

Research Article

GA Based Adaptive Singularity-Robust Path Planning of Space Robot for On-Orbit Detection

Jianwei Wu ¹, Deer Bin ¹, Xiaobing Feng ², Zhongpu Wen ¹ and Yin Zhang¹

¹Ultra-Precision Optoelectronic Instrumentation Engineering Centre, Harbin Institute of Technology, Harbin 150001, China

²Manufacturing Metrology Team, Faculty of Engineering, University of Nottingham, Nottingham NG8 1BB, UK

Correspondence should be addressed to Jianwei Wu; wujianwei@hit.edu.cn

Received 12 February 2018; Accepted 19 April 2018; Published 28 May 2018

Academic Editor: Zhile Yang

Copyright © 2018 Jianwei Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a new on-orbit detection platform, the space robot could ensure stable and reliable operation of spacecraft in complex space environments. The tracking accuracy of the space manipulator end-effector is crucial to the detection precision. In this paper, the Cartesian path planning method of velocity level inverse kinematics based on generalized Jacobian matrix (GJM) is proposed. The GJM will come across singularity issue in path planning, which leads to the infinite or incalculable joint velocity. To solve this issue, firstly, the singular value decomposition (SVD) is used for exposition of the singularity avoidance principle of the damped least squares (DLS) method. After that, the DLS method is improved by introducing an adaptive damping factor which changes with the singularity. Finally, in order to improve the tracking accuracy of the singularity-robust algorithm, the objective function is established, and two adaptive parameters are optimized by genetic algorithm (GA). The simulation of a 6-DOF free-floating space robot is carried out, and the results show that, compared with DLS method, the proposed method could improve the tracking accuracy of space manipulator end-effector.

1. Introduction

With the development of space technology, the demand for longer life and higher reliability of future spacecraft is increasing. On-orbit detection has currently become a crucial technology which can guarantee the stability and reliability of spacecraft in complex environment of space [1]. At present, typical on-orbit detection platforms include XSS [2] and MiniAERCam [3]. Due to good mobility, space robot, as a new on-orbit platform, will play an important role in on-orbit detection [4]. Through the motion of the space manipulator, the sensors like line structure light sensor carried by the end-effector can accurately track the specified detection trajectory and achieve the detailed detection of the spacecraft surface. Therefore, the Cartesian trajectory tracking accuracy of the space robot is very important for on-orbit detection.

Compared with a ground fixed-base robot, the movement of space manipulator would cause reaction and change the position and attitude of its carrier. Usually, the kinematic equation of position level cannot be used to plan the space

robot joint motion. Therefore, when the velocity level inverse kinematics is applied, the GJM [5] will come across singularity issue. If the solution based on Jacobi matrix inverse is still adopted, it will lead to the infinite or incalculable joint velocity, which makes the path planning and control algorithm invalid. Therefore, the singular avoidance and robustness of Jacobi matrix is necessary.

The problem of kinematic singularity is widely studied for base-fixed ground robot. Many researches have been proposed to realize singularity-robust algorithm in the proximity of singularities. The kinematics singularity of robot arm such as PUMA type robot was analyzed by Angeles, and it can be classified into three categories: shoulder singularity, elbow singularity, and wrist singularity [6]. To handle singularity problems, the DLS method is often utilized [7–10]. A framework for handling robotic singularities was proposed by Carmichael et al. The damping is applied asymmetrically depending on whether the robot is heading towards or away from singular configurations [11]. Cui et al. proposed a singularity avoidance algorithm, and singularity avoidance is achieved by replacing the common reciprocal with the

improved Gaussian distribution damped reciprocal [12]. Xu et al. proposed a method to isolate the singularity condition and decompose the workspace of a class of manipulators without spherical wrists, either redundant or nonredundant [13]. Lai et al. proposed an algorithm to form a new path for the pivot that can avoid the discretization near the singularity points [14].

The singularity analysis and avoidance of free-floating space robots are much more complicated. The kinematics and dynamics of a free-floating space robot are coupled, and the GJM contains not only the kinematic parameters, but also the dynamic parameters. Papadopoulos and Dubowsky first proposed the concept of dynamic singularity in the literature [15], the free space was divided into space robot path independent work space (PIW) and path dependent work space (PDW), avoiding dynamic singularity by transposed GJM. They also proposed an avoiding dynamic singularity by finding a trouble-free space in PIW [16]. Lampariello et al. proposed a parameterization method in joint space and planned the point to point Cartesian trajectory only through forward kinematics, so it is not affected by the dynamic singularity [17, 18]. The workspace of 3R robot was analyzed by Xu et al., and some suggestions were put forward for the design of space robot to reduce the influence of dynamic singularity [19]. Jin et al. proposed a reactionless control, and the dynamic singularity avoidance is achieved by singular value filtering method [20].

