Robotica: (2020) volume 38, pp. 652–668. © Cambridge University Press 2019
doi:10.1017/S0263574719000924

Minimum Base Disturbance Control of Free-Floating Space Robot during Visual Servoing Pre-capturing Process

Xiaoyu Zhao†, Zongwu Xie†*, Haitao Yang† and Jiarui Liu‡

†*State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China*
‡*Vanke Meisha Academy, Shenzhen, China*

(Accepted May 25, 2019. First published online: July 12, 2019)

SUMMARY
During visual servoing space activities, the attitude of free-floating space robot may be disturbed due to dynamics coupling between the satellite base and the manipulator. And the disturbance may cause communication interruption between space robot and control center on earth. However, it often happens that the redundancy of manipulator is not enough to fully eliminate this disturbance. In this paper, a method named off-line optimizing visual servoing algorithm is innovatively proposed to minimize the base disturbance during the visual servoing process where the degrees-of-freedom of the manipulator is not enough for a zero-reaction control. Based on the characteristic of visual servoing process and the robot system modeling, the optimal control method is applied to achieve the optimization, and a pose planning method is presented to achieve a second-order continuity of quaternion getting rid of the interruption caused by ambiguity. Then simulations are carried out to verify the method, and the results show that the robot is controlled with optimized results during visual servoing process and the joint trajectories are smooth.

KEYWORDS: Free-floating space robot, Dual-arm capturing, Minimum base disturbance, Visual servoing, Optimal control.

1. Introduction

In order to accomplish the extra-vehicular activity in a safe and labor-saving way with less cost, space robots are employed as astronaut assistants. As a main type of space robots, the free-floating space robot (FFSR), which is composed of the space manipulator and satellite base, attracts much attention for its ability to reduce vibration and fuel consumption caused by jetting during the base pose adjusting process.[1] Due to the microgravity in space, there exists dynamics coupling between satellite base and space manipulator, and thus pose accuracy of the end-effector will be decreased if no proper control method is applied.[2,3] At the same time, to ensure the communication between space robot and control center on the earth, it is important to keep the space robot base pointing to the earth during the task,[4,5] and in other words the attitude of satellite base should be restricted to a limited range.

Different countries and organizations have done researches on base attitude disturbance elimination. Steven Dubowsky from Massachusetts Institute of Technology applied the Enhanced Disturbance Map to solve the dynamic disturbance problem.[6,7] Papadopoulos E of McGill University studied the FFSR dynamics and proposed the theory of Zero Reaction Manipulator.[4] However, the methods mentioned above are not suitable for practical application in space due to the demand of

* Corresponding author. E-mail: xiezongwu@hit.edu.cn

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

large storage space or the complex manipulator structure with more degrees-of-freedom (DOF). The Rapidly Exploring Random Trees was employed by Francis James to deal with nonholonomic constraint problem in space robot.[8] However, because of the time-consuming calculation process, the method cannot be applied to visual servoing system, while visual servoing system usually plays an important role in the rendezvous and docking tasks.

One of the most popular methods that can be used for the disturbance elimination in visual servoing system is the Reactionless Null Space (RNS). For example, Akio Gouo applied RNS to dual-arm space robot and successfully realized reactionless manipulation.[9] S. V. Shah did research on the post-capturing process with RNS and carried out simulations to verify the theory.[10] The method was also studied by Hishinuma T[11] and K. Yoshida[12] *et al*. However, if the redundant DOFs of the system are lower than the critical value, which is 1 DOF for planar robot and 3 DOF for three-dimensional robot, the method would not lead to a feasible solution. Besides, some researchers tried to reduce the reaction force exerted by the manipulator to the base, and they planned the trajectories that can generate null reaction force. However, the method is also accomplished on the basis of enough redundant DOFs.[13] S. V. Shah further optimized the base movement with quadratic optimization method.[14] However, the sudden changes in velocity may lead the manipulator out of control. Another optimization method is the least square method employed by Silvio Cocuzza,[15] and it is aimed to find the local optimal result of the system. However, this method suffers from the high computational complexity introduced by the explicit calculation of the inertia matrix.

In conclusion, attentions are mainly focused on either the off-line situation or space robots with enough redundant DOFs. However, the situations cannot be always satisfied. First, since visual servoing plays an important role in the rendezvous and docking, the proposed method should be able to be applied to visual servoing system. Second, during the space tasks, the redundant DOFs are usually used to achieve different tasks, such as trajectory tracking, singularity avoiding, and joint limit avoiding, and then the left available redundant DOFs for disturbance reduction is limited.

This paper tries to reduce the base disturbance in the visual servoing process with redundant DOFs fewer than critical value. The method proposed in this paper makes use of the difference between robot control period and visual sampling period, and the visual servoing process is optimized by the off-line optimization method. To achieve the proposed algorithm, the control process is analyzed and simplified. In addition, the influence of the visual sampling period on the optimization result is also studied, and some simulations are carried out to prove the theory proposed in the paper.

Compared with the existing methods, the method proposed in this paper combines the on-line visual servoing with the off-line optimization method. As a result, the method can accomplish more complex tasks in the space when there exists conflict between the requirement of real-time and enough redundant DOF. Besides, since the off-line optimized method is employed in the method, it is able to "predict" the robot movement in a visual servoing period, which makes the robot joint trajectories smoother.

The organization of this paper is as follows: Section 2 introduces the basic idea of the method proposed in this paper. Section 3 is devoted to the visual sampling process and proposes a method which can overcome the singularity of Roll-Pitch-Yaw (RPY) angle and achieve a second-order continuity of the pose planning. Section 4 develops the robot model and optimization method, and the control method proposed in Section 2 is also illustrated in detail. Section 5 deals with simulations that validate the theory proposed in this paper. Finally, the paper is concluded in Section 6.

2. The General Scheme of Space Robot Control

In this section, we propose a new algorithm named off-line optimizing visual servoing (OOVS) algorithm. The OOVS method is used to minimize the base disturbance of space robot that lacks enough redundant DOFs during visual servoing process. Compared with the traditional visual servoing method, the new method focuses on the difference between the visual sampling period and robot control period. The basic idea of the OOVS method is elaborated on as follows.

From the researches that have been done, it can be concluded that the visual sampling and image processing are time-consuming, and the visual sampling frequency is usually 5 Hz,[16,17] while the robot control frequency is 500 Hz or higher. One of the methods to coordinate the frequencies is to apply the interpolation method to the visual servoing process.[18] Considering the fact that the

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

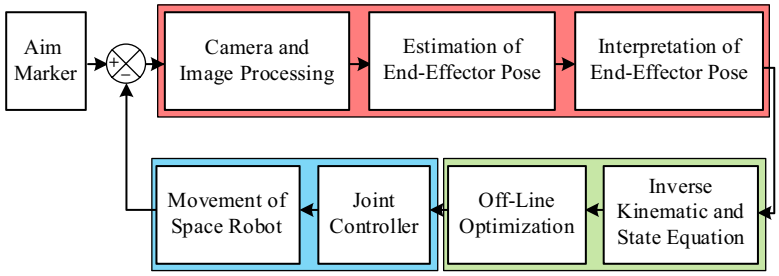


Fig. 1. The optimization algorithm scheme in the visual servoing system.

interpolation process plans the motion off-line, some off-line optimization algorithms can be applied to the visual servoing process.

The OOVS method can be accomplished by following the procedure shown in Fig. 1. The procedure is divided into three parts, including the visual servoing part (the red part), the optimization part (the green part), and the robot control part (the blue part). To accomplish the procedure, first, the relative pose between the aim marker and the robot end marker is measured, and a new pose is estimated for the next sampling moment. Next the estimated pose is used in the interpolation process to plan the desired end trajectory. Then based on the interpolation results and the space robot inverse kinematic, the robot model is built and the off-line optimization method is applied. Finally, the robot joint controller moves the robot until next period starts.

It can be concluded from the statements that the OOVS methods are accomplished based on the new idea of combining the on-line visual servoing with the off-line optimization method. And the two contradicting characteristics are combined by making full use of the difference of robot control period and visual sampling period. It would be proved in the following sections that the method completes the end pose planning process, decreases the calculation frequency of the time-consuming variables, and thus increases the computing efficiency. In addition, the method thus obtains smoother joint trajectories.

To accomplish the OOVS method, more details will be discussed. In the following sections, the visual servoing parts and the optimization parts are mainly concerned, and the key points are presented as follows:

- In visual servoing part, the paper aims to improve the method proposed by Han,[19] which will remove the ambiguity in the original method and is expanded to second order continuity.
- The optimization part can be finished by two steps, the first step is to build up the state equation which describes the FFSR system; the second one is to apply the optimization process to the equation.

3. The Visual Servoing Planning of End Pose

As mentioned above, to accomplish the OOVS method, the visual servoing process is the first step. Considering the representation singularity or named gimbal lock that may be introduced by Euler angle,[20] quaternion is chosen as the descriptor of end attitude, and thus the estimation of the quaternion in visual servoing process is introduced. Besides, since the robot control period is much shorter than the visual sampling period, interpolation is applied to the estimated result and a new second-order continuity interpolation method for the quaternion is proposed.

3.1. The estimation of end pose

In this paper, the position-based visual servoing (PBVS) is selected for end pose planning. According to François Chaumette and Peter Corke,[21–23] the end spatial velocity during the PBVS process is:

$$\tilde{\boldsymbol{v}}_{\mathrm{e}} = \rho\, \boldsymbol{e}_{\mathrm{ae}} \tag{1}$$

where $\tilde{\boldsymbol{v}}_{\mathrm{e}} = (\boldsymbol{v}_{\mathrm{e}}^{\mathrm{T}}, \boldsymbol{\omega}_{\mathrm{e}}^{\mathrm{T}})^{\mathrm{T}}$ is the spatial velocity, ρ is the coefficient that shows the relationship between the spatial velocity and the pose error, and $\boldsymbol{e}_{\mathrm{ae}}$ is the error between the robot end pose and the aim pose.

