. In Fig. 1(a), the two manipulators accomplish parallel tasks. Two manipulators are given separate, independent positioning tasks without contact, and the robot motion is planned in free space, similar to single-arm planning. In Fig. 1(b), the two manipulators accomplish tasks in a coordinated manner, such as capturing or recycling an object. In this situation, extra geometry and force constraints are imposed to manipulator ends; thus, further analysis is needed in motion planning.

The robot redundant DOFs are usually limited for satellite base disturbance elimination; thus, a real-time reactionless strategy is challenging. A reasonable solution is to plan the robot motion offline. In this manner, the concern is more about the initial and final states. Then, the planning goal becomes minimizing the final satellite attitude variation and ensuring precise positioning of the robot joints or ends.

In this paper, the task in free space is assumed to position the robot in the desired working pose, and the coordinated task is to capture and drag an object to the designated position. In both cases, the initial and final



(a)

(b)

Fig. 1 Task scenarios of free-floating space robot: (a) robot with free ends; (b) robot with end constraints.

robot configurations (joint angles and satellite base attitude) are predefined, and our planning is to design smooth joint trajectories with eliminated final base attitude variation.

2.2 Robot model

The robot model and the corresponding D-H coordinate system are illustrated in Fig. 2. The robot contains a satellite base supporting two same-structured 7-DOF manipulators. The base coordinate system is located at the mass center of the satellite, and the installation coordinate systems of the manipulators are positioned at the opposite sides of the satellite. The D-H and installation parameters are listed in Table 1, and the inertia parameters are presented in Table 2. The mass centers of the links are measured with respect to the D-H coordinate systems, and the inertia is measured with respect to the corresponding mass centers.

3 Enhanced bidirectional approach

The EBA is proposed to find a smooth trajectory with restorable base attitude. In the method, the initial and final robot configurations are given first, and the trajectory is planned offline. To find a trajectory between the two configurations, a considerable idea is to seek trajectories from the two configurations simultaneously until the two trajectories meet each other. Suppose the trajectory starting from the initial state is T1 and the trajectory sought from the final one is T2. The planned trajectory can be defined as a combination T1 and the inverse of T2. In this manner, trajectory planning can be modelled as a convergence problem of the error between T1 and T2.

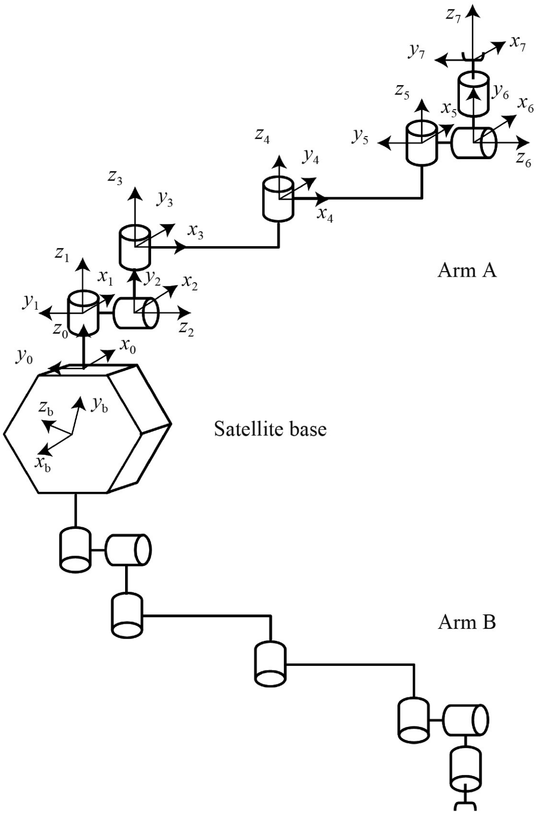


Fig. 2 D-H coordinate systems of dual-arm free-floating space robot.

To propose a base attitude restoration method for dual-arm FFSR, the dual-arm system is analyzed in two scenarios: dual-arm robot moving in free space and dual-arm robot with end constraints. Moreover, the traditional BA method is introduced briefly in the section for a better understanding of the EBA method.

3.1 Overview of the traditional bidirectional approach

Based on the said basic idea, trajectory planning can be equivalent to a “meeting motion” of the real robot and a virtual robot. The real robot is located at the initial configuration at the initial time as the opaque manipulator shown in Fig. 1, and the virtual robot stays at the final configuration at the initial time as the transparent one in Fig. 1. When the planning progress begins, the two robots start moving toward each other, and the process terminates when the state variables of the two robots coincide halfway. Then, T1 and T2 can be defined as the trajectories of the real and virtual robots, respectively.

To find the trajectory error between T1 and T2, the state equation is derived first. Based on the law of conservation of momentum, the satellite angular velocity of FFSR can be calculated as in Eq. (1):

$$\dot{\boldsymbol{\alpha}} = \begin{bmatrix} \dot{\alpha}_z \\ \dot{\alpha}_y \\ \dot{\alpha}_x \end{bmatrix} = \boldsymbol{J}_{\mathrm{s}\alpha}\dot{\boldsymbol{\theta}} = \begin{bmatrix} \cos\alpha_z \tan\alpha_y & \sin\alpha_z \tan\alpha_y & 1 \\ -\sin\alpha_z & \cos\alpha_z & 0 \\ \cos\alpha_z \sec\alpha_y & \sin\alpha_z \sec\alpha_y & 0 \end{bmatrix} \boldsymbol{J}_{\mathrm{s}\omega}\dot{\boldsymbol{\theta}}, \tag{1}$$

where $\boldsymbol{\alpha}$ is the roll-pitch-yaw (RPY) angle of satellite base, α_x, α_y, and α_z are respectively x, y, and z terms of the satellite base RPY angle, $\dot{\boldsymbol{\alpha}}_i$ and $\boldsymbol{\alpha}_i$ are corresponding terms of $\dot{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}$, respectively, $\boldsymbol{J}_{\mathrm{s}\alpha}$ and $\boldsymbol{J}_{\mathrm{s}\omega}$ are the analytical Base-Jacobian and geometric Base-Jacobian, respectively, and $\boldsymbol{\theta}$ is the joint angle.

The robot configuration is defined as the state variable, that is, $\boldsymbol{x} = [\boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}}]^{\mathrm{T}}$, and the input is defined by $\boldsymbol{u} = \dot{\boldsymbol{\theta}}$. The state equation can be stated as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\alpha}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\mathrm{s}\alpha} \\ \boldsymbol{I}_{N\times N} \end{bmatrix} \boldsymbol{u} = \boldsymbol{W}\boldsymbol{u}, \tag{2}$$

where N is the number of robot joints, $\boldsymbol{I}_{N\times N}$ is the identity matrix with N dimension, and $\boldsymbol{W}$ is the input matrix of the robot system in the BA.

To calculate trajectory error, the state equations of the real and virtual robots are established based on Eq. (3). If the states of the real and virtual robots are noted by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively, the following state equations can be obtained:

$$\begin{cases} \dot{\boldsymbol{x}}_1 = \boldsymbol{W}_1\boldsymbol{u}_1, \\ \dot{\boldsymbol{x}}_2 = \boldsymbol{W}_2\boldsymbol{u}_2, \end{cases} \tag{3}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the input matrices of the real and virtual robots in the BA, respectively, and $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the inputs of the real and virtual robots in the BA, respectively

To find a convergent trajectory error, the state error $\Delta\boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_2$ is selected to build the Lyapunov function, as shown in Eq. (4). The input ensuring a convergent trajectory error is derived as Eq. (5). With the input, the corresponding time-derivative of V is shown in Eq. (6).

$$V = \frac{1}{2}\Delta\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\Delta\boldsymbol{x} = (\boldsymbol{x}_1 - \boldsymbol{x}_2)^{\mathrm{T}}\boldsymbol{Q}(\boldsymbol{x}_1 - \boldsymbol{x}_2), \tag{4}$$

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_1^{\mathrm{T}} & \boldsymbol{u}_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = -\left(\boldsymbol{Q}\bar{\boldsymbol{W}}\right)^{\dagger}\Delta\boldsymbol{x}, \tag{5}$$

$$\dot{V} = -\Delta\boldsymbol{x}^{\mathrm{T}}\Delta\boldsymbol{x}, \tag{6}$$

where V is the Lyapunov function of the system, $\Delta\boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_2$ is the state error between the real and virtual robots, $\boldsymbol{Q}$ is an arbitrary symmetric positive-definite matrix, $\tilde{\boldsymbol{u}}$ is the augmented input composed by $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, $\bar{\boldsymbol{W}} = [\boldsymbol{W}_1, -\boldsymbol{W}_2]$, and $(\cdot)^{\dagger}$ means the pseudoinverse of corresponding matrix. Equations (4) and (6) show that the

Table 1 D-H parameters of free-floating space robot system

Joint	θ_i/(°)	α_i/(°)	a_i/m	d_i/m
1	0	0	0	0.225
2	0	90	0	0.240
3	−90	−90	0	0.240
4	0	0	0.56	0.200
5	90	0	0.5	0.195
6	0	90	0	0.180
7	0	−90	0	0.476

Note: To align O_{b} with $O_{0\mathrm{A}}$, translate O_{b} by a vector of (0.3, 0.51, 0.83) m, and rotate with an roll-pitch-yaw (RPY) angle of (90°, 26°, 0°); to align O_{b} with $O_{0\mathrm{B}}$, translate O_{b} by a vector of (0.3, −0.51, −0.83) m, and rotate with an RPY angle of (−90°, −26°, −180°).

Table 2 Inertia parameters of free-floating space robot system

Robot link	Mass/kg	Mass center coordinate			I_{xx}/(kg·m^2)	I_{xy}/(kg·m^2)	I_{xz}/(kg·m^2)	I_{yy}/(kg·m^2)	I_{yz}/(kg·m^2)	I_{zz}/(kg·m^2)
		X/m	Y/m	Z/m						
Base	510.00	0	0	0	625.00	−4.95	6.58	452.50	−2.39	450.00
Body1	6.34	0	−0.07	−0.05	0.09	0	0	0.05	0.02	0.07
Body2	6.34	0	0.07	−0.05	0.09	0	0	0.05	0.02	0.07
Body3	7.93	0.23	0	−0.03	0.08	0	0.12	0.57	0	0.53
Body4	5.67	0.24	0	0	0.03	0	0.05	0.33	0	0.31
Body5	2.80	0	−0.03	−0.03	0.01	0	0	0.01	0	0.01
Body6	2.70	−0.03	0.02	0	0.01	0	0	0.01	0	0.01
Body7	10.01	0	−0.01	−0.20	0.15	0	0	0.20	0.01	0.10

Lyapunov function $V \geqslant 0$, the corresponding time-derivative $\dot{V} \leqslant 0$, and the equal sign holds if and only if $\Delta \boldsymbol{x} = 0$. Thus, the system is asymptotically stable, and the steady-state value of V is zero.