In this paper, the kinematic model of a space robot with coupling dynamic parameters is first established, and the dynamic singularity characteristics are illustrated based on SVD. A damping adaptive singular avoidance method with varying value of singularity is proposed. An optimization objective function is established with the target of tracking accuracy of space manipulator end-effector. The selection of two parameters for adaptive damping factor is achieved by GA. Singularity-robust path planning of space robot is realized by using this method, and the difficult of GJM singular control is overcome. It achieves better tracking accuracy compared with the DLS method.

2. Cartesian Path Planning

Free-floating space robot is a typical nonholonomic system. The position and attitude of end-effector is not only related to the current joint angle, but also related to the previous motion of the joint. It cannot obtain the joint angle through the analytical position level inverse kinematics as the ground fixed-base robot. Therefore, a numerical method in velocity level inverse kinematics is usually employed. An n -DOF free-floating space manipulator vector model is shown in Figure 1.

Figure 1 is reproduced from Wu et al. [4]. (2016) (under the Creative Commons Attribution License/public domain). The velocity of position and attitude of space manipulator effector can be expressed as

$$\begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_g \dot{\boldsymbol{\theta}}, \quad (1)$$

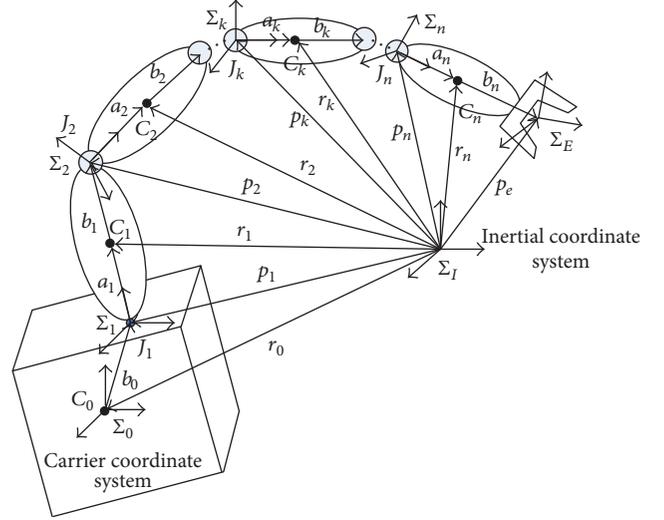


FIGURE 1: The n -DOF free-floating space manipulator vector model.

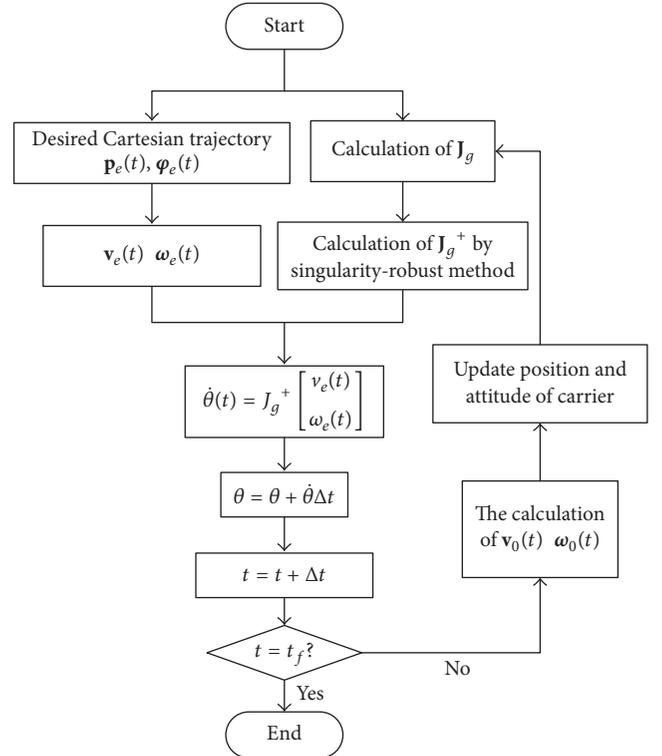


FIGURE 2: Calculation flow chart of the Cartesian path planning for free-floating space robot.

where \mathbf{J}_g is the GJM of the space robot, which is related to the attitude of the space robot carrier, the joint angle, and the mass and inertia of the whole system.