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

Suppose the pose vector is denoted by $\boldsymbol{s} = (\boldsymbol{p}^{\mathrm{T}}, \boldsymbol{q}^{\mathrm{T}})^{\mathrm{T}}$, where $\boldsymbol{p}$ is the displacement vector and $\boldsymbol{q} = (\eta, \boldsymbol{\epsilon}^{\mathrm{T}})^{\mathrm{T}}$ is the quaternion vector, then the pose error can be obtained by Eq. (2):

$$\boldsymbol{e}_{\mathrm{ae}} = \mathrm{f}(\boldsymbol{s}_{\mathrm{a}}, \boldsymbol{s}_{\mathrm{e}}) = \begin{pmatrix} \boldsymbol{p}_{\mathrm{a}} - \boldsymbol{p}_{\mathrm{e}} \\ \eta_{\mathrm{e}}\boldsymbol{\epsilon}_{\mathrm{a}} - \eta_{\mathrm{d}}\boldsymbol{\epsilon}_{\mathrm{a}} - \boldsymbol{\epsilon}_{\mathrm{a}} \times \boldsymbol{\epsilon}_{\mathrm{e}} \end{pmatrix} \tag{2}$$

where $\boldsymbol{s}_{\mathrm{a}}$ and $\boldsymbol{s}_{\mathrm{e}}$ are the aim and the real-time pose; $\mathrm{f}(\cdot)$ maps the poses to pose error.

It is considerable to assume that $\tilde{\boldsymbol{v}}_{\mathrm{e}}$ keeps constant during the estimation, that is:

$$\boldsymbol{s}_T = \begin{pmatrix} \boldsymbol{p}_{\mathrm{e}} + \boldsymbol{v}_{\mathrm{e}} T \\ \mathrm{e}^{\frac{1}{2}\boldsymbol{\omega}_{\mathrm{e}} T} \otimes \boldsymbol{q}_{\mathrm{e}} \end{pmatrix} \tag{3}$$

where $\boldsymbol{s}_T$ is the pose at time T, and the value of T equals to the visual sampling period, $\otimes$ denotes the quaternion product.

3.2. The interpolation of end pose

In this paper, polynomial of degree five is chosen for the interpolation of displacement vector, which can keep the trajectory smooth enough. And the polynomial is shown as follows:

$$\boldsymbol{p}_t = \boldsymbol{k}_0 + \boldsymbol{k}_1(t - t_0) + \boldsymbol{k}_2(t - t_0)^2 + \boldsymbol{k}_3(t - t_0)^3 + \boldsymbol{k}_4(t - t_0)^4 + \boldsymbol{k}_5(t - t_0)^5 \tag{4}$$

where $\boldsymbol{k}_0 = \boldsymbol{p}_{\mathrm{e}}$, $\boldsymbol{k}_1 = \boldsymbol{v}_{\mathrm{e}}/2$, $\boldsymbol{k}_2 = \boldsymbol{a}_{\mathrm{e}}$, $\boldsymbol{k}_3 = \left[20\boldsymbol{h} - (12\boldsymbol{v}_{\mathrm{e}} + 8\boldsymbol{v}_T)\, T - (3\boldsymbol{a}_{\mathrm{e}} - \boldsymbol{a}_T)\, T^2\right]/(2T^3)$, $\boldsymbol{k}_4 = \left[-30\boldsymbol{h} + (16\boldsymbol{v}_{\mathrm{e}} + 14\boldsymbol{v}_T)\, T + (3\boldsymbol{a}_{\mathrm{e}} - 2\boldsymbol{a}_T)\, T^2\right]/(2T^4)$, $\boldsymbol{k}_5 = \left[12\boldsymbol{h} - 6\,(\boldsymbol{v}_{\mathrm{e}} + \boldsymbol{v}_T)\, T - (\boldsymbol{a}_{\mathrm{e}} - \boldsymbol{a}_T)\, T^2\right]/(2T^5)$.

As for the quaternion, researches that have been done cannot ensure the continuity in the angular-acceleration level.[19,24–26] In this paper, the method proposed by Han[19] is improved, and the angular-acceleration level planning is achieved. Meanwhile, the ambiguity is also removed.

Considering the following equations:

$$\boldsymbol{\psi} = \phi \boldsymbol{n} \tag{5}$$

$$\dot{\boldsymbol{\psi}} = \left[\mathbf{I}_{3\times 3} + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)(\boldsymbol{n}\times\ \boldsymbol{n}\times) - \frac{\phi}{2}(\boldsymbol{n}\times)\right]\boldsymbol{\omega}_{\mathrm{e}} \tag{6}$$

where $\boldsymbol{\psi}$ is the angle-axis that represents the same attitude noted by quaternion; $\mathbf{I}_{3\times 3}$ is the 3×3 unit matrix; $\boldsymbol{\omega}_{\mathrm{e}}$ is the end angular velocity; ϕ and $\boldsymbol{n}$ are the rotation angle and rotation axis in $\boldsymbol{\psi}$, respectively, which can be obtained from quaternion by Eqs. (7) and (8); and $\boldsymbol{n}\times$ is the skew-symmetric matrix of vector, which is defined as Eq. (9).

$$\phi = \begin{cases} 2\arccos(\eta) & \boldsymbol{q}_{\mathrm{e}}^{\mathrm{T}}\boldsymbol{q}_T > 0 \\ 2\arccos(\eta) - 2\pi & \boldsymbol{q}_{\mathrm{e}}^{\mathrm{T}}\boldsymbol{q}_T < 0 \end{cases} \tag{7}$$

$$\boldsymbol{n} = \frac{\boldsymbol{\epsilon}}{\|\boldsymbol{\epsilon}\|} \tag{8}$$

$$\boldsymbol{n}\times = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} \tag{9}$$

In Eq. (7), $\boldsymbol{q}_{\mathrm{e}}$ and $\boldsymbol{q}_T$ are the quaternion at current and next sampling moment, respectively. And the values listed in Eq. (7) match with two possible rotation directions. As a result, $\boldsymbol{q}_{\mathrm{e}}^{\mathrm{T}}\boldsymbol{q}_T$ is employed as the selector of the rotation direction, and it ensures a minimum rotation angle to eliminate the ambiguity.

If Eq. (6) is taken a derivative, the following equation can be obtained:

$$\begin{aligned} \ddot{\boldsymbol{\psi}} = \Bigg[&\left(-\frac{1}{2}\cot\frac{\phi}{2} + \frac{\phi}{4}\csc^2\frac{\phi}{2}\right)\dot{\phi}\,(\boldsymbol{n}\times\ \boldsymbol{n}\times) + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)(\dot{\boldsymbol{n}}\times\ \boldsymbol{n}\times + \boldsymbol{n}\times\ \dot{\boldsymbol{n}}\times) \\ &- \frac{\dot{\phi}}{2}(\boldsymbol{n}\times) - \frac{\phi}{2}(\dot{\boldsymbol{n}}\times)\Bigg]\boldsymbol{\omega}_{\mathrm{e}} + \left[\mathbf{I}_{3\times 3} + \left(1 - \frac{\phi}{2}\cot\frac{\phi}{2}\right)(\boldsymbol{n}\times\ \boldsymbol{n}\times) - \frac{\phi}{2}(\boldsymbol{n}\times)\right]\boldsymbol{\alpha}_{\mathrm{e}} \end{aligned} \tag{10}$$

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

where $\dot{\phi}$ and $\dot{\boldsymbol{n}}$ can be obtained by Eqs. (11) and (12):

$$\dot{\phi} = \boldsymbol{\omega}_{\mathrm{e}}^{\mathrm{T}}\boldsymbol{n} = \dot{\boldsymbol{\psi}}^{\mathrm{T}}\boldsymbol{n} = \frac{\dot{\boldsymbol{\psi}}^{\mathrm{T}}\boldsymbol{\psi}}{\|\boldsymbol{\psi}\|} \tag{11}$$

$$\dot{\boldsymbol{n}} = \frac{1}{\phi}\left(\dot{\boldsymbol{\psi}} - \frac{\dot{\boldsymbol{\psi}}^{\mathrm{T}}\boldsymbol{\psi}}{\phi}\boldsymbol{n}\right) = -\frac{1}{2}\left[\cot\frac{\phi}{2}(\boldsymbol{n}\times\ \boldsymbol{n}\times) + (\boldsymbol{n}\times)\right]\boldsymbol{\omega}_{\mathrm{e}} \tag{12}$$