The BA plans the trajectories of the real and virtual manipulators simultaneously. Supposing the two robot states coincide at meeting time t_{m}, the real system input is combined as follows:

$$\boldsymbol{u}(t) = \begin{cases} \boldsymbol{u}_1(t), & 0 \leqslant t \leqslant t_{\mathrm{m}}, \\ -\boldsymbol{u}_2(2t_{\mathrm{m}} - t), & t_{\mathrm{m}} < t \leqslant 2t_{\mathrm{m}}. \end{cases} \tag{7}$$

The derivation and simulation results of the BA show the following: 1) The desired joint angular velocities $\boldsymbol{u}(0)$ and $\boldsymbol{u}(2t_{\mathrm{m}})$ are nonzero, which results in velocity jumps at the initial and final time. 2) When state error $\Delta\boldsymbol{x}$ approaches zero, $\boldsymbol{W}_1$ asymptotically equals $\boldsymbol{W}_2$. Thus, $\bar{\boldsymbol{W}}$ asymptotically tends to be singular. The simulation results show that though $\tilde{\boldsymbol{u}}$ is the product of $\bar{\boldsymbol{W}}^{\dagger}$ and $\Delta\boldsymbol{x}$, in most cases, $\tilde{\boldsymbol{u}}$ is divergent resulting from the singularity of $\bar{\boldsymbol{W}}$. Then, the joint angular velocity jumps at time t_{m}. 3) Owing to the velocity jumps above, when the BA is implemented in real robot control, the difference between the actual and desired trajectories results in extra base attitude variation.

3.2 Enhanced bidirectional approach for dual-arm robots without end constraints

The BA was proposed for the single-arm space robot. In practice, the BA can also be applied to dual-arm system without end constraints because both systems can plan the manipulator joint trajectories freely, that is, no additional constraint equations are needed if the BA is applied to the dual-arm system without end constraints. The EBA improves the BA by planning a smoother joint trajectory, and the situation of Fig. 1(a) is studied first. In the EBA, the system state equation is reconstructed, a new acceleration-level state equation is proposed first, and the corresponding system input is calculated and optimized for system stability and computing efficiency.

3.2.1 System model and input

The angular velocity discontinuities in the BA at time 0 (initial time), t_{m} (meeting time), and $2t_{\mathrm{m}}$ (final time) suggest that only convergent $\Delta\boldsymbol{x}$ is not sufficient for successful robot control. Zero velocity is also needed at time 0, t_{m}, and $2t_{\mathrm{m}}$. Thus, an acceleration-level state equation is proposed in the EBA.

Joint angular acceleration is selected as input $\boldsymbol{U}$, and then the new acceleration-level state equation for the FFSR can be defined as follows:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{N\times N} & \boldsymbol{0}_{N\times(N+3)} \\ \boldsymbol{W} & \boldsymbol{0}_{(N+3)\times(N+3)} \end{bmatrix} \boldsymbol{X} + \begin{bmatrix} \boldsymbol{I}_{N\times N} \\ \boldsymbol{0}_{(N+3)\times N} \end{bmatrix} \boldsymbol{U} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}\boldsymbol{U}, \tag{8}$$

where $\boldsymbol{X} = [\boldsymbol{z}^{\mathrm{T}}, \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}}$ is the state variable of robot in the EBA, and $\boldsymbol{z} = \dot{\boldsymbol{\theta}}$ is the joint angular velocity, $\boldsymbol{x} = [\boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}}]^{\mathrm{T}}$ represents a combination of the satellite RPY angle and robot joint angle, $\boldsymbol{A}$ and $\boldsymbol{B}$ are the state and input matrices of the system in the EBA, respectively, and $\boldsymbol{W}$ can be obtained by Eq. (2).

Based on Eq. (8), the state equations of the real and virtual robots are listed as follows:

$$\begin{cases} \dot{\boldsymbol{X}}_1 = \boldsymbol{A}_1\boldsymbol{X}_1 + \boldsymbol{B}_1\boldsymbol{U}_1, \\ \dot{\boldsymbol{X}}_2 = \boldsymbol{A}_2\boldsymbol{X}_2 + \boldsymbol{B}_2\boldsymbol{U}_2, \end{cases} \tag{9}$$

where $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ and $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ are the state and input matrices of the real and virtual robots in the free-end system in the EBA, respectively, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and $\boldsymbol{U}_1$, $\boldsymbol{U}_2$ are the state variables and the inputs of the real and virtual robots, respectively.

To ensure the trajectory error between the real and virtual robots converging with zero joint velocities, trajectory error and joint velocity are combined to build the Lyapunov function. The combined variable $\boldsymbol{s} \in \boldsymbol{R}^{2N}$ is introduced in Eq. (10), and the Lyapunov function is defined in Eq. (11).

$$\boldsymbol{s} = \boldsymbol{P}(t)\Delta\boldsymbol{x} + \boldsymbol{z}, \tag{10}$$

$$V = \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s}, \tag{11}$$

where $\boldsymbol{P} \in \boldsymbol{R}^{2N\times(N+3)}$ is an undetermined intermediate matrix that unifies the dimensions of $\Delta\boldsymbol{x}$ and $\boldsymbol{z}$, and will be determined in the next section.

Taking a time derivative of Eq. (11) gives

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{s}} = \boldsymbol{s}^{\mathrm{T}}\left(\dot{\boldsymbol{P}}(t)\Delta\boldsymbol{x} + \boldsymbol{P}(t)\bar{\boldsymbol{W}}\boldsymbol{z} + \boldsymbol{U}\right). \tag{12}$$

Equations (11) and (12) show that if the system input is calculated as in Eq. (13), $\dot{V}$ is not larger than zero, as shown in Eq. (14).

$$\tilde{\boldsymbol{U}} = \begin{bmatrix} \boldsymbol{U}_1^{\mathrm{T}} & \boldsymbol{U}_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = -\dot{\boldsymbol{P}}(t)\Delta\boldsymbol{x} - \boldsymbol{P}(t)\bar{\boldsymbol{W}}\boldsymbol{z} - k\boldsymbol{s}, \tag{13}$$

$$\dot{V} = -k\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s} \leqslant 0, \tag{14}$$

where k is an arbitrary positive number, and $\tilde{\boldsymbol{U}}$ is the augmented input composed by $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$. The equal sign holds if and only if $\boldsymbol{s} = \boldsymbol{0}$. Thus, the system is asymptotically stable, and the steady-state value of V is zero.

3.2.2 Calculation of undetermined matrix $\boldsymbol{P}$

Input $\tilde{\boldsymbol{U}}$ calculated by Eq. (13) ensures that $\boldsymbol{s}$ converges to $\boldsymbol{0}$. Then Eq. (10) is reformulated as

$$\boldsymbol{z} = -\boldsymbol{P}(t)\Delta\boldsymbol{x}. \tag{15}$$

Substituting the joint angular velocity $\boldsymbol{z}$ into Eq. (2) gives

$$\Delta\dot{\boldsymbol{x}} = \bar{\boldsymbol{W}}\boldsymbol{z} = -\bar{\boldsymbol{W}}\boldsymbol{P}\Delta\boldsymbol{x}. \tag{16}$$

$\boldsymbol{P} = m\bar{\boldsymbol{W}}^{\dagger}$ is assumed, where m is an arbitrary positive number. Equation (16) is rewritten as

$$\Delta\dot{\boldsymbol{x}} = -m\Delta\boldsymbol{x}. \tag{17}$$

Define $\Delta\boldsymbol{\alpha}$ as the base RPY angle error and $\Delta\boldsymbol{\theta}$ as the joint angle error, then Eq. (17) shows that $\Delta\boldsymbol{x} = [\Delta\boldsymbol{\alpha}^{\mathrm{T}}, \Delta\boldsymbol{\theta}^{\mathrm{T}}]^{\mathrm{T}}$ decays exponentially until around zero. If $\bar{\boldsymbol{W}}$ is a nonsingular matrix, $\boldsymbol{z} = \dot{\boldsymbol{\theta}}$ will also converge to zero according to Eq. (15). However, similar to the BA, $\bar{\boldsymbol{W}}$ will gradually become singular as $\Delta\boldsymbol{x}$ approaches zero; thus, further input modification is required to keep the system stable.

3.3 System input modification and parameter analysis

3.3.1 System input modification

Substituting $\boldsymbol{P} = m\bar{\boldsymbol{W}}^{\dagger}$ into Eq. (13) gives

$$\boldsymbol{U} = -\left(m\dot{\bar{\boldsymbol{W}}}^{\dagger} + km\bar{\boldsymbol{W}}^{\dagger}\right)\Delta\boldsymbol{x} - \left(m\bar{\boldsymbol{W}}^{\dagger}\bar{\boldsymbol{W}} + k\boldsymbol{I}_{N\times N}\right)\boldsymbol{z}, \tag{18}$$

where $\dot{\bar{\boldsymbol{W}}}^{\dagger}$ is the time derivative of $\bar{\boldsymbol{W}}^{\dagger}$ and can be determined by Eq. (19). $\dot{\bar{\boldsymbol{W}}}^{\dagger}$ is a high-order term of $\bar{\boldsymbol{W}}^{\dagger}$ and is more susceptible to singularity [25].

$$\dot{\bar{\boldsymbol{W}}}^{\dagger} = -\bar{\boldsymbol{W}}^{\dagger}\dot{\bar{\boldsymbol{W}}}\bar{\boldsymbol{W}}^{\dagger} + \left(\boldsymbol{I} - \bar{\boldsymbol{W}}^{\dagger}\bar{\boldsymbol{W}}\right)\dot{\bar{\boldsymbol{W}}}^{\mathrm{T}}\left(\bar{\boldsymbol{W}}^{\dagger}\right)^{\mathrm{T}}\bar{\boldsymbol{W}}^{\dagger}. \tag{19}$$