As shown in Figure 2, the position and attitude velocity of the end-effector are determined by desired Cartesian position trajectory $\mathbf{p}_e(t)$ and attitude trajectory $\boldsymbol{\varphi}_e(t)$, which are expressed as

$$\mathbf{v}_e(t) = \dot{\mathbf{p}}_e(t)$$

$$\boldsymbol{\omega}_e(t) = \vec{\mathbf{r}} \dot{\boldsymbol{\phi}}_e(t), \quad (2)$$

where $\vec{\mathbf{r}}$ is the axis unit vector of rotation from the initial attitude to the target attitude.

The GJM \mathbf{J}_g is calculated according to the initial conditions, and the robust inverse \mathbf{J}_g^+ of the GJM is obtained by using the corresponding singularity-robust algorithm. Based on the velocity level inverse kinematics, the joint angular velocity $\dot{\boldsymbol{\theta}}(t)$ is planned:

$$\dot{\boldsymbol{\theta}}(t) = \mathbf{J}_g^+ \begin{bmatrix} \mathbf{v}_e(t) \\ \boldsymbol{\omega}_e(t) \end{bmatrix} \quad (3)$$

To calculate the GJM at next calculation period, the position and attitude of carrier need to be updated before the next period started.

3. Adaptive Singularity-Robust Algorithm

3.1. Damped Least Squares (DLS) Method. DLS method was first introduced to robot kinematics by Wampler in 1986 [8]. The idea of DLS is to make a compromise between the tracking accuracy and the joint velocity and minimize the function \mathbf{B} which is expressed in

$$\mathbf{B} = \|\mathbf{J}\dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}\|^2 + \lambda^2 \|\dot{\boldsymbol{\theta}}\|^2, \quad (4)$$

where $\dot{\mathbf{x}}$ is the terminal velocity vector $[\mathbf{v}_e(t) \ \boldsymbol{\omega}_e(t)]^T$. λ^2 is the damping factor, which can be explained as the relative weight factor of $\|\mathbf{J}\dot{\boldsymbol{\theta}} - \dot{\mathbf{x}}\|^2$ and $\|\dot{\boldsymbol{\theta}}\|^2$. By solving the normal equation,

$$\begin{bmatrix} \mathbf{J} \\ \lambda \mathbf{I} \end{bmatrix} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\mathbf{x}} \\ 0 \end{bmatrix} \quad (5)$$

Then we can get the unique solution to make (12) minimized:

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \lambda^2 \mathbf{I})^{-1} \dot{\mathbf{x}} \quad (6)$$

or

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_g^+ \dot{\mathbf{x}} \quad (7)$$

3.2. GA Based Singularity-Robust Algorithm. The principle of DLS method is further clarified through the SVD. The singular value of \mathbf{J}_g can be expressed as

$$\mathbf{J}_g = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T, \quad (8)$$

where σ_i is the singular value of \mathbf{J}_g , \mathbf{u}_i and \mathbf{v}_i are the singular vectors of \mathbf{J}_g , and

$$\mathbf{J}_g^T \mathbf{J}_g + \lambda^2 \mathbf{I} = \sum_{i=1}^r (\sigma_i^2 + \lambda^2) \mathbf{u}_i \mathbf{v}_i^T \quad (9)$$

Putting (9) into (6) and (7),

$$\begin{aligned} \dot{\boldsymbol{\theta}}^{(\lambda)} &= (\mathbf{J}_g^T \mathbf{J}_g + \lambda^2 \mathbf{I})^{-1} \mathbf{J}_g^T \begin{bmatrix} \mathbf{v}_e(t) \\ \boldsymbol{\omega}_e(t) \end{bmatrix} \\ &= \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \mathbf{u}_i^T \begin{bmatrix} \mathbf{v}_e(t) \\ \boldsymbol{\omega}_e(t) \end{bmatrix} \end{aligned} \quad (10)$$

The velocity vector of the manipulator end-effector $[\mathbf{v}_e(t) \ \boldsymbol{\omega}_e(t)]^T$ can be represented as a the output vector \mathbf{u}_i linear combination:

$$\begin{bmatrix} \mathbf{v}_e(t) \\ \boldsymbol{\omega}_e(t) \end{bmatrix} = \sum_{i=1}^r \dot{\mathbf{x}}_i \mathbf{u}_i \quad (11)$$