Suppose $\boldsymbol{Q} = \left(\boldsymbol{q}^{\mathrm{T}}, \boldsymbol{\omega}^{\mathrm{T}}, \boldsymbol{\alpha}^{\mathrm{T}}\right)^{\mathrm{T}}$ and $\boldsymbol{\Psi} = \left(\boldsymbol{\psi}^{\mathrm{T}}, \dot{\boldsymbol{\psi}}^{\mathrm{T}}, \ddot{\boldsymbol{\psi}}^{\mathrm{T}}\right)^{\mathrm{T}}$, then $\boldsymbol{Q}$ can be mapped into $\boldsymbol{\Psi}$ by Eqs. (5), (6), and (10). To finish the pose plan, it is necessary to map $\boldsymbol{\Psi}$ back to $\boldsymbol{Q}$ by the following equations:

$$\boldsymbol{q} = \left(\cos\frac{\phi}{2} \quad \sin\frac{\phi}{2}\boldsymbol{n}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{13}$$

$$\boldsymbol{\omega}_{\mathrm{e}} = \left[\mathbf{I}_{3\times3} + \frac{1-\cos\phi}{\phi}(\boldsymbol{n}\times) + \frac{\phi-\sin\phi}{\phi}(\boldsymbol{n}\times\ \boldsymbol{n}\times)\right]\dot{\boldsymbol{\psi}} \tag{14}$$

$$\begin{aligned}\boldsymbol{\alpha}_{\mathrm{e}} = &\left[\frac{\dot{\phi}}{\phi^2}\left(\phi\sin\phi - 2 + 2\cos\phi\right)(\boldsymbol{n}\times) + \frac{\dot{\phi}}{\phi^2}\left(3\sin\phi - \phi\cos\phi - 2\phi\right)(\boldsymbol{n}\times\ \boldsymbol{n}\times)\right]\dot{\boldsymbol{\psi}} \\ &+ \frac{\phi-\sin\phi}{\phi^2}\dot{\boldsymbol{\psi}}\times\ (\boldsymbol{n}\times\dot{\boldsymbol{\psi}}) + \left[\mathbf{I}_{3\times3} + \frac{1-\cos\phi}{\phi}(\boldsymbol{n}\times) + \frac{\phi-\sin\phi}{\phi}(\boldsymbol{n}\times\ \boldsymbol{n}\times)\right]\ddot{\boldsymbol{\psi}}\end{aligned} \tag{15}$$

By virtue of Eqs. (5), (6), (10) and (13), (14), (15), the interpolation for quaternion can be achieved by the following process. First, attitudes at present and the next sampling moment expressed by $\boldsymbol{Q}$ are converted to $\boldsymbol{\Psi}$. Then $\boldsymbol{\Psi}$ is interpolated by polynomial of degree five, just as shown in Eq. (4). Finally, the interpolation results are converted to the $\boldsymbol{Q}$ again for the robot control.

The interpolation process expands the existing methods and is able to realize a better trajectory without the ambiguity. Besides, the method can achieve acceleration-level continuity for the quaternion, and thus helps with further studies.

To sum up, the visual servoing process is shown in the flow diagram in Fig. 2, where end pose $\boldsymbol{s}_{\mathrm{e}}$ is substituted by $\boldsymbol{s}_{\mathrm{e0}}$ to represent the pose at visual sampling moment. The variable $\boldsymbol{P}$ is defined as $\boldsymbol{P} = \left(\boldsymbol{p}^{\mathrm{T}}, \boldsymbol{v}^{\mathrm{T}}, \boldsymbol{a}^{\mathrm{T}}\right)^{\mathrm{T}}$, and represents the robot end displacement.

4. The Optimization of Space Robot Base Disturbance

After the pose planning is finished, it is necessary to build the space robot model and apply the optimization method to achieve the OOVS method.

To build the space model, we expand the method proposed by Sunil Gupta[27] and apply it to the FFSR. When applying the trajectory planning, the improved method expands the end 3D position vector to the 6D spatial vector (including the position and attitude), which is more common for robot control. Moreover, the new method further reduces the model complexity and increases the computing efficiency.

Since the optimization method is applied to the visual servoing process, it is necessary for the optimization process to be efficient. Thus, the optimal control method is selected for base disturbance optimization.

4.1. The space robot state model

When the robot tracking trajectory is controlled by second-order model, the tracking error satisfies the following equation:[28]

$$\Delta\tilde{\boldsymbol{a}}_{\mathrm{de}} + \boldsymbol{D}\Delta\tilde{\boldsymbol{v}}_{\mathrm{de}} + \boldsymbol{K}\boldsymbol{e}_{\mathrm{de}} = 0 \tag{16}$$

where $\tilde{\boldsymbol{a}}$ is the end spatial acceleration, and $\Delta\tilde{\boldsymbol{a}}_{\mathrm{de}} = \left(\boldsymbol{a}_{\mathrm{d}}^{\mathrm{T}} - \boldsymbol{a}_{\mathrm{e}}^{\mathrm{T}}, \boldsymbol{\alpha}_{\mathrm{d}}^{\mathrm{T}} - \boldsymbol{\alpha}_{\mathrm{e}}^{\mathrm{T}}\right)^{\mathrm{T}}$ is the error between desired spatial acceleration and real spatial acceleration; $\Delta\tilde{\boldsymbol{v}}_{\mathrm{de}} = \left(\boldsymbol{v}_{\mathrm{d}}^{\mathrm{T}} - \boldsymbol{v}_{\mathrm{e}}^{\mathrm{T}}, \boldsymbol{\omega}_{\mathrm{d}}^{\mathrm{T}} - \boldsymbol{\omega}_{\mathrm{e}}^{\mathrm{T}}\right)^{\mathrm{T}}$ is the error between desired spatial velocity and real spatial velocity; $\boldsymbol{e}_{\mathrm{de}}$ can be obtained by Eq. (2), except that $\boldsymbol{s}_{\mathrm{a}}$ is substituted by $\boldsymbol{s}_{\mathrm{d}}$.

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

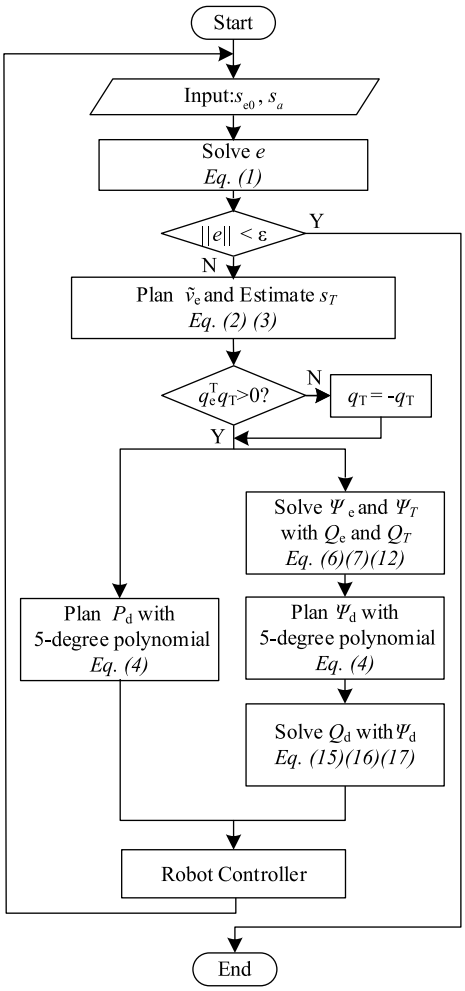


Fig. 2. The flow diagram of visual servoing process.

Consider second-order differential kinematic equation

$$\tilde{\boldsymbol{a}}_e = \dot{\boldsymbol{J}}_G\dot{\boldsymbol{\theta}} + \boldsymbol{J}_G\ddot{\boldsymbol{\theta}} \tag{17}$$

where $\boldsymbol{J}_G$ is general Jacobian; $\dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$ are the joint velocity and acceleration, respectively.

Then $\ddot{\boldsymbol{\theta}}$ can be solved by:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}_G^{\dagger}\left(\tilde{\boldsymbol{a}}_e - \dot{\boldsymbol{J}}_G\dot{\boldsymbol{\theta}}\right) + \left(\mathbf{I} - \boldsymbol{J}_G^{\dagger}\boldsymbol{J}_G\right)\boldsymbol{u} \tag{18}$$

where $\boldsymbol{J}_G^{\dagger}$ is the pseudo-inverse of $\boldsymbol{J}_G$; $\boldsymbol{u}$ is a random vector and is chosen as the control vector.

Substituting Eq. (16) into Eq. (18) leads to the following equation.

$$\ddot{\boldsymbol{\theta}} = -\boldsymbol{J}_G^{\dagger}\dot{\boldsymbol{J}}_G\dot{\boldsymbol{\theta}} - \boldsymbol{D}\boldsymbol{J}_G^{\dagger}\boldsymbol{J}_G\dot{\boldsymbol{\theta}} + \left(\mathbf{I} - \boldsymbol{J}_G^{\dagger}\boldsymbol{J}_G\right)\boldsymbol{u} + \boldsymbol{J}_G^{\dagger}(\tilde{\boldsymbol{a}}_d + \boldsymbol{D}\tilde{\boldsymbol{v}}_d + \boldsymbol{K}\boldsymbol{e}_{de}) \tag{19}$$

Equation (19) is the state equation of space robot. It is worthy to be noted that Eq. (19) considers the pose of the robot end, while Sunil Gupta considers only the position.[27] As a result, though Eq. (19) looks like the equation shown by Sunil Gupta, the meanings are not exactly the same, especially for $\boldsymbol{e}_{de}$.