To eliminate the singularity influence on the system stability, $\dot{\bar{\boldsymbol{W}}}$ is neglected in the paper, and the Moore–Penrose inverse is substituted by the damped least square inverse [26]. Then, Eq. (20) is obtained:

$$\tilde{\boldsymbol{U}} = \left[\boldsymbol{U}_1^{\mathrm{T}} \quad \boldsymbol{U}_2^{\mathrm{T}}\right]^{\mathrm{T}} = -km\bar{\boldsymbol{W}}^{\dagger\dagger}\Delta\boldsymbol{x} - \left(m\bar{\boldsymbol{W}}^{\dagger\dagger}\bar{\boldsymbol{W}} + k\boldsymbol{I}_{N\times N}\right)\boldsymbol{z}, \tag{20}$$

where $\bar{\boldsymbol{W}}^{\dagger\dagger}$ is the damped least square inverse of $\bar{\boldsymbol{W}}$ and defined as

$$\bar{\boldsymbol{W}}^{\dagger\dagger} = \left(\bar{\boldsymbol{W}}^{\mathrm{T}}\bar{\boldsymbol{W}} + \lambda\boldsymbol{I}_{N\times N}\right)^{-1}\bar{\boldsymbol{W}}^{\mathrm{T}}, \tag{21}$$

where λ is the damping factor, if $\lambda = 0$, $\bar{\boldsymbol{W}}^{\dagger\dagger} = \bar{\boldsymbol{W}}^{\dagger}$. Then, the system input is derived as

$$\boldsymbol{U}(t) = \begin{cases} \boldsymbol{U}_1(t), & 0 \leqslant t \leqslant t_{\mathrm{m}}, \\ \boldsymbol{U}_2(2t_{\mathrm{m}} - t), & t_{\mathrm{m}} < t \leqslant 2t_{\mathrm{m}}. \end{cases} \tag{22}$$

3.3.2 Influence of neglecting $\dot{\bar{\boldsymbol{W}}}$

The absence of $\dot{\bar{\boldsymbol{W}}}$ in Eq. (22) may affect the convergence of $\boldsymbol{s}$ or even invalidate the algorithm. To decrease the influence of neglecting $\dot{\bar{\boldsymbol{W}}}$, a proper selection of the coefficient set is required. The analysis of Eq. (18) shows that when $k >> m$, $\dot{\bar{\boldsymbol{W}}}$ would have much less effect than $\bar{\boldsymbol{W}}$ on the system and confines the neglection effect.

3.3.3 Influence of damped least square term $\bar{\boldsymbol{W}}^{\dagger\dagger}$

The damped least square term may influence the system in two ways. A small damping factor λ has a minimal influence on the system in the inception phase; thus, $\Delta\boldsymbol{x}$ still converges to zero. Though $\bar{\boldsymbol{W}}^{\dagger}$ approaches infinity, $\bar{\boldsymbol{W}}^{\dagger\dagger}$ still stays finite; hence, the first term in Eq. (20) converges to zero. The following Lyapunov function is considered:

$$V = \frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}. \tag{23}$$

Taking a time derivative gives

$$\dot{V} = \boldsymbol{z}^{\mathrm{T}}\dot{\boldsymbol{z}} = -\boldsymbol{z}^{\mathrm{T}}\left(m\bar{\boldsymbol{W}}^{\dagger\dagger}\bar{\boldsymbol{W}} + k\boldsymbol{I}_{N\times N}\right)\boldsymbol{z}, \tag{24}$$

where if $k >> m$, $m\bar{\boldsymbol{W}}^{\dagger\dagger}\bar{\boldsymbol{W}} + k\boldsymbol{I}_{N\times N}$ is positive definite, thus, V is asymptotic stable, and $\boldsymbol{z}$ gradually converges to zero.

When damping factor λ is large, a large damping factor has a greater effect on the Base-Jacobian because $\boldsymbol{W}$ is mainly composed of the Base-Jacobian matrix and the identity matrix, where the former has a much lower order than the latter. The greater influence on the Base-Jacobian then leads to a fluctuated $\Delta\boldsymbol{\alpha}$. As a combination of $\Delta\boldsymbol{\alpha}$ and $\Delta\boldsymbol{\theta}$, $\Delta\boldsymbol{x}$ no longer approaches zero, and $\boldsymbol{z}$ fluctuates around 0 at time t_{m}.

To sum up, a small damping factor causes overlarge joint angular acceleration and velocity, whereas a large damping factor results in large tracking errors. In practice, the damping factor should be selected according to the actual scenario.

3.4 Enhanced bidirectional approach for dual-arm robots with end constraints

The former section introduced the method for the single-arm or dual-arm space robot without end constraints. In this section, the trajectory planning problem in the dual-arm coordinated operation is addressed. In this situation, the EBA is integrated with geometric constraints, and the corresponding input is modified.

The dual-arm FFSR system is schematically illustrated in Fig. 3. O_{g} is the inertial coordinate system; $\boldsymbol{r}_{\mathrm{a}}$ and $\boldsymbol{r}_{\mathrm{b}}$ are the end vectors of arms A and B, respectively; $\boldsymbol{r}_{\mathrm{ab}}$ is the vector pointing from arm B end to arm A end; $\boldsymbol{v}_{\mathrm{a}}$, $\boldsymbol{v}_{\mathrm{b}}$, $\boldsymbol{\omega}_{\mathrm{a}}$,

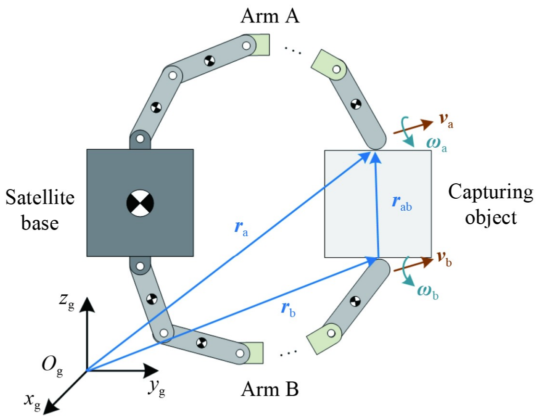


Fig. 3 Structure of dual-arm free-floating space robot system.

and $\boldsymbol{\omega}_{\rm b}$ are the end velocities and the end angular velocities of arms A and B, respectively. All vectors are presented with respect to $O_{\rm g}$.

When the dual-arm robot is performing a coordinated operation, the velocities and angular velocities of the two arm ends satisfy the following constraints:

$$\begin{cases} \boldsymbol{v}_{\rm a} = \boldsymbol{v}_{\rm b} + \boldsymbol{\omega}_{\rm b} \times \boldsymbol{r}_{\rm ab}, \\ \boldsymbol{\omega}_{\rm a} = \boldsymbol{\omega}_{\rm b}. \end{cases} \tag{25}$$

According to robot inverse kinematics, the constraints can be transformed into

$$\begin{bmatrix} \boldsymbol{J}_{{\rm G}v{\rm b}} - \boldsymbol{J}_{{\rm G}v{\rm a}} + (\boldsymbol{r}_{\rm ab}\times)\boldsymbol{J}_{{\rm G}\omega{\rm a}} \\ \boldsymbol{J}_{{\rm G}\omega{\rm b}} - \boldsymbol{J}_{{\rm G}\omega{\rm a}} \end{bmatrix} \dot{\boldsymbol{\theta}} = \boldsymbol{H}\dot{\boldsymbol{\theta}} = \boldsymbol{0}, \tag{26}$$

where $\boldsymbol{H}$ is the coefficient of the geometric constraints in coordinated operation, $\boldsymbol{J}_{{\rm G}v{\rm a}}$, $\boldsymbol{J}_{{\rm G}v{\rm b}}$ and $\boldsymbol{J}_{{\rm G}\omega{\rm a}}$, $\boldsymbol{J}_{{\rm G}\omega{\rm b}}$ are the traditional end velocity and the angular velocity Jacobians of arms A and B, respectively, $\boldsymbol{r}\times$ is the skew-symmetric matrix and is defined as

$$\boldsymbol{r}\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \tag{27}$$

Equation (26) shows that $\dot{\boldsymbol{\theta}}$ is located in the null space of $\boldsymbol{H}$ and

$$\dot{\boldsymbol{\theta}} = \left(\boldsymbol{I}_{N\times N} - \boldsymbol{H}^{\dagger}\boldsymbol{H}\right)\boldsymbol{\xi} = \boldsymbol{L}\boldsymbol{\xi}, \tag{28}$$

where $\boldsymbol{L}$ is the null space of $\boldsymbol{H}$, and $\boldsymbol{\xi}$ is an arbitrary vector. $\boldsymbol{U} = \boldsymbol{\xi}$ is selected, and then Eq. (8) can be transformed into

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{N\times N} & \boldsymbol{0}_{N\times(N+3)} \\ \boldsymbol{W}_L & \boldsymbol{0}_{(N+3)\times(N+3)} \end{bmatrix} \boldsymbol{X} + \begin{bmatrix} \boldsymbol{I}_{N\times N} \\ \boldsymbol{0}_{(N+3)\times N} \end{bmatrix} \boldsymbol{U} = \boldsymbol{A}_L\boldsymbol{X} + \boldsymbol{B}_L\boldsymbol{U}, \tag{29}$$

where $\boldsymbol{W}_L$ is the mapping matrix from the variable $\boldsymbol{z}$ to variable $\dot{\boldsymbol{x}}$ in the constraint-end robot system and is defined by $\boldsymbol{W}_L = \boldsymbol{W}\cdot\boldsymbol{L}$, $\boldsymbol{A}_L$ is the state matrices of constraint-end system in the EBA and $\boldsymbol{B}_L$ is the input matrices of the constraint-end system in the EBA. Corresponding complete state equations are

$$\begin{cases} \dot{\boldsymbol{X}}_1 = \boldsymbol{A}_{L1}\boldsymbol{X}_1 + \boldsymbol{B}_{L1}\boldsymbol{U}_1, \\ \dot{\boldsymbol{X}}_2 = \boldsymbol{A}_{L2}\boldsymbol{X}_2 + \boldsymbol{B}_{L2}\boldsymbol{U}_2, \end{cases} \tag{30}$$

where $\boldsymbol{A}_{L1}$ and $\boldsymbol{A}_{L2}$ are the state matrices of the real and virtual robots in the constraint-end system in the EBA, and $\boldsymbol{B}_{L1}$ and $\boldsymbol{B}_{L2}$ are the input matrices of the real and virtual robots in the constraint-end system in the EBA.