\mathbf{u}_i and \mathbf{u}_j are orthogonal to each other, and i is not equal to j ; $\dot{\mathbf{x}}_i$ can be expressed as

$$\dot{\mathbf{x}}_i = \mathbf{u}_i^T \begin{bmatrix} \mathbf{v}_e(t) \\ \boldsymbol{\omega}_e(t) \end{bmatrix} \quad (12)$$

Putting (12) into (10),

$$\dot{\boldsymbol{\theta}}^{(\lambda)} = \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \dot{\mathbf{x}}_i \quad (13)$$

Let $\dot{\boldsymbol{\theta}}_i^{(\lambda)} = (\sigma_i / (\sigma_i^2 + \lambda^2)) \mathbf{v}_i \dot{\mathbf{x}}_i$, then (13) can be rewritten as $\dot{\boldsymbol{\theta}}^{(\lambda)} = \sum_{i=1}^r \dot{\boldsymbol{\theta}}_i^{(\lambda)}$, and obviously $\dot{\boldsymbol{\theta}}_i^{(\lambda)}$ are orthogonal to each other. So

$$\|\dot{\boldsymbol{\theta}}^{(\lambda)}\|^2 = \sum_{i=1}^r \|\dot{\boldsymbol{\theta}}_i^{(\lambda)}\|^2 = \sum_{i=1}^r \left(\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \right)^2 \dot{\mathbf{x}}_i^2 \quad (14)$$

Tracking errors can be expressed as

$$\left\| \begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} - \mathbf{J}_g \dot{\boldsymbol{\theta}}^{(\lambda)} \right\|^2 = \sum_{i=1}^r \dot{\mathbf{x}}_i^2 \left(\frac{\lambda^2}{\sigma_i^2 + \lambda^2} \right)^2 + \sum_{i=r+1}^m \dot{\mathbf{x}}_i^2, \quad (15)$$

where r and m are the rank of the GJM when the rank is full and not full, and the error consists of two parts. $\sum_{i=r+1}^m \dot{\mathbf{x}}_i^2$ is caused by the singular value being zero and the output speed being zero; that is to say, no matter how fast the joint is, there is no speed at the end-effector, which leads to tracking errors. This part of the errors exists objectively, which theoretically cannot be overcome. However, the error $\sum_{i=1}^r \dot{\mathbf{x}}_i^2 (\lambda^2 / (\sigma_i^2 + \lambda^2))^2$ can be reduced by adjusting the damping λ .

Considering the velocity continuity of end-effector and minimizing the tracking error, an adaptive damping factor λ is introduced:

$$\lambda_i = \begin{cases} \frac{\lambda_{\max} (1 + \cos(\pi \sigma_i / \varepsilon))}{2} & \sigma_i < \varepsilon \\ 0 & \sigma_i \geq \varepsilon, \end{cases} \quad (16)$$

where λ_{\max} is the maximum damping factor; ε is a parameter for evaluating the singularity of GJM. Different from the DLS, it is not to introduce damping for every σ_i but to introduce

it at a certain degree. The selection principle of λ_{\max} and ε is to make the joint speed smooth and the tracking error as small as possible. If the value of λ_{\max} is too large, although the singularity avoidance is effective, a large tracking error is introduced. On the contrary, if the singularity is not effectively solved, the motion of manipulator is not good. It is generally necessary to adjust these two parameters to get the appropriate value. Usually λ_{\max} and ε are on the same order of magnitude, and the intelligent optimization algorithm can be used to select these two parameters. In this paper, GA is adopted, and an objective function B is established as shown in formula (17).

$$B = w_1 \Delta p + w_2 \Delta p_\phi + P_l, \quad (17)$$

where Δp and Δp_ϕ are the position errors, respectively, which can be expressed by (18). In the process of model, the objective function of the dimensionless unified approach and the target weight coefficients are explored. Through the introduction of weight coefficients w_1 and w_2 , we can transform the problem of multiobjective optimization to that of single objective optimization. According to the different attentiveness, we can appropriately choose w_1 and w_2 and make a compromise between position and attitude errors.

$$\Delta p = \sqrt{\Delta p_x^2 + \Delta p_y^2 + \Delta p_z^2} \quad (18)$$

$$\Delta p_\phi = \sqrt{(\Delta p_x \cdot \Delta \phi_x)^2 + (\Delta p_y \cdot \Delta \phi_y)^2 + (\Delta p_z \cdot \Delta \phi_z)^2}$$