4.2. Simplification of space robot model

4.2.1. The linearization of space robot model. Considering that $\dot{\boldsymbol{J}}_G$ is a nonlinear function with respect to the term $\dot{\boldsymbol{\theta}}$, in order to reduce the control complexity, it is necessary to linearize the equation. Considering the fact that the visual sampling period is short enough, the Taylor Series

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

can be applied to $\boldsymbol{J}_{\mathrm{G}}^{\dagger}\dot{\boldsymbol{J}}_{\mathrm{G}}$ in Eq. (19), and only the first-order term is reserved. Then the following equation can be obtained:

$$\ddot{\boldsymbol{\theta}} \approx -\left[\frac{\partial\left(\boldsymbol{J}_{\mathrm{G}}^{\dagger}\dot{\boldsymbol{J}}_{\mathrm{G}}\dot{\boldsymbol{\theta}}\right)}{\partial\dot{\boldsymbol{\theta}}}\right]_{\dot{\boldsymbol{\theta}}_0}\left(\dot{\boldsymbol{\theta}}-\dot{\boldsymbol{\theta}}_0\right)-\boldsymbol{D}\boldsymbol{J}_{\mathrm{G}}^{\dagger}\boldsymbol{J}_{\mathrm{G}}\dot{\boldsymbol{\theta}}+\left(\mathbf{I}-\boldsymbol{J}_{\mathrm{G}}^{\dagger}\boldsymbol{J}_{\mathrm{G}}\right)\boldsymbol{u}+\boldsymbol{J}_{\mathrm{G}}^{\dagger}(\tilde{\boldsymbol{a}}_{\mathrm{d}}+\boldsymbol{D}\tilde{\boldsymbol{v}}_{\mathrm{d}}+\boldsymbol{K}\boldsymbol{e}_{\mathrm{de}}) \tag{20}$$

Since the variable $\boldsymbol{\theta}$ is obtained by the integration of $\dot{\boldsymbol{\theta}}$, $\boldsymbol{J}_{\mathrm{G}}$ and $\boldsymbol{s}$ which are the functions of $\boldsymbol{\theta}$ should be further simplified. Because the sampling period is short, the simplification is shown as follows:

$$\boldsymbol{J}_{\mathrm{G}}=\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t+o(t)\approx\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t \tag{21}$$

$$\begin{aligned}\boldsymbol{s}_{\mathrm{e}}&=\begin{pmatrix}\boldsymbol{p}_{\mathrm{e0}}+\boldsymbol{v}_{\mathrm{e0}}t+o(t)\\ \left[1+\frac{1}{2}\boldsymbol{\omega}_{\mathrm{e0}}t+o(t)\right]\otimes\boldsymbol{q}_{\mathrm{e0}}\end{pmatrix}\approx\begin{pmatrix}\boldsymbol{p}_{\mathrm{e0}}+\boldsymbol{v}_{\mathrm{e0}}t\\ \boldsymbol{q}_{\mathrm{e0}}+\frac{1}{2}\boldsymbol{\omega}_{\mathrm{e0}}\otimes\boldsymbol{q}_{\mathrm{e0}}t\end{pmatrix}\\&=\boldsymbol{s}_{\mathrm{e0}}+\begin{bmatrix}\mathbf{0}_{3\times3}&\mathbf{0}_{3\times1}\\ \mathbf{0}_{4\times3}&\mathbf{c}(\boldsymbol{q})\end{bmatrix}\tilde{\boldsymbol{v}}_{\mathrm{e0}}t=\boldsymbol{s}_{\mathrm{e0}}+\boldsymbol{C}\tilde{\boldsymbol{v}}_{\mathrm{e0}}t\end{aligned} \tag{22}$$

where o(·) represents the higher order of the corresponding variable, **0**s denote the zero matrix, and $\boldsymbol{c}(\boldsymbol{q})$ is the matrix that can convert angular velocity to the differential of quaternion, which is defined by Eq. (23).

$$\boldsymbol{c}(\boldsymbol{q})=\begin{bmatrix}-\epsilon_1&-\epsilon_2&-\epsilon_3\\ \eta&\epsilon_3&-\epsilon_2\\ -\epsilon_3&\eta&\epsilon_1\\ \epsilon_2&-\epsilon_1&\eta\end{bmatrix} \tag{23}$$

Substituting Eqs.(21) and (22) into Eq. (20), the following conclusion can be obtained:

$$\dot{\boldsymbol{x}}=\boldsymbol{A}(t)\boldsymbol{x}+\boldsymbol{B}(t)\boldsymbol{u}+\boldsymbol{\xi} \tag{24}$$

where $\boldsymbol{x}=\dot{\boldsymbol{\theta}}$ and $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, and $\boldsymbol{\xi}$ are defined by the following equations:

$$\boldsymbol{A}(t)=-\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)^{\dagger}\left[\frac{\partial\left(\dot{\boldsymbol{J}}_{\mathrm{G0}}\boldsymbol{x}\right)}{\partial\boldsymbol{x}}\right]_{\boldsymbol{x}_0}-\boldsymbol{D}\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)^{\dagger}\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right) \tag{25}$$

$$\boldsymbol{B}(t)=\left[\mathbf{I}-\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)^{\dagger}\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)\right] \tag{26}$$

$$\begin{aligned}\boldsymbol{\xi}=&\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)^{\dagger}\cdot\left[\tilde{\boldsymbol{a}}_{\mathrm{d}}+\boldsymbol{D}\tilde{\boldsymbol{v}}_{\mathrm{d}}+\boldsymbol{K}\,\mathrm{f}(\boldsymbol{s}_{\mathrm{d}},\boldsymbol{s}_{\mathrm{e0}}+\boldsymbol{C}\tilde{\boldsymbol{v}}_{\mathrm{e0}}t)-\dot{\boldsymbol{J}}_{\mathrm{G0}}\boldsymbol{x}_0\right]\\&+\left(\boldsymbol{J}_{\mathrm{G0}}+\dot{\boldsymbol{J}}_{\mathrm{G0}}t\right)^{\dagger}\left[\frac{\partial\left(\dot{\boldsymbol{J}}_{\mathrm{G0}}\boldsymbol{x}\right)}{\partial\boldsymbol{x}}\right]_{\boldsymbol{x}_0}\boldsymbol{x}_0\end{aligned} \tag{27}$$

4.2.2. Numerical solution of the differential term. In the practical situation, the numerical solutions of $\dot{\boldsymbol{J}}_{\mathrm{G}}$ and $\partial(\dot{\boldsymbol{J}}_{\mathrm{G}}\boldsymbol{x})/\partial\boldsymbol{x}$ can be obtained much easier than the analytic one.[29] And two feasible numerical methods are listed and compared below.

Method 1: Since the general Jacobian of the FFSR is a function of base attitude and joint angle, it can be expressed as $\boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta},\boldsymbol{\sigma})$, where $\boldsymbol{\sigma}$ is the RPY angle of the base. Thus, $\dot{\boldsymbol{J}}_{\mathrm{G}}$ can be approximated by the following equation:

$$\dot{\boldsymbol{J}}_{\mathrm{G}}\left(\boldsymbol{\theta},\boldsymbol{\sigma},\dot{\boldsymbol{\theta}}\right)\approx\frac{\boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta}+\dot{\boldsymbol{\theta}}\delta t,\boldsymbol{\sigma}+\dot{\boldsymbol{\sigma}}\delta t)-\boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta},\boldsymbol{\sigma})}{\delta t} \tag{28}$$

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

where

$$\dot{\boldsymbol{\sigma}} = \begin{pmatrix} \dot{\sigma}_1 \\ \dot{\sigma}_2 \\ \dot{\sigma}_3 \end{pmatrix} = \boldsymbol{J}_{\mathrm{b}\sigma}\dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 & -\sin(\sigma_1) & \cos(\sigma_1)\cos(\sigma_2) \\ 0 & \cos(\sigma_1) & \sin(\sigma_1)\cos(\sigma_2) \\ 1 & 0 & -\sin(\sigma_2) \end{bmatrix} \boldsymbol{J}_{\mathrm{b}\omega}\dot{\boldsymbol{\theta}} \tag{29}$$

where $\dot{\sigma}_i$ and σ_i are the ith terms of $\dot{\boldsymbol{\sigma}}$ and $\boldsymbol{\sigma}$, respectively; $\boldsymbol{J}_{\mathrm{b}\sigma}$ and $\boldsymbol{J}_{\mathrm{b}\omega}$ are the analytical Jacobian and geometric Jacobian with respect to the base attitude, respectively.[20]

Furthermore, the following equation is obtained:

$$\frac{\partial \dot{\boldsymbol{J}}_{\mathrm{G}}\dot{\boldsymbol{\theta}}}{\partial \dot{\boldsymbol{\theta}}} = \left[\frac{\partial \dot{\boldsymbol{J}}_{\mathrm{G}}\dot{\boldsymbol{\theta}}}{\partial \dot{\theta}_k}\right] \approx \left[\frac{\dot{\boldsymbol{J}}_{\mathrm{G}}(\boldsymbol{\theta}, \boldsymbol{\sigma}, \dot{\boldsymbol{\theta}} + \delta\dot{\boldsymbol{\theta}}_k)\cdot(\dot{\boldsymbol{\theta}} + \delta\dot{\boldsymbol{\theta}}_k) - \dot{\boldsymbol{J}}_{\mathrm{G}}(\boldsymbol{\theta}, \boldsymbol{\sigma}, \dot{\boldsymbol{\theta}})\cdot\dot{\boldsymbol{\theta}}}{\|\delta\dot{\boldsymbol{\theta}}_k\|}\right] \tag{30}$$

where $[\cdot]$ means the matrix with the variable acting as the column vector, and $\delta\dot{\boldsymbol{\theta}}_i$ represents the vector that all terms except the ith one in the vector equals to zero and the ith term is $\|\delta\dot{\boldsymbol{\theta}}_i\|$.

Method 2:

Again, since the general Jacobian of the FFSR is a function of base attitude and joint angle, $\dot{\boldsymbol{J}}_{\mathrm{G}}$ can be obtained as follows:

$$\dot{\boldsymbol{J}}_{\mathrm{G}} = \sum_{i=1}^{3}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \sigma_i}\dot{\sigma}_i + \sum_{k=1}^{N}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \theta_k}\dot{\theta}_k \tag{31}$$

where θ_k is the angle of the kth joint, and N is the total angle number of the robot.

Substituting Eq. (29) into Eq. (31), it can be obtained that

$$\dot{\boldsymbol{J}}_{\mathrm{G}} = \sum_{k=1}^{N}\sum_{i=1}^{3}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \sigma_i}j_{ik}\dot{\theta}_k + \sum_{k=1}^{N}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \theta_k}\dot{\theta}_k \tag{32}$$

where j_{ik} is the term that lies on the ith row and kth column of $\boldsymbol{J}_{\mathrm{b}\alpha}$.