Then, in analogy with Eq. (20), the input can be derived as Eq. (31):

$$\tilde{\boldsymbol{U}} = -km\bar{\boldsymbol{W}}_L^{\dagger\dagger}\Delta\boldsymbol{x} - \left(m\bar{\boldsymbol{W}}_L^{\dagger\dagger}\bar{\boldsymbol{W}}_L + k\boldsymbol{I}_{N\times N}\right)\boldsymbol{z}, \tag{31}$$

where $\bar{\boldsymbol{W}}_L$ is the augmented mapping matrix in the constraint-end system, and can be calculated in analogy with Eq. (5).

As a null space of $\boldsymbol{H}$, $\boldsymbol{L}$ is not rank full; thus, $\boldsymbol{W}_L = \boldsymbol{W}\cdot\boldsymbol{L}$ is a singular matrix. To limit the robot joint velocities, a large damping factor λ is needed in $\bar{\boldsymbol{W}}_L^{\dagger\dagger}$, and this factor selection results in the fluctuated $\Delta\boldsymbol{\alpha}$ and $\boldsymbol{z}$. To tackle this problem, a further modification is introduced to Eq. (31) to ensure the convergence of $\boldsymbol{z}$:

$$\begin{aligned} \boldsymbol{U} = & -m\left[k + {\rm Sig}(t,t_0)\left({\rm Sig}(t,t_0) - 1\right)\right] \\ & \cdot {\rm Sig}(t,t_0)\bar{\boldsymbol{W}}_L^{\dagger\dagger}\Delta\boldsymbol{x} - \left(m\bar{\boldsymbol{W}}_L^{\dagger\dagger}\bar{\boldsymbol{W}}_L + k\boldsymbol{I}_{N\times N}\right)\boldsymbol{z}, \end{aligned} \tag{32}$$

where ${\rm Sig}(t, t_0)$ is the Sigmoid function and can be obtained by Eq. (33), and t_0 is the time when velocities are desired to be zero. In practice, t_0 can be selected as the time when $\Delta\boldsymbol{\theta}$ is located near zero.

$${\rm Sig}(t,t_0) = 1 - \frac{1}{1 + {\rm e}^{-t+t_0}}. \tag{33}$$

4 Simulation and verification

The EBA aims to plan joint trajectories satisfying the joint angle demands and satellite base attitude demands. In this paper, two simulations are carried out: One aims to verify the control of the FFSR in the free space, and the other aims to validate the control of the dual-arm robot system with end constraints.

Each simulation is composed of two processes: planning and controlling. In planning, the initial and final conditions are first determined, and then the joint trajectories are planned. In controlling, the planned trajectories work as the joint controller inputs, and the joint space torque control method (closed-loop inverse dynamic control method for free space moving and master–slave for coordinated operation) is implemented for robot control.

4.1 Simulation of the free-floating space robot without end constraints

The experimental and controlled groups are set in the simulation to validate the EBA methods. The BA, the 5-degree-polynomial planning method, and the near-optimal control approach are employed in the controlled group. The 5-degree-polynomial planning method is an offline method, and it plans the joint trajectory with 5-degree-polynomial from the initial joint angle to the aim joint angle, as shown in Eq. (34). The near-optimal control approach is also an offline method, and it is proposed to optimize the final base attitude variation. In the method, the system state equation is also described by Eq. (2), and the system input is designed by a combination of finite number of Fourier orthogonal basis, as shown in Eq. (35). The coefficients of the orthogonal basis are solved by modified quasi-Newton algorithm during optimization [16]. In the controlling simulation, the traditional closed-loop proportional-differential (PD) inverse dynamic control method [27] is applied for robot control.

$$\begin{aligned}\boldsymbol{\theta}_t = &\boldsymbol{k}_0 + \boldsymbol{k}_1(t-t^*) + \boldsymbol{k}_2(t-t^*)^2 + \boldsymbol{k}_3(t-t^*)^3 \\ &+ \boldsymbol{k}_4(t-t^*)^4 + \boldsymbol{k}_5(t-t^*)^5,\end{aligned} \tag{34}$$

where t^* is the initial time of trajectory planning, and the coefficients are calculated by the boundary condition of the trajectory.

$$\boldsymbol{u} = \begin{bmatrix} \dfrac{k_{00}}{2} + \sum\limits_{i=1}^{h}\left[k_{\mathrm{c}0i}\cos\left(\dfrac{2\pi i}{T}t\right) + k_{\mathrm{s}0i}\sin\left(\dfrac{2\pi i}{T}t\right)\right] \\ \dfrac{k_{10}}{2} + \sum\limits_{i=1}^{h}\left[k_{\mathrm{c}1i}\cos\left(\dfrac{2\pi i}{T}t\right) + k_{\mathrm{s}1i}\sin\left(\dfrac{2\pi i}{T}t\right)\right] \\ \vdots \\ \dfrac{k_{N0}}{2} + \sum\limits_{i=1}^{h}\left[k_{\mathrm{c}Ni}\cos\left(\dfrac{2\pi i}{T}t\right) + k_{\mathrm{s}Ni}\sin\left(\dfrac{2\pi i}{T}t\right)\right] \end{bmatrix}, \tag{35}$$

where h is the number of the Fourier orthogonal basis, $k_{\mathrm{c}ji}$ and $k_{\mathrm{s}ji}$ are the coefficients of the ji terms of the Fourier orthogonal basis, and T is the total planning time of the near-optimal method.

4.1.1 Simulation parameter setting

The joint trajectory is planned according to the joint angles and the base attitudes at the initial and final times. The initial joint angles are defined as [(−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°), (−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°)], the final joint angles are defined as [(−23.44°, −80°, −17.49°, 134.8°, −12.33°, 111.56°, −38°), (−23.44°, −180°, 47.51°, 144.8°, 7.67°, 86.56°, −38°)], and the initial and final RPY angles are defined as [0°, 0°, 0°].

Four methods are applied in the planning simulation, and the parameters are selected as follows. For the EBA, the parameters are set as k = 1.3, and m = 0.125. Given that k is larger than 10m, it is in accordance with $k >> m$ and meets the demand of Eq. (20). Moreover, to eliminate the influence of the Sigmoid function and the damped least square term, t_0 = 1000 s and λ = 0 are set. For the BA, the effect of $\boldsymbol{Q}^{-1}$ on the system is similar to m in the EBA. Thus, $\boldsymbol{Q} = 8\boldsymbol{I}$ is selected. For the near-optimal control approach, the number of Fourier orthogonal basis is 98 (=14×7), and the penalty factor is 10000.

The simulation time of planning is determined as follows. For BA and EBA, the time is determined by the value of $\Delta\boldsymbol{x}$. The simulation time of 5-degree-polynomial planning method and the near-optimal control approach is affected by the joint velocity and joint acceleration. The unlimited-time simulation results show that the velocities of the BA are divergent, whereas the EBA works with a fine convergence and preferable stability. For an effective comparison, the simulation time of the planning simulation is set as 300 s. Within this period, meeting velocities of the EBA converge to 0.001°/s while the condition of the BA is acceptable. The simulation time of the two other controlled groups is set as 20 s, which limits the joint velocity and joint acceleration within 10°/s and 10°/s^2.

In the controlling simulation, the traditional closed-loop PD inverse dynamic control method is applied. In the method, the PD parameters are selected according to the joint acceleration. For the feasibility of the control system, excessively large acceleration should be eliminated to enable motor driving. In the BA method and near-optimal control approach, small PD parameters are selected to limit joint acceleration because the velocities are discontinuous. In the EBA and the 5-degree-polynomial planning method, continuous velocities permit much larger PD parameters. In the simulation, the acceleration is limited by 10°/s^2; thus, PD parameters are selected as proportional parameter K_p = 0.15, differential parameter K_d = 0.6 for the velocity jumping group, and K_p = 10, K_d = 40 for the continuous velocity group. In the controlling simulation, the simulation time is extended to show the step response process at the velocity jumping moment.

4.1.2 Simulation results and analysis

Though the simulation time of the 5-degree-polynomial planning method and the near-optimal control approach is shorter than that of BA and EBA, the actual calculation time of BA and EBA is approximately 370 s, and the actual run time of the 5-degree-polynomial planning method and the near-optimal control approach are approximately 0.42 and 6300 s, respectively. The run time of the 5-degree-polynomial planning method is much less than that of the others because the calculation is achieved only by a polynomial computation. This process is fast, but the method does not optimize the base attitude. For the other methods, the BA and the EBA have higher calculation efficiency than the near-optimal control approach.

The joint angular velocities of simulations are illustrated in Fig. 4. Figures 4(a)–4(d) are the joint angular velocities of the EBA, the BA, the 5-degree-polynomial planning method, and the near-optimal control approach, respectively. To compare the BA and the EBA clearly, their joint angular velocities are dispersed along the axis labelled as “Robot joint order,” as shown in Figs. 4(a) and 4(b). In both figures, joint orders 1–7 represents the joints 1–7 of arm A, and orders 8–14 mean the joints 1–7 of arm B. The controlling simulation time for the BA, the EBA, and the near-optimal control approach is extended. For the BA and the near-optimal control approach, extensions are used to show the complete the step response process, and the extension of the EBA is used as a comparison of BA.