P_l is the limitation factor of joint velocity range, which is given by the following expression:

$$P_l = \begin{cases} +\infty & \theta_i > \theta_{+i.\text{limit}} \\ +\infty & \theta_i < \theta_{-i.\text{limit}} \\ 0 & \theta_{-i.\text{limit}} \leq \theta_i \leq \theta_{+i.\text{limit}} \end{cases} \quad (19)$$

In the calculation, if the max joint velocity exceeds the limit range $(\theta_{-i.\text{limit}}, \theta_{+i.\text{limit}})$, P_l is equal to positive infinity. Therefore, the smoothness of motion can be guaranteed automatically by introducing the limitation factor P_l in optimization algorithm.

The calculation steps of GA are as follows.

Step 1 (the generation of the initial population). The initial population of the parameters $(\lambda_{\max}, \varepsilon)$, which contains M individuals, is randomly generated with the search range. Therefore, according to the parameterized equation, M singularity avoidance trajectories are obtained.

Step 2 (evaluation of individual fitness). The opportunity of each individual is determined by GA according to the probability that it is proportional to the fitness. To calculate the probability correctly, the fitness of all individuals must be nonnegative. So, the fitness function is the objective function B in this paper.

Step 3 (selection). Selection operations employ the roulette selection method. The probability for each individual is equal

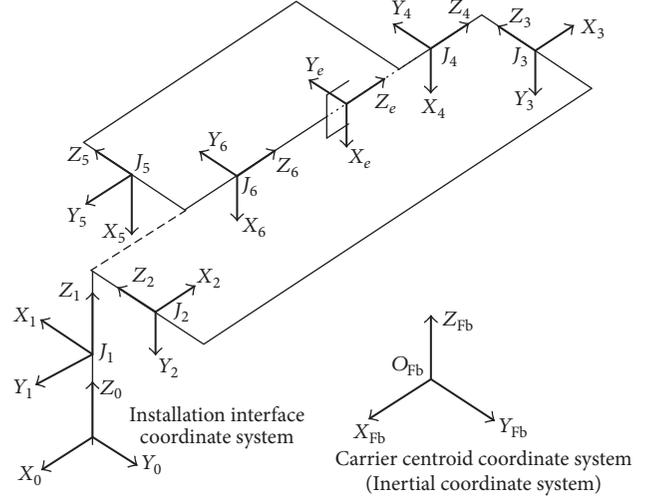


FIGURE 3: Linkage coordinate frames for a 6-DOF manipulator.

to a proportion of its fitness and individual fitness sum in the whole population. If there are M_c individuals of the population and the fitness of individual i is f_i , then the probability of individual i can be expressed as

$$P_i = \frac{S_i}{\sum_{k=1}^{M_c} S_k} \quad (20)$$

When the selection is given, a random number of 0 to 1 is generated to determine which individuals will cross at next step. The individual which has large selection probability will be selected many times, and its genetic gene will be expanded in the population. On the contrary, it will be eliminated.

Step 4 (cross). Select two individuals randomly after selection operation, and a crosspoint of two individuals is generated. By exchanging part of the gene code at the crosspoint, two new individuals are formed.

Step 5 (mutation). According to the probability of gene mutation, the small probability change of binary genetic code of some individuals in the population is realized.

Step 6 (new population). The new population is generated by inserting new individuals.

4. Simulation

The simulation object is a typical free-floating 6-DOF space robot, whose reference coordinate system is shown in Figure 3. The linkage parameters a_i , b_i and the moment of inertia are listed in Table 1. The definition of a_i , b_i can be referred to in literature [4].

Figure 3 is reproduced from Wu et al. [4] (2016) (under the Creative Commons Attribution License/public domain).

In the simulation, a point to point linear trajectory is planned, the initial and expected position and attitude of the end-effector are 1.37, -0.08, 0.64, -3.14, 0, and -2.97 and 1.50, 0, -0.2, -3.16, -0.26, and -3.14. The initial value of the