Thus, the following equation can be obtained:

$$\frac{\partial \dot{\boldsymbol{J}}_{\mathrm{G}}\dot{\boldsymbol{\theta}}}{\partial \dot{\boldsymbol{\theta}}} = \dot{\boldsymbol{J}}_{\mathrm{G}} + \left[\hat{\boldsymbol{J}}_k\right] = \sum_{k=1}^{N}\sum_{i=1}^{3}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \sigma_i}j_{ik}\dot{\theta}_k + \sum_{k=1}^{N}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \theta_k}\dot{\theta}_k + \left[\hat{\boldsymbol{J}}_k\right] \tag{33}$$

where $\hat{\boldsymbol{J}}_k$ is defined as in Eq. (34) and $(\partial \boldsymbol{J}_{\mathrm{G}}/\partial \boldsymbol{\theta}_k)$ and $(\partial \boldsymbol{J}_{\mathrm{G}}/\partial \boldsymbol{\sigma}_i)$ can be calculated by Eqs. (35) and (36).

$$\hat{\boldsymbol{J}}_k = \left(\sum_{i=1}^{3}\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \sigma_i}j_{ik} + \frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \theta_k}\right)\dot{\boldsymbol{\theta}} \tag{34}$$

$$\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \sigma_i} \approx \frac{\boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta}, \boldsymbol{\sigma} + \delta\boldsymbol{\sigma}_i) - \boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta}, \boldsymbol{\sigma})}{\|\delta\boldsymbol{\sigma}_i\|} \tag{35}$$

$$\frac{\partial \boldsymbol{J}_{\mathrm{G}}}{\partial \theta_k} \approx \frac{\boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}_k, \boldsymbol{\sigma}) - \boldsymbol{J}_{\mathrm{G}}(\boldsymbol{\theta}, \boldsymbol{\sigma})}{\|\delta\boldsymbol{\theta}_k\|} \tag{36}$$

When the two methods are compared, the following conclusions can be obtained: the calculation process of method 1 is more efficient than the other one, and the calculation times of the general Jacobian that consumes the most is $N + 2$, while method 2 is $N + 4$. However, the difference quotient in method 1 is calculated in series, and thus subtle variations in the variable δ may lead to large rounding error or truncation error. In method 2, the corresponding calculated process is independent, and thus it can convert to a precise result more easily, in other words, there is a larger range of choice for δ.

4.3. Optimization algorithm and the OOVS method

After the space robot end task and the state equation are defined, the optimization method can be applied to the system. Before the optimization process, a proper cost function is selected.

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

To minimize the base attitude disturbance, the base angular velocity is chosen as the cost function, that is:

$$\Omega = \frac{1}{2}\int_{t_0}^{t_0+T} \left\{ [\boldsymbol{J}_{b\omega}(\boldsymbol{\theta}, \boldsymbol{\sigma})\boldsymbol{x}(t)]^{\mathrm{T}} \boldsymbol{O} [\boldsymbol{J}_{b\omega}(\boldsymbol{\theta}, \boldsymbol{\sigma})\boldsymbol{x}(t)] + \boldsymbol{u}(t)^{\mathrm{T}} \boldsymbol{R}\boldsymbol{u}(t) \right\} \mathrm{d}t \tag{37}$$

where $\boldsymbol{O}$ and $\boldsymbol{R}$ are both positive defined matrices.

Then, if $\boldsymbol{J}_{b\omega}(\boldsymbol{\theta}, \boldsymbol{\sigma})$ is estimated by the Eq. (38), which is a function of time t, Eq. (39) can be derived from Eq. (37).

$$\boldsymbol{J}_{b\omega}(\boldsymbol{\theta}, \boldsymbol{\sigma}) = \boldsymbol{J}_{b\omega 0} + \dot{\boldsymbol{J}}_{b\omega 0} t + o(t) \approx \boldsymbol{J}_{b\omega 0} + \dot{\boldsymbol{J}}_{b\omega 0} t = \boldsymbol{J}_{b\omega}(t) \tag{38}$$

$$\Omega = \frac{1}{2}\int_{t_0}^{t_0+T} \left\{ [\boldsymbol{J}_{b\omega}(t)\boldsymbol{x}(t)]^{\mathrm{T}} \boldsymbol{O} [\boldsymbol{J}_{b\omega}(t)\boldsymbol{x}(t)] + \boldsymbol{u}(t)^{\mathrm{T}} \boldsymbol{R}\boldsymbol{u}(t) \right\} \mathrm{d}t \tag{39}$$

Since $\boldsymbol{O}$ is positive defined, $\boldsymbol{J}_{b\omega}(t)^{\mathrm{T}}\boldsymbol{O}\boldsymbol{J}_{b\omega}(t)$ can be obtained as a positive semi-definite matrix. As a result, Eq. (39) can be employed as the cost function of optimal control method.[30]

To apply the optimal method, the Hamiltonian can be defined as:

$$H = \frac{1}{2}(\boldsymbol{J}_{b\omega}\boldsymbol{x})^{\mathrm{T}}\boldsymbol{O}(\boldsymbol{J}_{b\omega}\boldsymbol{x}) + \frac{1}{2}\boldsymbol{u}(t)^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}(t) + \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{\xi}) \tag{40}$$

where all the variables except $\boldsymbol{O}$ and $\boldsymbol{R}$ are the function of time t, and H is the functional of $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{\lambda}$.

Suppose $\boldsymbol{x}^*$, $\boldsymbol{u}^*$, and $\boldsymbol{\lambda}^*$ are the optimal state vector, control vector, and Lagrange multipliers, respectively, then:

$$\dot{\boldsymbol{x}}^* = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{B}\boldsymbol{u}^* + \boldsymbol{\xi} \tag{41}$$

$$\dot{\boldsymbol{\lambda}}^* = -\frac{\partial H}{\partial \boldsymbol{x}} = -\boldsymbol{J}_{b\omega}^{\mathrm{T}}\boldsymbol{O}\boldsymbol{J}_{b\omega}\boldsymbol{x}^* - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\lambda}^* \tag{42}$$

$$\boldsymbol{0} = \frac{\partial H}{\partial \boldsymbol{u}} = \boldsymbol{R}\boldsymbol{u}^* + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\lambda}^* \tag{43}$$

According to Donald E. Kirk, it is reasonable to obtain $\boldsymbol{\lambda}^*$ from the following equation:[30]

$$\boldsymbol{\lambda}^* = \boldsymbol{K}\boldsymbol{x}^* + \boldsymbol{\gamma} \tag{44}$$

then Eqs. (41)–(44) give:

$$\boldsymbol{u}^* = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\left(\boldsymbol{K}\boldsymbol{x}^* + \boldsymbol{\gamma}\right) \tag{45}$$

$$\dot{\boldsymbol{K}} = -\boldsymbol{A}^{\mathrm{T}}\boldsymbol{K} + \boldsymbol{K}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{A} - \boldsymbol{J}_{b\omega}^{\mathrm{T}}\boldsymbol{O}\boldsymbol{J}_{b\omega} \tag{46}$$

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{K}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\gamma} - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\gamma} - \boldsymbol{K}\boldsymbol{\xi} \tag{47}$$

besides, the boundary condition is $\boldsymbol{K}_T = \boldsymbol{0}$ and $\boldsymbol{\gamma}_T = \boldsymbol{0}$.

When $\boldsymbol{K}$ and $\boldsymbol{\gamma}$ are derived by numerical integration, the control law $\boldsymbol{u}^*$ will be obtained. With $\boldsymbol{u} = \boldsymbol{u}^*$, Eq. (19) can be employed to calculate the joint acceleration and it is the input of space robot dynamic control.

4.4. The detailed scheme of OOVS algorithm

Since the visual servoing part and optimization part have been introduced, the OOVS method can be accomplished according to the block diagram shown in Fig. 3. Figure 3 is the expansion of Fig. 1, and in the figure, the visual servoing block has been explained concretely in Fig. 2. The robot sensor signal is divided into two parts by zero-order-holder: the first part is sampled with robot control period and the second one, which is denoted by "0" in the index, is sampled with visual servoing period starts.

When $\boldsymbol{Q}$ and $\boldsymbol{P}$ are obtained under the interpolation, they are employed as the input of simplified state equation. And the output of the equations, $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\xi}$ together with the feedback signal work

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

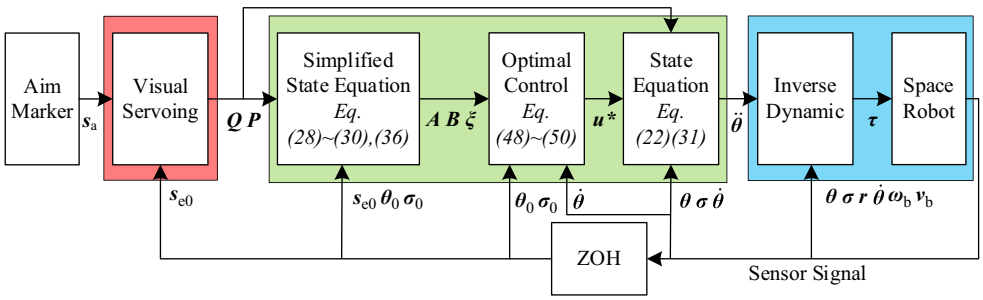
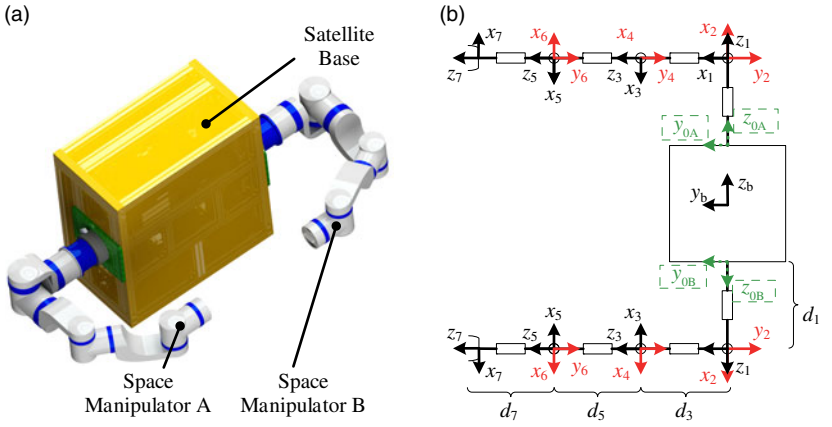


Fig. 3. The process of the off-line optimizing visual servoing algorithm.