In Fig. 4, the desired joint angular velocity provided by the planning simulation is plotted by dashed lines, and the

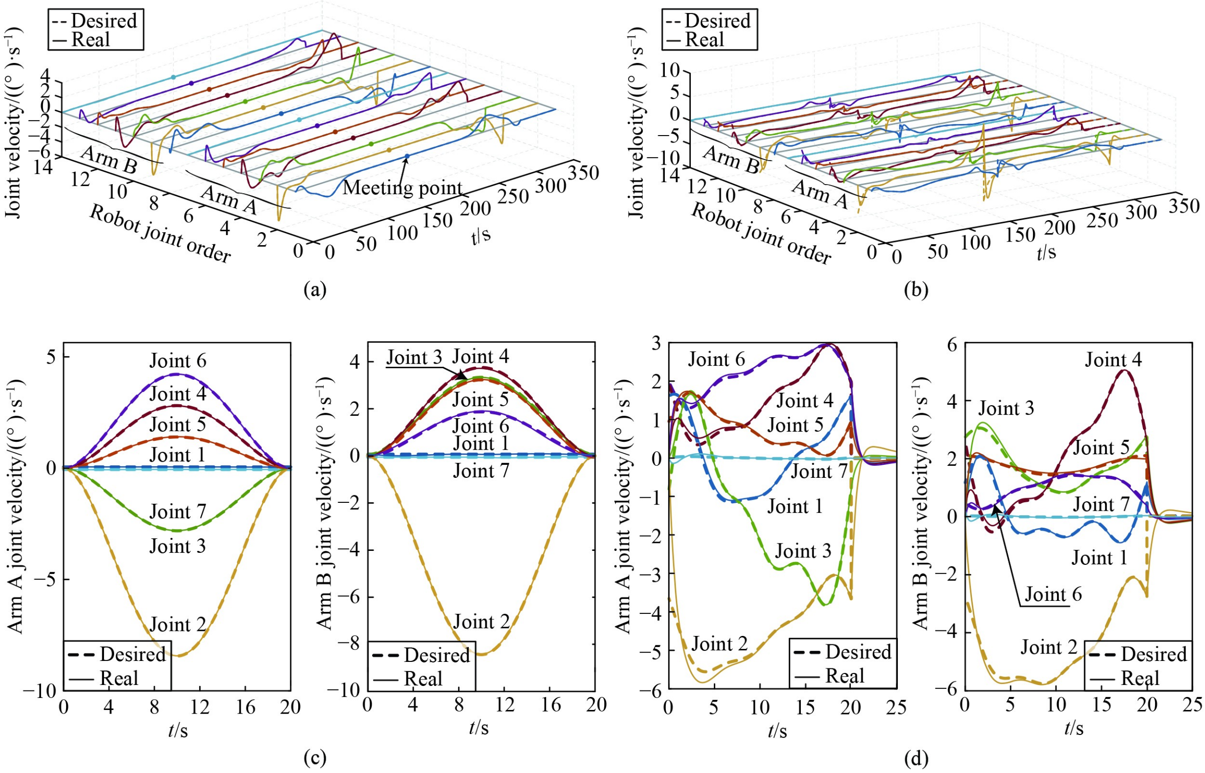
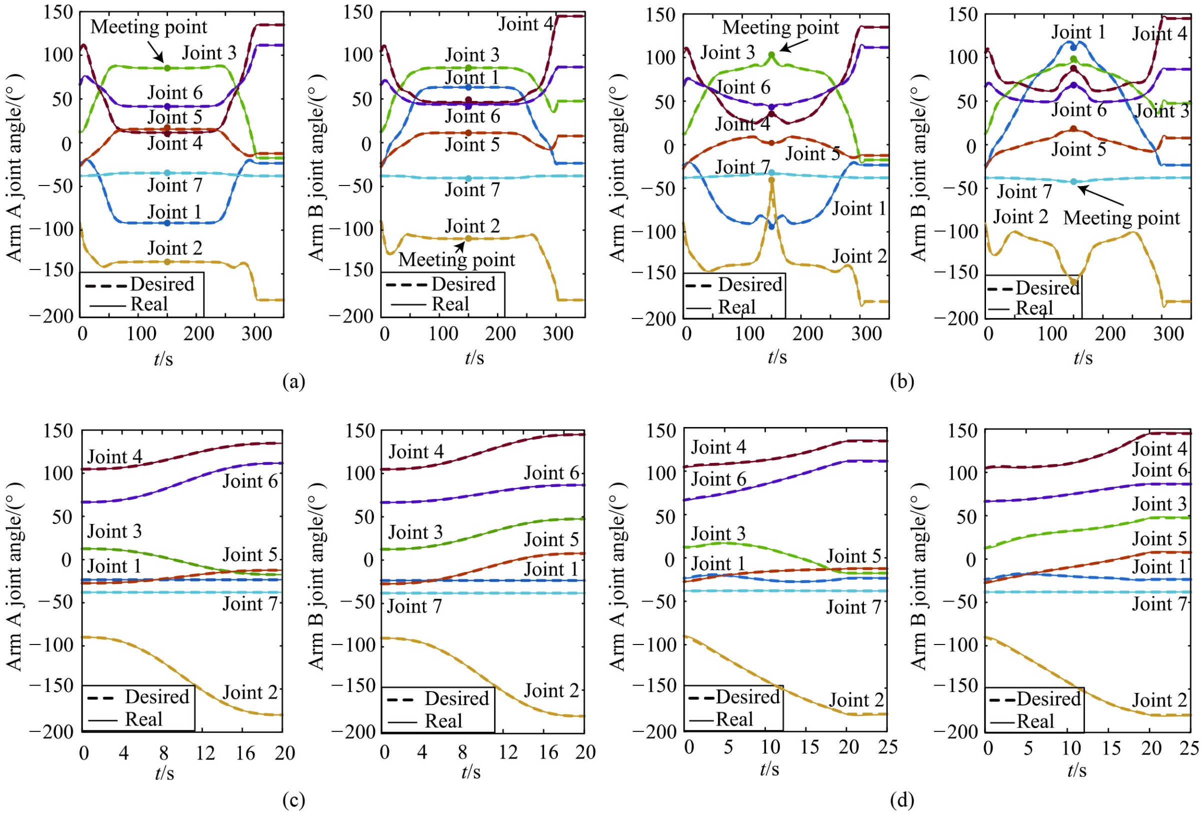


Fig. 4 Joint angular velocity of free-floating space robot without end constraints: (a) joint angular velocity in the enhanced bidirectional approach, (b) joint angular velocity in the bidirectional approach, (c) joint angular velocity in the 5-degree-polynomial planning method, and (d) joint angular velocity in the near-optimal control approach.

real joint angular velocity obtained from the controlling simulation is depicted by solid lines. Figure 4(a) shows that the real joint angular velocity of the EBA fits the desired values well. The velocities at the initial and final times of all joints are zero, and at the meeting time, the joint velocities are also near zero. Moreover, the "meeting points" are marked by filled circle in the figure, the curves before the points are trajectory T1, and the curves after the point are the trajectory T2. In Fig. 4(b), the desired joint angular velocity of BA jumps at t = 0 s, t = 150 s, and t = 300 s. These sudden changes occur at the initial, meeting, and final times, and induce step responses to real curves. For the 5-degree-polynomial planning method shown in Fig. 4(c), the real curves coincide with the desired curves well. As a well-known second-order continuous method, the controlling simulation time is not expanded, and the results prove that the robot will stop moving at the final time without step response. Figure 4(d) shows that the desired joint angular velocity is continuous during the planning except at the initial and final times. The discontinuity of the desired trajectory causes a step response and tracking error in the real trajectory and decreases the reliability of the method. In the figure, the real (solid lines) and desired curves (dashed lines) are plotted to show the complete responses in the real robot control (especially the step response for the discontinuous desired trajectory). The step response introduces tracking errors to the system; thus, the final base attitude deviates from its theoretical value.

The joint angles of simulations are illustrated in Fig. 5 to show the robot working state, and the real and desired trajectories are plotted for a complete response presentation. Figures 5(a)–5(d) are the joint angles of the EBA, the BA, the 5-degree-polynomial planning method, and the near-optimal control approach, respectively. As in Fig. 4(a), the "meeting points" are marked in Figs. 5(a) and 5(b) to distinguish the trajectory of "real robot" and "virtual robot." In Figs. 5(a)–5(c), the initial and end joint angles are all [(−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°), (−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°)] and [(−23.44°, −80°, −17.49°, 134.8°, −12.33°, 111.56°, −38°), (−23.44°, −180°, 47.51°, 144.8°, 7.67°, 86.56°, −38°)]. In Fig. 5(d), the initial angles are the same while the final joint angle is [(−23.43°, −90.01°, −17.49°, 134.81°, −12.33°, 111.57°, −38°), (−23.43°, −90.01°, 47.52°, 144.81°, 7.68°, 86.56°, −38°)]. Though the final state error of the near-optimal control approach is slightly larger than those of the three other methods, it is still located in the neighbor of the required angle and satisfies the initial and final conditions. For BA, the curve peaks of Fig. 5(b) suggest that the method works with less controlling precision than the three other methods. The figures show that all the methods meet the general requirement of ensuring precise robot arm positioning. The advantage and disadvantage of the methods can be concluded from the trajectory continuity, final base attitude change, and computing efficiency.

The trajectories of satellite base attitudes are plotted in Fig. 6. Figures 6(a), 6(b), and 6(d) are the base RPY angles of the EBA, the BA, the 5-degree-polynomial planning method, and the near-optimal control approach, respectively. For the EBA, the final desired and real base RPY angles are [0°, 0°, 0°] and [−0.01°, 0°, 0.01°]. The final desired and real RPY angles of BA are [0°, 0°, 0°] and [−0.07°, 0.02°, 0.13°]. For the 5-degree-polynomial planning method, the final desired and real base RPY angles are [−0.81°, 0.53°, 0.78°] and [−0.82°, 0.52°, 0.79°]. The final desired and real base RPY angles of the near-optimal control approach are [0°, 0°, 0°] and [0.01°, 0.00°, −0.04°]. The figures show that for the BA and the near-optimal control approach, though the desired final base attitudes are zero, the real values are much larger due to the trajectory discontinuity and the trajectory tracking errors shown in Figs. 4 and 5. The final base attitude of the 5-degree-polynomial planning method is



(a) (b) (c) (d)

Fig. 5 Joint angle of free-floating space robot without end constraints: (a) joint angle in the enhanced bidirectional approach, (b) joint angle in the bidirectional approach, (c) joint angle in the 5-degree-polynomial planning method, and (d) joint angle in the near-optimal control approach.

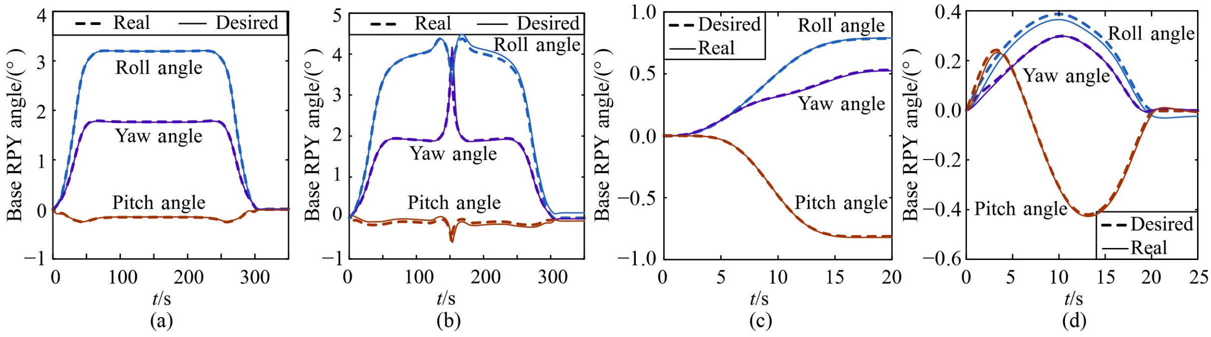


Fig. 6 Satellite base attitude of free-floating space robot without end constraints: (a) RPY angle in the enhanced bidirectional approach, (b) RPY angle in the bidirectional approach, (c) RPY angle in the 5-degree-polynomial planning method, and (d) RPY angle in the near-optimal control approach.

the largest one because no optimization is applied in this method. This phenomenon also suggests the effectiveness of the other optimization methods. For the EBA, the real and desired final base attitudes are the smallest. It proves that the method can decrease the final base attitude variation effectively, and the smooth trajectory has a minimal perturbation on the system.

4.2 Simulation of free-floating space robot with end constraints

The experimental and controlled groups are set in the simulation to validate the EBA method. The last section proved that the trajectories planned by the BA perform worst in robot control. If the method is applied to robot tasks with end constraints, the situation is even worse. Thus, in the section, BA is not selected in the controlled group. To fit the coordinated operation, the leader–follower method is applied instead of the 5-degree-polynomial planning method [28]. In the method, one arm (leader) is planned by the 5-degree polynomial as in Eq. (34), and the other one (follower) is planned by velocity-level inverse kinematics with Jacobian matrix, that is, the controlled group consists of the leader–follower method and the near-optimal control approach.

4.2.1 Simulation parameter setting

In the dual-arm coordinated operation, the relative poses of the two arm ends are kept constant during the whole process. At the initial time, the joint angles are set as [(−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°), (−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°)]. To apply the leader–follower method, the end pose is also given: The end positions are [(0.543, 0.793, 1.888), (1.529, −0.268, −1.332)] m, and the end RPY angles are [(−97.38°, −62.71°, 73.34°), (87.72°, −38.51°, 65.24°)]. At the final time, the joint angles are calculated as [(−23.44°, −141.29°, 16.92°, 168.53°, −110.15°, 75.40°, 16.68°), (−23.44°, −84.38°, 66.04°, 95.38°, −71.42°, 63.47°, −41.08°)], and the corresponding end pose is [(0.104, 0.648, 1.992), (−0.119, −0.147, −2.264)] m, and [(−61.17°, −76.93°, −0.46°), (−14.49°, 29.30°, 167.12°)]. The satellite base RPY angles at the initial and final times are both [0°, 0°, 0°].

In the planning simulation, the simulation time of the EBA is 200 s, and the parameters are selected as $k = 1.1$, $m = 0.1$, $t_0 = 95$ s, and $\lambda = 10^{-5}$. When t_0 is 95 s, the meeting angles converge to 0.01°, which is in accordance with the requirement in Eq. (32). For the near-optimal control approach and the leader–follower method, the simulation time is 20 s, the number of Fourier orthogonal basis is 98 (=14×7), and the penalty factor is 10000. As mentioned above, the simulation time and parameters are selected based on the joint velocity and the acceleration limits.

In the controlling simulation, additional position and force constraints are imposed. The position constraints are satisfied by the planning method, and the force constraints are satisfied by the master–slave method [29]. Given that the EBA plans the robot motion in the joint space, joint space impedance control is applied to the slave arm, and the impedance parameters are selected as $M = 1000$, $B = 400000$, and $K = 10$. Moreover, the PD parameters in the master–slave method affect the end positioning precision. To obtain a precise relative end pose, the PD parameters for the EBA and the leader–follower method are $K_\mathrm{p} = 1000$ and $K_\mathrm{d} = 800$. For the near-optimal control approach, as a result of velocity discontinuity, the parameters are selected as $K_\mathrm{p} = 0.15$ and $K_\mathrm{d} = 0.6$.

4.2.2 Simulation results and analysis

The actual run time of the leader–follower method, the EBA, and the near-optimal control approach are approximately 22, 270, and 6200 s, respectively. The planning efficiency of the EBA is lower than that of the leader–follower method but higher than that of the near-optimal control approach.

Figures 7(a)–7(c) are the joint angular velocities of the EBA, the leader–follower method, and the near-optimal control approach, respectively. The real and desired curves are plotted to indicate the complete system responses. In Fig. 7(a), the real joint angular velocity of the EBA coincides with the desired one, and both start from zero and end at zero. Moreover, the transition velocity of the EBA is also zero, and the transition period is marked by the filled circles in the figure. For the leader–follower method, the initial and final velocities are also zeros. The joint velocities of the two arms show different performances because the leader arm and the follower arm are planned in different ways. In the near-optimal control approach, the desired initial and final velocities are not zero, and step responses exist in the curves. As a result, the real velocity does not fit the desired one well, and the controlling simulation time is extended to 25 s to show the step response. In the coordinated operation, the step responses also introduce trajectory tracking error, and it not only results in additional variation to the base attitude control but also has an influence on the system internal force.

The joint angles of the simulations are shown in Fig. 8. The angles are plotted here to show the working state of the different methods, and “meeting points” are plotted in the figures. The initial joint angles of all figures are [(−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°), (−23.44°, −90°, 12.51°, 104.8°, −27.33°, 66.56°, −38°)]. In Fig. 8(a), the final joint angle is [(−23.44°, −141.29°, 16.92°, 168.53°, −110.15°, 75.40°, 16.68°), (−23.44°,

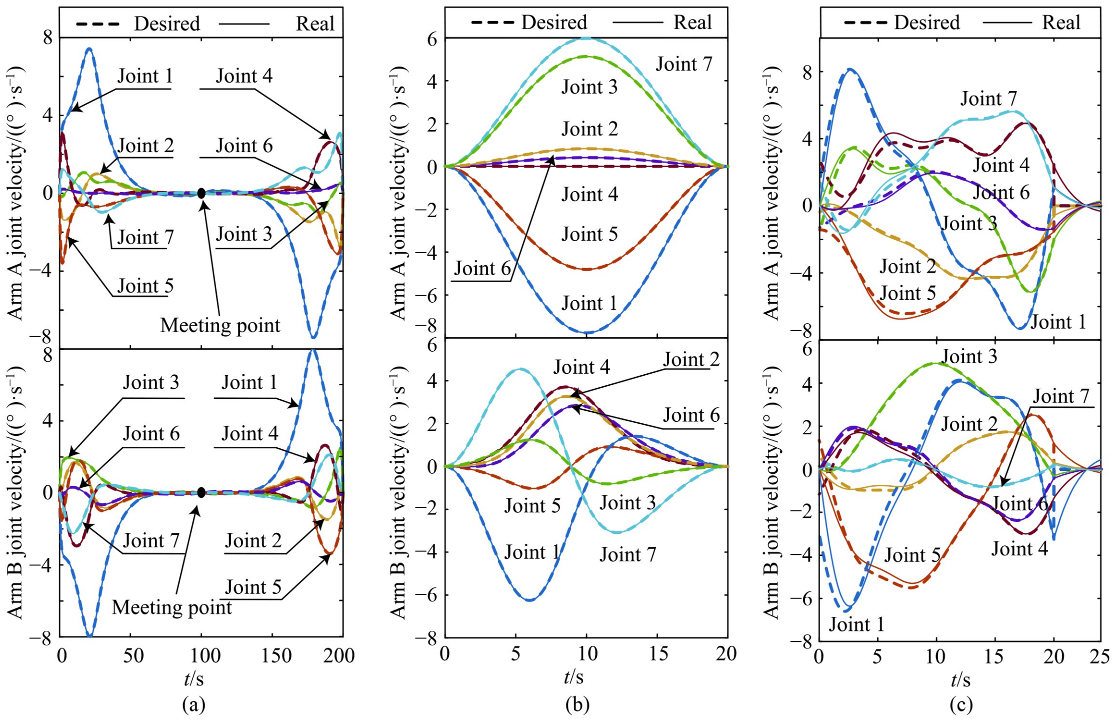


Fig. 7 Joint angular velocity of free-floating space robot with end constraints: (a) joint angular velocity of the enhanced bidirectional approach, (b) joint angular velocity of the leader–follower method, and (c) joint angular velocity of the near-optimal control approach.

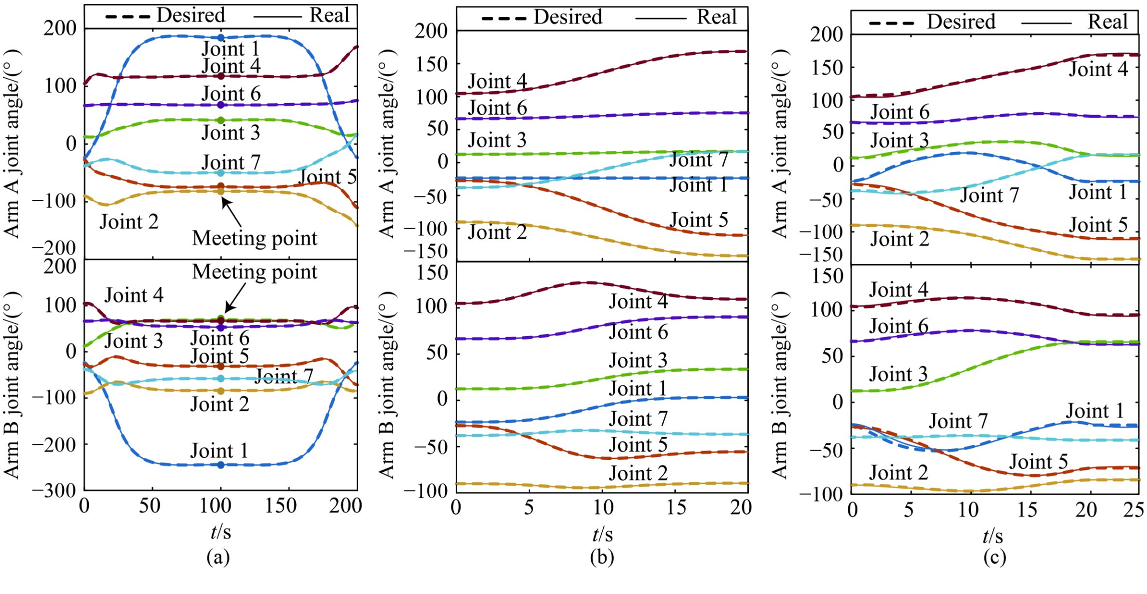


Fig. 8 Joint angle of free-floating space robot with end constraints: (a) joint angle of the enhanced bidirectional approach, (b) joint angle of the leader–follower method, and (c) joint angle of the near-optimal control approach.