The appearance of the robot system — The D-H coordinate of the robot system

Fig. 4. Simulation model of free-floating space robot system.

as the input of optimal control method. When the optimal control law is obtained, the state equation can be solved, and further, the FFSR can be controlled by the joint controller until another visual servoing period starts.

4.5. Applicability analysis

Since the OOVS method is applied to the visual servoing system, it is necessary to guarantee that the whole algorithm can be finished within a control period. By analyzing and checking all the parts in the method, it can be concluded that the most time-consuming processes are the calculation of $\partial \dot{\boldsymbol{J}}_G \dot{\boldsymbol{\theta}} / \partial \dot{\boldsymbol{\theta}}$ and the iterative process of optimal method. As a result, linearization is applied to the system for a higher computing efficiency. In this way, the two processes need only one calculation during one visual sampling period, which means that the two processes would only have influence on one or two control periods. Besides, when the calculated processes are programmed by C-code, it is verified that the processes can both be finished within 2 ms. In order to eliminate the influence on the further control caused by the two processes, the two parts are calculated in two separate control periods in this paper, and thus, the time requirements are satisfied. In other words, the method can be applied to the practical situations.

5. The Simulation and Verification

5.1. Simulation model

The dual-arm FFSR system for the simulation is shown in Fig. 4(a), and it is mainly composed of two 7-DOF space manipulators and a satellite base. The D–H coordinate system is shown in Fig. 4(b) and corresponding D–H parameters are shown in Table I. In the D–H coordinate system, the base coordinate, O_b, does not align with the manipulator installation coordinate, O_0, and the transform

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

Table I. The D–H parameters of the free-floating space robot system[a,b].

Joint	θ_i(°)	α_i(°)	a_i(m)	d_i(m)
1	90	0	0	0.253
2	90	90	0	0
3	180	90	0	0.4
4	180	90	0	0
5	180	90	0	0.4
6	180	90	0	0
7	0	90	0	0.171

[a] To align O_b with O_{0A}, a translation of (−0.039, −0.005, 0.564)m along z_b is needed
[b] To align O_b with O_{0B}, a translation of (−0.039, −0.005, −0.564)m along z_b followed by a rotation of 180° about y_b is needed.

Table II. The inertia parameter of the free-floating space robot system.

	Base	**Body1**	**Body2**	**Body3**	**Body4**	**Body5**	**Body6**	**Body7**
Mass (kg)	84.027	2.749	5.101	2.729	3.352	2.238	2.818	0.706
Mass Center (m)	0	0	0	0	0	0	0	0
	0	0.063	−0.058	0.048	−0.046	0.055	−42.911	0
	0	−0.069	−0.018	−0.142	−0.031	−0.149	−0.018	−0.026
I_{xx}(kg · m^2)	13.6386	0.0207	0.0496	0.0328	0.0163	0.0312	0.0125	0.0009
I_{xy}(kg · m^2)	−0.0448	0	0	0	0	0	0	0
I_{xz}(kg · m^2)	−0.0007	0	0	0	0	0	0	0
I_{yy}(kg · m^2)	10.6145	0.0162	0.0158	0.0291	0.0073	0.0287	0.0060	0.0009
I_{yz}(kg · m^2)	−0.0006	−0.0072	−0.0529	−0.0099	−0.0031	−0.0083	−0.0022	0.0001
I_{zz}(kg · m^2)	5.9666	5.9666	0.0304	0.0116	0.0136	0.0822	0.0105	0.0011

rule is also presented in Table I. In Table II, the inertia parameters are listed, which are indispensable for the general Jacobian and dynamics. The center mass in the table is measured with respect to the D–H coordinate system, and the inertia is measured with respect to the mass center. Since the dynamic coupling is greatly affected by the inertia parameters ratio of base to the manipulator, in order to make the results move clearly, the ratio used in this paper is lower than the real one.

5.2. The simulation of pre-capturing process

In order to clamp the target with dual-arm space robot, it is necessary to position the end-effector to a proper pose, and the positioning process is usually called pre-capturing process.

5.2.1. The simulation condition of visual servoing method. The simulation for minimizing the base disturbance during the pre-capturing process is set as follows:

In the simulation, the initial joint angle is [50, 10, 0, −50, 30, −40, 0, −50, 10, 0, −50, −30, −40, 0]°, and the initial base pose is [0, 0, 0] m and [0, 0, 0]°, respectively. Suppose the target stands still during the simulation and the aim displacement and RPY angle of the robot end are [−0.60, 0.15, 0.35, −0.60, 0.15, −0.3] m and [135, 0, 180, 0, 0, 0]°. In order to make the result more convincing, the aim end displacement is selected in an asymmetric way, and the only restriction on the aim end attitude is that the z-axes should be collinear. Besides, in the simulation, the visual servoing period is 200 ms, the control period of manipulator is 2 ms, and the coefficients are selected as $\rho = 0.05$, $\boldsymbol{R} = 1.5\mathbf{I}$, and $\boldsymbol{O} = 8000\mathbf{I}$.

5.2.2. The verification of visual servoing method. In order to verify the visual servoing method proposed in Section 3, the end trajectory of the FFSR is presented in Fig. 5. Fig. 5(a) and (b) are the end displacements of the manipulators. Roll-Pitch-Yaw (RPY) angle is shown in Fig. 5(b) and (e) just for a convenient illustration, while the end attitudes are planned by quaternion. Figure 5(c) and

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

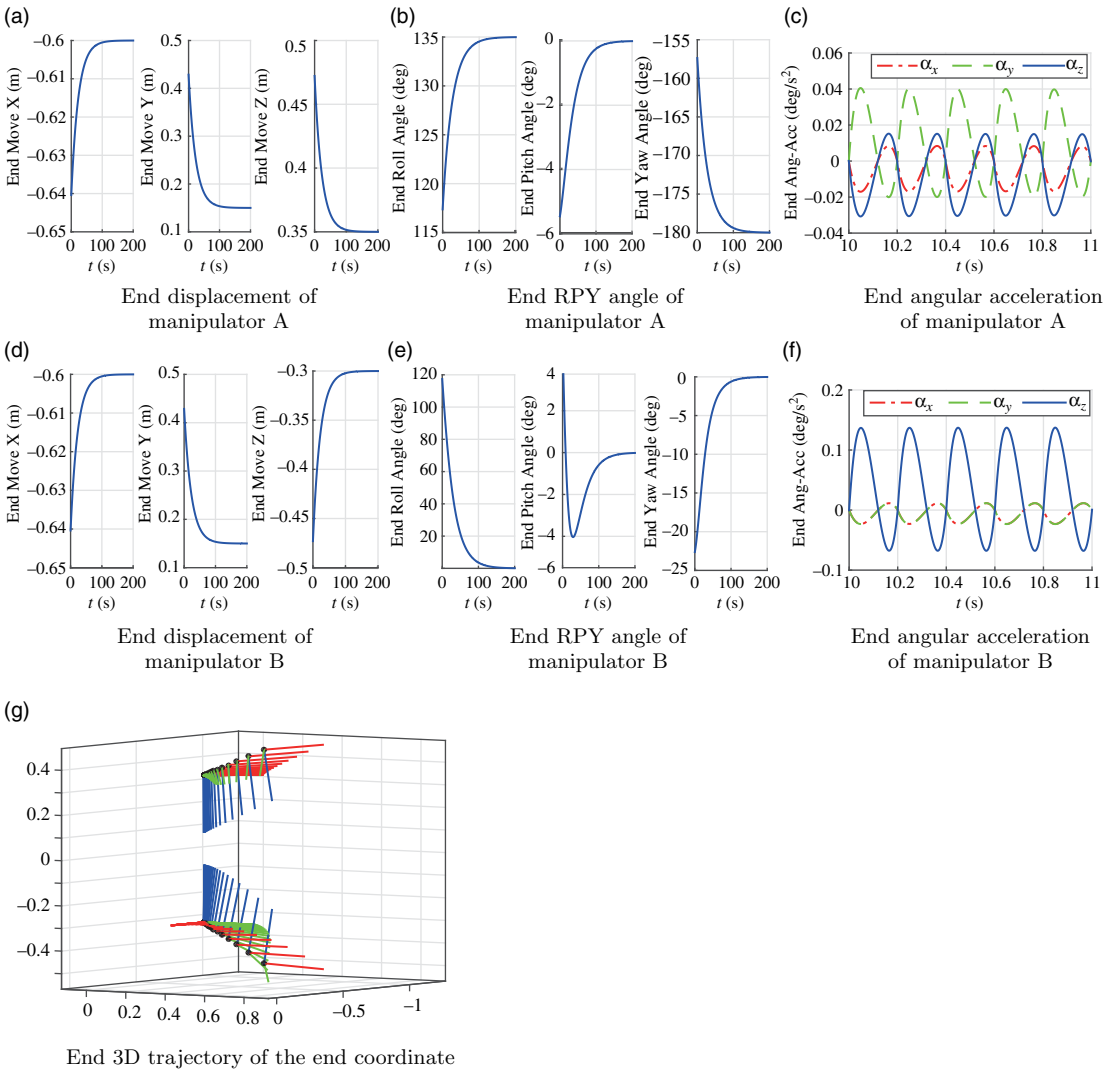


Fig. 5. The end trajectory of the space robot.