−84.38°, 66.04°, 95.38°, −71.42°, 63.47°, −41.08°)]. In Fig. 8(b), the final joint angle is [(−23.44°, −141.29°, 16.92°, 168.53°, −110.15°, 75.40°, 16.68°), (3.15°, −89.45°, 33.67°, 109.37°, −55.55°, 90.16°, −36.61°)]. In Fig. 8(c), the final joint angle is [(−23.46, −141.29, 16.97, 168.50, −110.14, 75.43, 16.66), (−23.3547, −84.39, 66.05, 95.40°, −71.46°, 63.49°, −41.08°)]. For the EBA and the near-optimal control approach, the initial and final angles are all located in the neighbor of the required values (though the error of the near-optimal control approach is slightly larger than that of the EBA). For the leader–follower method, because the joint motion of arm A is planned by the 5-degree polynomial and joint motion of arm B are planned by the Jacobian-based method, the final joint angle of arm A satisfies the initial and final conditions, whereas the joint angle of arm B does not.

However, the end pose of the two arms are the same as those of the other methods. The angles in Fig. 8 show that the robot ends are located at the neighbor of the desired pose relative to the base at the final time.

The satellite base RPY angles are plotted in Figs. 9(a)–9(c). The desired and real RPY angles of the leader–follower method at the end time are [0.32°, 0.67°, 1.75°] and [0.33°, 0.66°, 1.76°]. The desired and real base RPY angles of the near-optimal control approach at the end time are [0°, 0°, 0°] and [−0.02°, −0.05°, −0.18°]. For the EBA, the desired and real base RPY angles at the end time are [0.11°, 0.01°, −0.05°] and [0.08°, 0.00°, −0.04°]. Further simulation results show that the base attitude of the EBA can be reduced more when a smaller damping factor λ is selected. However, the motion range would further expand and eventually exceed the defined joint angle scope with a smaller λ. The base attitude of the leader–follower method becomes largest because the method does not have the ability of optimization. Though the desired final base attitude of the near-optimal control approach is zero, the trajectory tracking error greatly increases the real value. As a result, the real final attitude of EBA is still the smallest one.

For the coordinated operation, a large internal force may damage the robot system; thus, the force control results are focused here as an important item under supervision to show the influence of tracking error on the system thoroughly. In this paper, the master–slave method is applied to decrease the robot internal force. In the method, the master arm is controlled by the rigid body

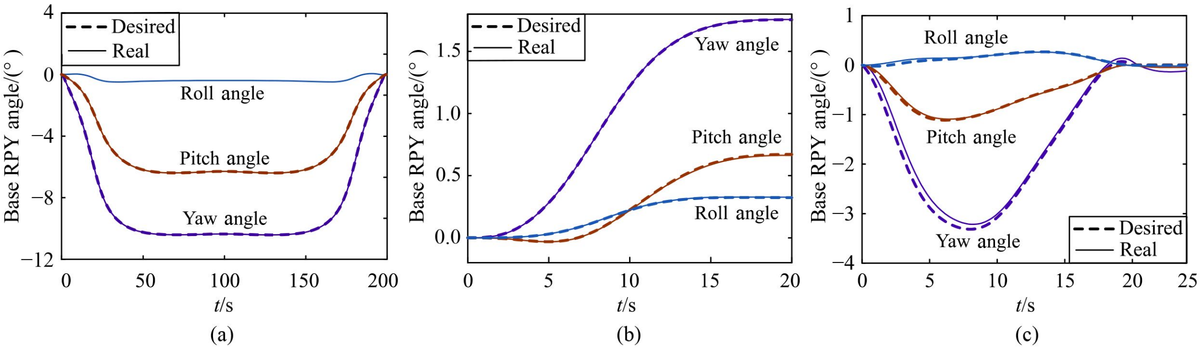


Fig. 9 Satellite base attitude of free-floating space robot with end constraints: (a) RPY angle in the enhanced bidirectional approach, (b) RPY angle in the leader–follower method, and (c) RPY angle in the near-optimal control approach.

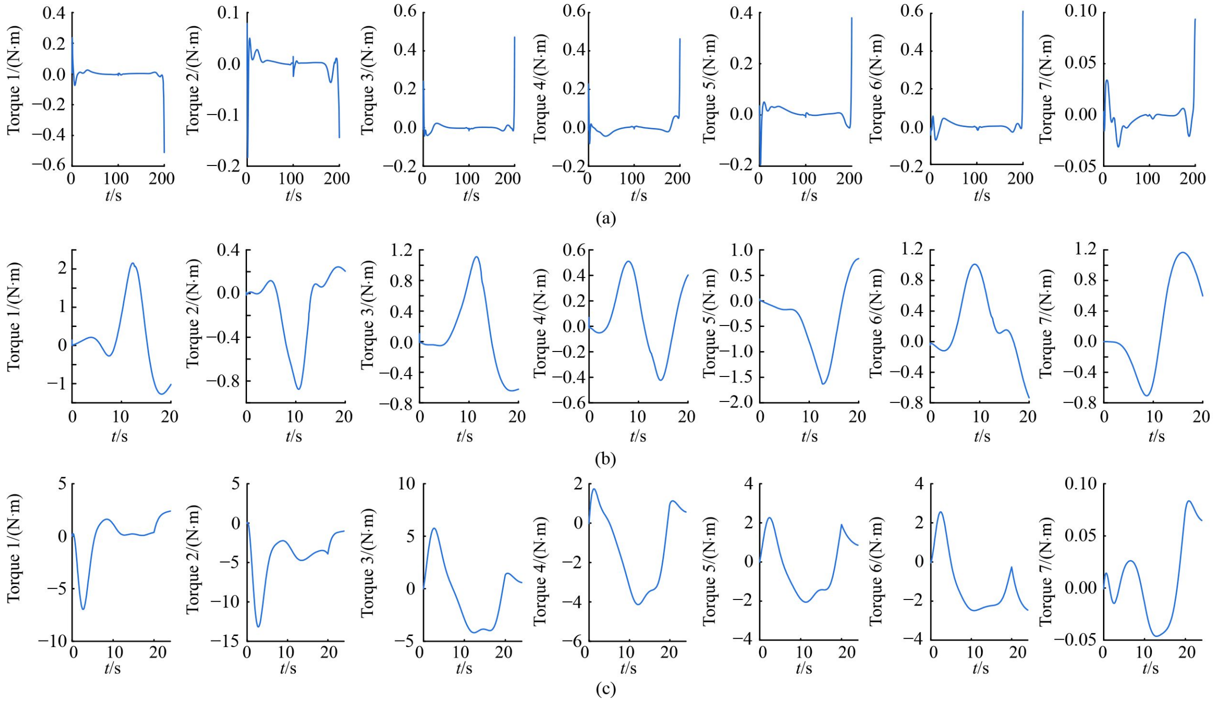


Fig. 10 Impedance force of slave arm in dual-arm system without end constraints: (a) impedance force in the enhanced bidirectional approach, (b) impedance force in the leader–follower method, and (c) impedance force in the near-optimal control approach.

inverse dynamics, and the slave arm is controlled by the joint space impedance control method. The internal impedance torque of the slave arm is shown in Fig. 10. For the EBA, the internal torques are all below 0.6 N·m, and the values can be even limited within 0.1 N·m if the start and end moments are excluded. In Fig. 10(b), the internal joint impedance torques are slightly larger, and the upper limits of the torques are approximately 2 N·m. Simulations show that increased planning time decreases internal torque. For example, if planning time is 35 s, the upper limit of the internal torque will be 0.25 N·m. Compared with the two other methods, the internal joint impedance torques of the near-optimal control approach are much larger, and the upper limit is more than 10 N·m. The internal force of the near-optimal control approach is the largest one, and this phenomenon results from step responses in the tracking trajectories during the controlling.

In summary, the EBA is compared with the four methods in the whole simulations: the BA, the 5-degree-polynomial planning method, the leader–follower method, and the near-optimal control approach. The EBA shows the best performance in final base attitude optimization and joint trajectory tracking, and has the minimum internal force in the coordinated operation. The 5-degree-polynomial planning method and the leader–follower method work worst because they plan the joint trajectory without optimization. The trajectory discontinuity of the BA and the near-optimal control approach introduce additional attitude variation and internal force, thus proving the necessity of the improvement of the BA on velocity-level continuous trajectory. Moreover, compared with the other methods, the EBA has a considerable computing efficiency and is suitable for base attitude restoration in space tasks.

5 Conclusions

To enable the FFSR to start and end with a desired configuration while restoring the base attitude, the EBA is investigated in this paper. By reformulating the joint trajectory planning into an acceleration-level "meeting problem", a robust system model is proposed. The EBA is an off-line planning method that aims to achieve robot free-space tasks and dual-arm robot coordinated operations. Simulations are carried out with and without end constraints. The results verify that compared with the other methods, the EBA decreases the final base variation effectively and ensures smooth joint trajectories when generating robot joint trajectories. The smooth trajectories avoid additional base attitude error and internal force in the robot control; thus, it ensures positioning accuracy and control stability, and benefits the robot in system control, end positioning, and communication. Moreover, the high computing efficiency makes the method practical in real space tasks.