(f) are the end angular acceleration which is selected randomly with duration 1 s, and Fig. 5(g) is the trajectory of the space robot end coordinates.

In Fig. 5, the robot accomplishes the task within 200 s and the robot end pose varies from $[-0.640, 0.429, 0.472, -0.640, 0.429, -0.473]$ m, $[117.4, -5.5, -157.4, 117.4, 5.5 - 22.6]°$ to $[-0.60, 0.15, 0.35, -0.60, 0.15, -0.3]$ m, $[135, 0, -180, 0, 0, 0]°$, respectively. The angular accelerations reflect that the visual servoing method achieves a second-order continuity during the attitude planning. Besides, the 3D trajectory shows that the space robot ends point to each other when the simulation ends and thus can be applied to the further capturing. Besides, it can be concluded from the figure that though the simulation time is 200 s, the robot ends motions almost finish within 100 s and the rest robot motions are carried out nearly in the null space. Thus, the 200 s simulation can show a full view of the whole process.

5.2.3. The disturbance of satellite base. In this paper, the OOVS method is compared with the Quadratic Optimization method (Quad)[14] and the general Jacobian-based visual servoing method (Jac-based). In order to show the result in a concise way, the OOVS method, Quad method, and Jac-based method are noted by Case 1, Case 2, and Case 3, respectively.

During the pre-capturing process, the PRY angle and angular velocity of the satellite base are shown in Figs. 6 and 7. The figures show that when the FFSR is controlled with Jac-based method (Case 3), the maximum base disturbance is 27.3°, while the OOVS (Case 1) and Quad (Case 2)

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

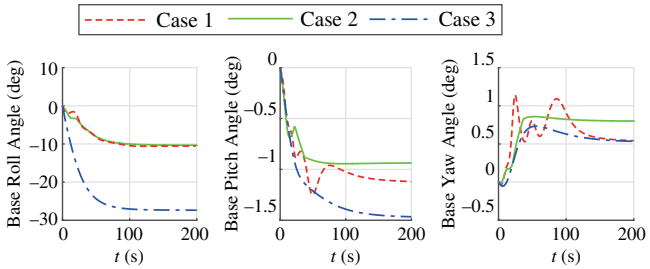


Fig. 6. The RPY angle of the satellite base.

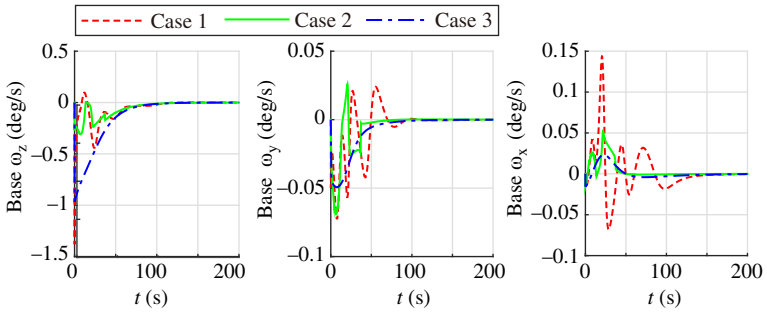


Fig. 7. The angular velocity of the satellite base.

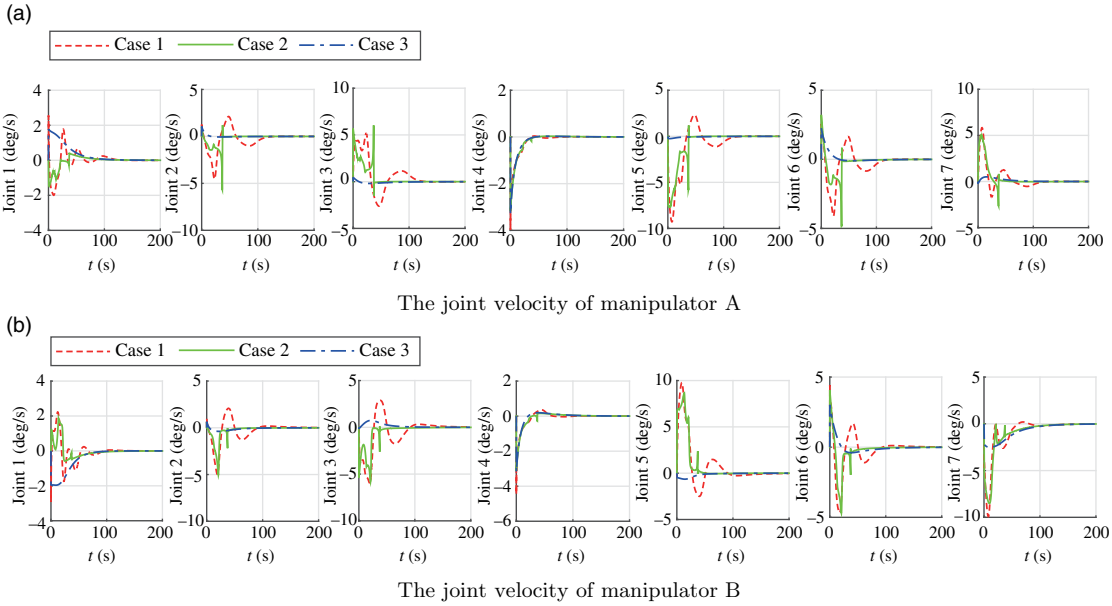


Fig. 8. The joint velocity of the space manipulator.

methods are about 10.7° and 10.3°, respectively, and the disturbance is decreased by 62%. And in the minor disturbance directions, the base disturbances are either reduced to a relative low value or located close to each other. In Fig. 7, it can be obtained that in the major rotation direction, the $\boldsymbol{\omega}_z$s of OOVS (Case 1) and Quad (Case 2) are smaller than the Jac-based method (Case 3) and thus can help with reducing the disturbance. In other directions, though the absolute value of the angular velocity is larger for the optimization methods, the reciprocating rotation leads to an acceptable result.

The joint velocities in the process are shown in Fig. 8. There are spikes for Quad method curves (Case 2), while the curves of the OOVS methods (Case 1) are much smoother. Since the spikes may

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

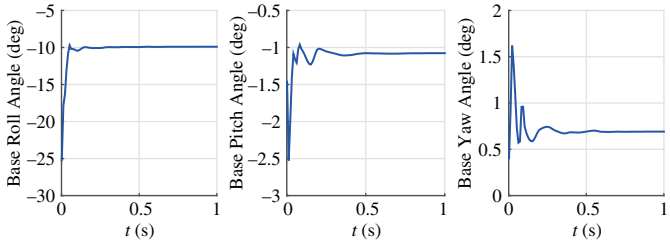


Fig. 9. The RPY angle of the satellite base when T varies from 0.002 to 1 s.

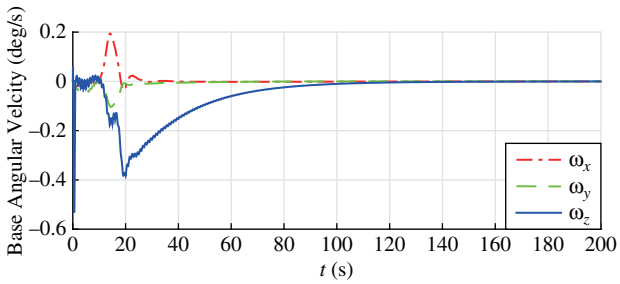


Fig. 10. The angular velocity of the satellite base when the visual sampling period is 1 s.

result in impact to the manipulator, OOVS method (Case 1) is suggested to be much better for robot controlling.

Then the phenomenon is analyzed qualitatively. Though Quad method and OOVS method can both optimize the base disturbance, the former tries to achieve the optimization instantaneously, while the latter considers the process during certain time. As a result, a prediction of the manipulator motion can be made by OOVS method and finally achieve a better optimization.

5.2.4. Further verifying simulations. To verify the optimized method further, more simulations are carried out to compare OOVS and Jac-based methods in this part. In these simulations, the aim positions of the robot ends are selected randomly in the workspace and the attitudes are selected with collinear z-axes, which is resulted from the restriction of the robot clamping task. The simulation results are listed in Table III, and the data show that in most cases, the OOVS method can greatly reduce the base disturbance. Besides, the base disturbances in row 5 and row 6 are located close to zero, and to make the results more clear, the PRY angles are transformed to angle-axi value here, which is 0.86° (OOVS) and 1.06° (Jac-based) for row 5, and 1.46° (OOVS) and 1.97° (Jac-based) for row 6. The results show that even when it is close to zero, the OOVS performs still better compared with Jac-based method. To sum up, the OOVS method has the advantage of reducing base disturbance and thus can help with the space tasks.

5.2.5. The influence of the visual sampling period to the optimization. The visual sampling period effects the optimization result in two aspects. If T is tiny, the OOVS method would have little influence on the final consequence. And if T is too large, the error introduced by the estimation in Eqs. (20)–(22) may be an obstruction of a fine optimization result.