Nomenclature

Abbreviations

BA	Bidirectional approach
DOF	Degree of freedom
EBA	Enhanced bidirectional approach
FFSR	Free-floating space robot
RPY	Roll-pitch-yaw

Variables

$\boldsymbol{A}$, $\boldsymbol{A}_L$	State matrices of the free-end system and constraint-end system in the EBA
$\boldsymbol{A}_1$, $\boldsymbol{A}_2$	State matrices of the real and virtual robots in the free-end system in the EBA
$\boldsymbol{A}_{L1}$, $\boldsymbol{A}_{L2}$	State matrices of the real and virtual robots in the constraint-end system in the EBA
$\boldsymbol{B}$, $\boldsymbol{B}_L$	Input matrices of the free-end system and constraint-end system in the EBA
$\boldsymbol{B}_1$, $\boldsymbol{B}_2$	Input matrices of the real and virtual robots in the free-end system in the EBA
$\boldsymbol{B}_{L1}$, $\boldsymbol{B}_{L2}$	Input matrices of the real and virtual robots in the constraint-end system in the EBA
h	Number of the Fourier orthogonal basis
$\boldsymbol{H}$	Coefficient of the geometric constraints in coordinated operation
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}_{\mathrm{Gva}}$, $\boldsymbol{J}_{\mathrm{Gvb}}$	Velocity general-Jacobian matrix of arms A and B
$\boldsymbol{J}_{\mathrm{G\omega a}}$, $\boldsymbol{J}_{\mathrm{G\omega b}}$	Angular velocity general-Jacobian matrix of arm i
$\boldsymbol{J}_{\mathrm{s}\alpha}$, $\boldsymbol{J}_{\mathrm{s}\omega}$	Analytical and geometric Base-Jacobian
k	Arbitrary positive number
k_{ij}, $k_{\mathrm{c}ij}$, $k_{\mathrm{s}ij}$	Coefficients of the near-optimal control approach
$\boldsymbol{k}_i$	Coefficients of the 5-degree-polynomial
K_{p}	Proportional parameter in the closed-loop PD inverse dynamic control method
K_{d}	Differential parameter in the closed-loop PD inverse dynamic control method
$\boldsymbol{L}$	Null space of $\boldsymbol{H}$
m	Arbitrary positive number
N	Joint number of space robot
O_{g}	Inertial coordinate system
$\boldsymbol{P}$	Undetermined intermediate matrix that unifies the dimensions of $\Delta\boldsymbol{x}$ and $\boldsymbol{z}$
$\boldsymbol{Q}$	Arbitrary symmetric positive-definite matrix
$\boldsymbol{r}_{\mathrm{a}}$, $\boldsymbol{r}_{\mathrm{b}}$	End vectors of arms A and B, respectively

$\boldsymbol{r}_{\mathrm{ab}}$	Vector pointing from arm B end to arm A end
$\boldsymbol{s}$	Combined variable used for Lyapunov function
t_0	Time when the joint velocities are desired to be zero
t_{m}	Meeting time
t^*	Initial time of trajectory planning
T	Total planning time of the near-optimal method
$\boldsymbol{u}$	System input in the BA
$\boldsymbol{u}_1$, $\boldsymbol{u}_2$	Inputs of the real and virtual robots in the BA, respectively
$\tilde{\boldsymbol{u}}$	Augmented input composed by $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, and $\tilde{\boldsymbol{u}}^{\mathrm{T}} = [\boldsymbol{u}_1^{\mathrm{T}}, \boldsymbol{u}_2^{\mathrm{T}}]^{\mathrm{T}}$
$\boldsymbol{U}$	System input in the EBA
$\boldsymbol{U}_1$, $\boldsymbol{U}_2$	Inputs of the real and virtual robots in the EBA, respectively
$\tilde{\boldsymbol{U}}$	Augmented input composed by $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$, and $\tilde{\boldsymbol{U}}^{\mathrm{T}} = [\boldsymbol{U}_1^{\mathrm{T}}, \boldsymbol{U}_2^{\mathrm{T}}]^{\mathrm{T}}$
$\boldsymbol{v}_{\mathrm{a}}$, $\boldsymbol{v}_{\mathrm{b}}$	End velocity of arms A and B, respectively
V	Lyapunov function of the system
$\boldsymbol{W}$	Input matrix of the robot system in the BA
$\boldsymbol{W}_1$, $\boldsymbol{W}_2$	Input matrices of the real and virtual robot systems in the BA, respectively
$\bar{\boldsymbol{W}}$	Augmented input matrix of the robot system in BA, and $\bar{\boldsymbol{W}} = [\boldsymbol{W}_1, -\boldsymbol{W}_2]$
$\boldsymbol{W}_L$	Mapping matrix from variable $\boldsymbol{z}$ to variable $\dot{\boldsymbol{x}}$ of the constraint-end robot system
$\bar{\boldsymbol{W}}_L$	Augmented mapping matrix in the constraint-end system
$\boldsymbol{x}$	State variable of robot in the BA
$\boldsymbol{x}_1$, $\boldsymbol{x}_2$	System state variables of the real and virtual robots in the BA, respectively
$\Delta\boldsymbol{x}$	System state error defined by $\Delta\boldsymbol{x} = \boldsymbol{x}_1 - \boldsymbol{x}_2$
$\boldsymbol{X}$	State variable of robot in the EBA
$\boldsymbol{X}_1$, $\boldsymbol{X}_2$	System state variable of the real and virtual robots in the EBA, respectively
$\boldsymbol{z}$	Joint angular velocity in the EBA and is a component of the system state variable in the EBA
$\boldsymbol{\alpha}$	Vector of satellite base roll-pitch-yaw (RPY) angle, rad
α_x, α_y, α_z	x, y, z terms of the satellite base RPY angle, respectively
$\Delta\boldsymbol{\alpha}$	Base RPY angle error, rad
$\boldsymbol{\theta}$	Vector of joint angle, rad
$\Delta\boldsymbol{\theta}$	Joint angle error, rad
$\boldsymbol{\omega}_{\mathrm{a}}$, $\boldsymbol{\omega}_{\mathrm{b}}$	End angular velocities of arms A and B, respectively
λ	Damping factor
$\boldsymbol{\xi}$	Arbitrary vector

Acknowledgements This study was funded by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 91848202), and the National Natural Science Foundation of China (Grant No. 51875114). No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication.

References

1. Abdul Hafez A H, Mithun P, Anurag V V, Shah S V, Madhava Krishna K. Reactionless visual servoing of a multi-arm space robot combined with other manipulation tasks. Robotics and Autonomous Systems, 2017, 91: 1–10
2. Xu W F, Meng D S, Liu H D, Wang X Q, Liang B. Singularity-free trajectory planning of free-floating multiarm space robots for keeping the base inertially stabilized. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 2019, 49(12): 2464–2477
3. Naveen B, Shah S V, Misra A K. Momentum model-based minimal parameter identification of a space robot. Journal of Guidance, Control, and Dynamics, 2019, 42(3): 508–523
4. MA B Y, Xie Z W, Jiang Z N, Liu H. Precise semi-analytical inverse kinematic solution for 7-DOF offset manipulator with arm angle optimization. Frontiers of Mechanical Engineering, 2021, 16(3): 435–450
5. Wu Y H, Yu Z C, Li C Y, He M J, Hua B, Chen Z M. Reinforcement learning in dual-arm trajectory planning for a free-floating space robot. Aerospace Science and Technology, 2020, 98: 105657
6. Zhang X, Liu J G. Effective motion planning strategy for space robot capturing targets under consideration of the berth position. Acta Astronautica, 2018, 148: 403–416
7. Xu W F, Peng J Q, Liang B, Mu Z G. Hybrid modeling and analysis method for dynamic coupling of space robots. IEEE Transactions on Aerospace and Electronic Systems, 2016, 52(1): 85–98
8. Papadopoulos E, Abu-Abed A. On the design of zero reaction manipulators. Journal of Mechanical Design, 1996, 118(3): 372–376
9. Xu S F, Wang H L, Zhang D Z, Yang B H. Adaptive reactionless motion control for free-floating space manipulators with uncertain kinematics and dynamics. IFAC Proceedings Volumes, 2013, 46(20): 646–653
10. Nguyen-Huynh T C, Sharf I. Adaptive reactionless motion and parameter identification in postcapture of space debris. Journal of Guidance, Control, and Dynamics, 2013, 36(2): 404–414
11. Yoshida K, Hashizume K, Abiko S. Zero reaction maneuver: flight validation with ETS-VII space robot and extension to kinematically redundant arm. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), 2001, 1: 441–446
12. Cocuzza S, Pretto I, Debei S. Reaction torque control of redundant

space robotic systems for orbital maintenance and simulated microgravity tests. Acta Astronautica, 2010, 67(3–4): 285–295

13. Cocuzza S, Pretto I, Debei S. Least-squares-based reaction control of space manipulators. Journal of Guidance, Control, and Dynamics, 2012, 35(3): 976–986
14. Torres M A, Dubowsky S. Minimizing spacecraft attitude disturbances in space manipulator systems. Journal of Guidance, Control, and Dynamics, 1992, 15(4): 1010–1017
15. Yamada K. Attitude control of space robot by arm motion. Journal of Guidance, Control, and Dynamics, 1994, 17(5): 1050–1054
16. Liang X C, Xu L S, Li L. Research on trajectory planning of a robot inspired by free-falling cat based on modified quasi-Newton algorithm. In: Proceedings of 2016 IEEE International Conference on Mechatronics and Automation. Harbin: IEEE, 2016, 552–557
17. Nakamura Y, Mukherjee R. Nonholonomic path planning of space robots via bi-directional approach. In: Proceedings of IEEE International Conference on Robotics and Automation. Cincinnati: IEEE, 1990, 1764–1769
18. Yan L, Xu W F, Hu Z H, Liang B. Multi-objective configuration optimization for coordinated capture of dual-arm space robot. Acta Astronautica, 2020, 167: 189–200
19. Zhang X, Liu J G, Feng J K, Liu Y W, Ju Z J. Effective capture of nongraspable objects for space robots using geometric cage pairs. IEEE/ASME Transactions on Mechatronics, 2020, 25(1): 95–107
20. Shi L L, Jayakody H, Katupitiya J, Jin X. Coordinated control of a dual-arm space robot: novel models and simulations for robotic control methods. IEEE Robotics & Automation Magazine, 2018, 25(4): 86–95
21. Zhang X, Liu J G, Gao Q, Ju Z J. Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. Nonlinear Dynamics, 2020, 100(3): 2449–2467
22. James F, Shah S V, Singh A K, Krishna K M, Misra A K. Reactionless maneuvering of a space robot in precapture phase. Journal of Guidance, Control, and Dynamics, 2016, 39(10): 2419–2425
23. Gouo A, Nenchev D N, Yoshida K, Uchiyama M. Motion control of dual-arm long-reach manipulators. Advanced Robotics, 1998, 13(6): 617–631
24. Xie Y E, Wu X D, Inamori T, Shi Z, Sun X Z, Cui H T. Compensation of base disturbance using optimal trajectory planning of dual-manipulators in a space robot. Advances in Space Research, 2019, 63(3): 1147–1160
25. Magnus J R, Neudecker H. Matrix Differential Calculus with Applications in Statistics and Econometrics. 3rd ed. Oxford: John Wiley & Sons, 2019, 169–171
26. Wampler C W. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. IEEE Transactions on Systems, Man, and Cybernetics, 1986, 16(1): 93–101
27. Siciliano B, Sciavicco L, Villani L, Oriolo G. Robotics: Modelling, Planning and Control. London: Springer Science & Business Media, 2009
28. Luh J Y S, Zheng Y F. Constrained relations between two coordinated industrial robots for motion control. International Journal of Robotics Research, 1987, 6(3): 60–70
29. Nakano E, Ozaki S, Ishida T, Kato I. Cooperational control of the anthropomorphous manipulator “MELARM”. In: Proceedings of the 4th International Symposium on Industrial Robots. 1974, 251–260