The final base disturbance of the OOVS method with T varies from 0.002 to 1 s is shown in Fig. 9. In Fig. 9, the disturbance gradually converges to a certain value as the sampling time increases. When $T = 0.002$ s, which equals to the robot control period, the base rotates nearly the same as the one shown in Fig. 6 with Jac-based method, and if $T > 0.05$ s, the final result would locate around 10°.

The base angular velocity of the base with $T = 1$ s is shown in Fig. 10. Though the final result of Fig. 10 is acceptable, the control process is corrupted by the vibration. Thus, it is not a good choice for the robot.

It is verified by more simulations that when the visual sampling period is located in 0.05–0.25 s, the optimization can meet the needs of practical application. The upper bound of the interval is

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

Table III. Comparisons between the OOVS method and the Jac-based method.

Index	**End Aim Displacement (m)**	**End Aim PRY (°)**	**Base RPY with OOVS method (°)**	**Base RPY with Jac-based method (°)**
1	[−0.60, 0.10, 0.30, −0.60, 0.10, −0.35]	[0, 0, 180, 60, 0, 0]	[−16.06, −0.09, 0.43]	[−26.67, −0.33, 0.29]
2	[−0.60, 0.15, 0.25, −0.60, 0.15, −0.40]	[45, 0, 180, 40, 0, 0]	[−12.71, 0.88, −0.16]	[−23.72, 0.93, 0.05]
3	[−0.50, 0.20, 0.20, −0.50, 0.20, −0.45]	[90, 0, 180, 20, 0, 0]	[−6.28, 2.98, 0.24]	[−14.95, 2.67, 0.44]
4	[−0.50, 0.25, 0.40, −0.50, 0.25, −0.25]	[120, 180, 0, −20, 0, 0]	[−3.72, −1.91, −0.82]	[−10.78, −2.21, −1.08]
5	[−0.40, 0.30, 0.35, −0.40, 0.30, −0.30]	[60, 180, 0, −40, 0, 0]	[−0.60, −0.21, −0.58]	[−0.76, −0.09, −0.73]
6	[−0.40, 0.35, 0.30, −0.40, 0.35, −0.35]	[15, 180, 0, −60, 0, 0]	[0.72, 0.03, 1.27]	[1.74, 0.70, 0.61]
7	[−0.30, 0.40, 0.25, −0.30, 0.40, −0.40]	[−45, 180, 0, −80, 0, 0]	[6.86, −0.50, 2.93]	[12.96, 1.42, 2.14]
8	[−0.30, 0.45, 0.20, −0.30, 0.45, −0.45]	[−90, 180, 0, −100, 0, 0]	[9.26, −0.60, 3.79]	[15.89, 2.24, 3.15]

ascertained by the actual condition, and the lower bound suggests that the method proposed in the paper is suitable for the further visual servoing process with shortened sampling period.

6. Conclusion

In this paper, we make use of the difference between visual servoing sampling periods and robot control period, and propose OOVS method to deal with satellite base disturbance especially when the number of redundant DOFs of space robot is less than a critical value during the visual servoing process. First, the visual servoing process is studied and the interpolation method in the process guarantees a second-order continuity of quaternion. Then the robot model is built based on the characteristic of visual servoing process and the robot inverse kinematic. The OOVS methods are validated by the simulation and the results show that in most cases, the base disturbance can be greatly decreased compared with Jacobian-based methods. Compared with the Quadratic Optimization method, the OOVS method generates a smoother joint trajectory and can fit better for robot control. In the future, we will apply the OOVS method to the satellite capturing experiment in the ground-based verification system for further validation.

Acknowledgments

This project was supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 91848202).

References

1. W. Xu, D. Meng, H. Liu, X. Wang and B. Liang, "Singularity-free trajectory planning of free-floating multiarm space robots for keeping the base inertially stabilized," *IEEE Trans. Syst. Man Cybernet.: Syst*. (99), 1–14 (2017).
2. W. Xu, B. Liang and Y. Xu, "Survey of modeling, planning, and ground verification of space robotic systems," *Acta Astron.* **68**(11–12), 1629–1649 (2011).
3. W. Xu, J. Peng, H. Liu, B. Liang and Z. Mu, "Hybrid modeling and analysis method for dynamic coupling of space robots," *IEEE Trans. Aerosp. Electron. Syst.* **52**(1), 85–98 (2016).
4. E. Papadopoulos and A. Abu-Abed, "On the design of zero reaction manipulators," *J. Mech. Design*. **118**(3), 372–376 (1996).
5. T. C. Nguyen-Huynh and I. Sharf, "Adaptive Reactionless Motion for Space Manipulator When Capturing an Unknown Tumbling Target," *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China (2011) pp. 4202–4207.
6. S. Dubowsky and M. A. Torres, "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, California, USA (1991) pp. 2522–2528.
7. M. A. Torres and S. Dubowsky, "Minimizing spacecraft attitude disturbances in space manipulator systems," *J. Guid. Control Dyn*. **15**(4), 1010–1017 (1992).
8. F. James, S. V. Shah, A. K. Singh, K. M. Krishna and A. K. Misra, "Reactionless maneuvering of a space robot in precapture phase," *J. Guid. Control Dyn*. **39**(10), 2419–2425 (2016).
9. A. Gouo, D. N. Nenchev, K. Yoshida and M. Uchiyama, "Motion control of dual-arm long-reach manipulators," *Adv. Robot*. **13**(6), 617–631 (1998).
10. S. V. Shah, I. Sharf and A. Misra, "Reactionless Path Planning Strategies for Capture of Tumbling Objects in Space Using a Dual-arm Robotic System," *AIAA Guidance, Navigation, and Control (GNC) Conference*, Boston, Massachusetts, USA (2013) p. 4521.
11. T. Hishinuma and D. N. Nenchev, "Singularity-consistent Vibration Suppression Control with a Redundant Manipulator Mounted on a Flexible Base," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 3237–3242.
12. K. Yoshida, K. Hashizume and S. Abiko, "Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm," *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea (2001) pp. 441–446.
13. S. Cocuzza, I. Pretto and S. Debei, "Reaction torque control of redundant space robotic systems for orbital maintenance and simulated microgravity tests," *Acta Astron*. **67**(3–4), 285–295 (2010).
14. A. H. Abdul Hafez, P. Mithun, V. V. Anurag, S. V. Shah and K. M. Krishna, "Reactionless visual servoing of a multi-arm space robot combined with other manipulation tasks," *Robot. Auton. Syst*. **91**, 1–10 (2017).
15. S. Cocuzza, I. Pretto and S. Debei, "Least-squares-based reaction control of space manipulators," *J. Guid. Control Dyn*. **35**(3), 976–986 (2012).
16. A. C. M. Allen, C. Langley, R. Mukherji, A. B. Taylor, M. Umasuthan and T. D. Barfoot, "Rendezvous Lidar Sensor System for Terminal Rendezvous, Capture, and Berthing to the International Space Station," *Proceedings of SPIE 6958, Sensors and Systems for Space Applications II*, Orlando, Florida, USA (2008).
17. P. Motaghedi, "On-orbit Performance of the Orbital Express Capture System," *Proceedings of SPIE 6958, Sensors and Systems for Space Applications II*, Orlando, Florida, USA (2008).

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press

18. K. Yoshida, K. Hashizume, D. Nenchev, N. Inaba and M. Oda, "Control of a Space Manipulator for Autonomous Target Capture-ETS-VII Flight Experiments and Analysis," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, Colorado, USA (2000) p. 4376.
19. H. Dapeng, F. Xiao and W. Qing, "Rotation Interpolation Based on the Geometric Structure of Unit Quaternions," *IEEE International Conference on Industrial Technology*, Chengdu, China (2008) pp. 1–6.
20. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, "Robotics: Modelling, Planning and Control," *Adv. Textbooks Control Sig. Process*. **4**(12), 76–82 (2009).
21. F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag*. **13**(4), 82–90 (2006).
22. P. Corke, "Robotics, Vision and Control: Fundamental Algorithms," **In:** *MATLAB® Second, Completely Revised*, vol. 118 (Springer, Berlin Heidelberg, 2017).
23. G. Chesi and K. Hashimoto, *Visual Servoing via Advanced Numerical Methods*, vol. 401 (Springer, Berlin Heidelberg, 2010).
24. J. Solà, "Quaternion kinematics for the error-state Kalman filter," *CoRR*, abs/1711.02508 (2017).
25. K. Myoung-Jun, K. Myung-Soo and S. Sung Yong, "A C^2-continuous B-spline Quaternion Curve Interpolating a Given Sequence of Solid Orientations," *Proceedings of Computer Animation*, Geneva, Switzerland (1995) pp. 72–81.
26. M.-J. Kim, M.-S. Kim and S. Y. Shin, "A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives," *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, California, USA (1995) pp. 369–376.
27. S. Gupta and J. Y. S. Luh, "Closed-loop Control of Manipulators with Redundant Joints Using the Hamilton-Jacobi-Bellman Equation," *IEEE International Conference on Proceedings of Robotics and Automation*, Sacramento, California, USA (1991) pp. 472–477.
28. C. Natale, *Interaction Control of Robot Manipulators: Six Degrees-of-Freedom Tasks*, vol. 3 (Springer Science and Business Media, Berlin Heidelberg, 2003).
29. P. Mithun, V. V. Anurag, M. Bhardwaj and S. V. Shah, "Real-time Dynamic Singularity Avoidance While Visual Servoing of a Dual-arm Space Robot," *Proceedings of the Conference on Advances in Robotics*, Goa, India (2015), p. 31.
30. D. E. Kirk, *Optimal Control Theory: An Introduction* (Dover Publications, Mineola, New York, 2012).

https://doi.org/10.1017/S0263574719000924 Published online by Cambridge University Press