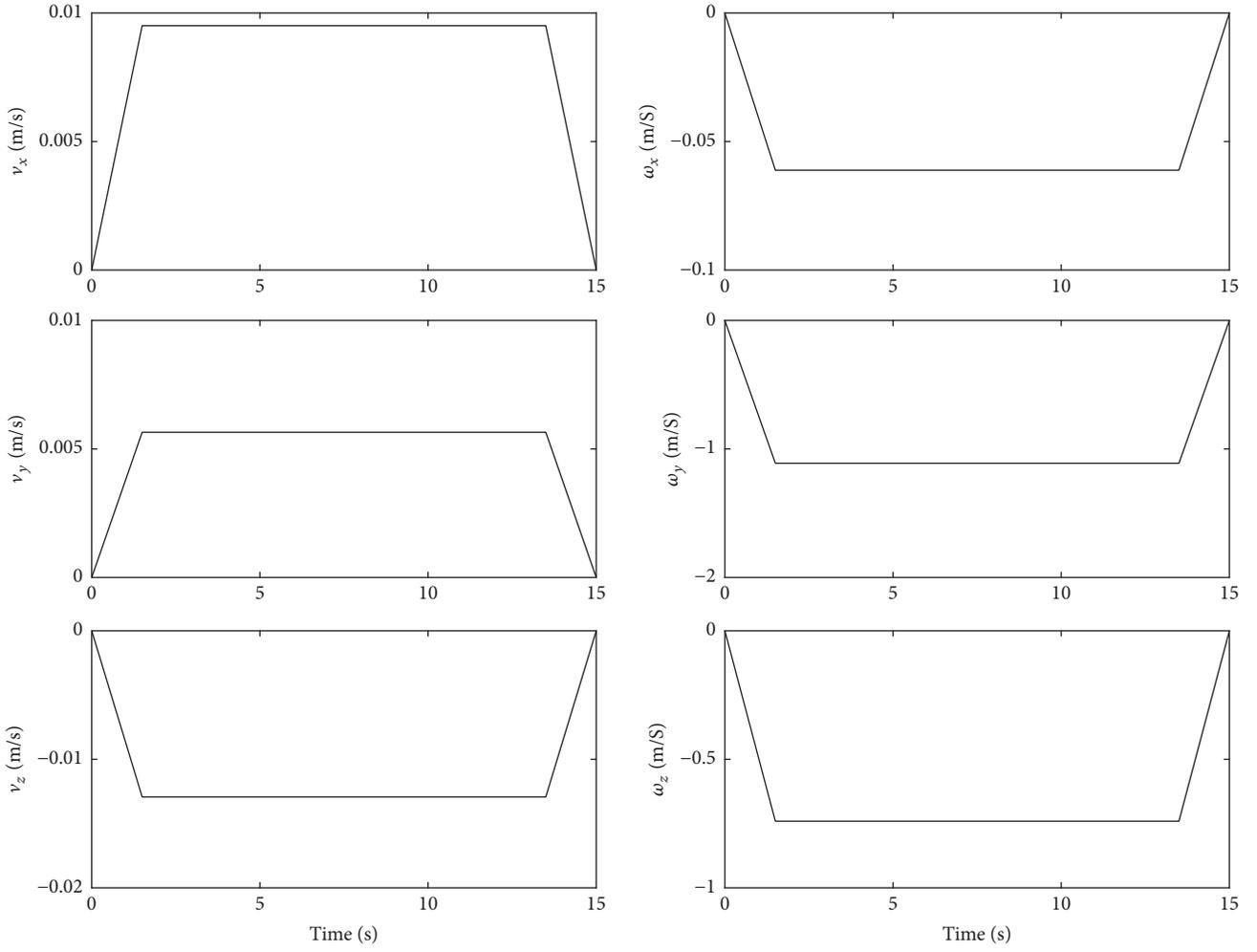


FIGURE 4: Desired end-effector velocity of space manipulator end-effector.

TABLE 1: Linkage parameters of the space robot.

Parameters	Carrier	Linkage 1	Linkage 2	Linkage 3	Linkage 4	Linkage 5	Linkage 6
Mass (kg)	450.0	1.5	9.6	1.5	9.0	1.5	10.5
a_i (mm)	0	0	-492.5	0	292.0	0	-136.0
	0	0	54.0	-121.0	-150.0	121.0	0
	0	124.0	0	0	0	0	0
b_i (mm)	500.0	0	-492.5	124.0	349.0	-124.0	-164.0
	0	121.0	-54.0	0	29.0	0	0
	751.0	0	0	0	0	0	0
Moment of inertia (kg/m ²)							
I_{xx}	200.00	3.31×10^{-3}	3.10×10^{-2}	3.31×10^{-3}	0.73	3.31×10^{-3}	0.10
I_{yy}	200.00	3.31×10^{-3}	1.50	3.31×10^{-3}	0.60	3.31×10^{-3}	0.10
I_{zz}	200.00	3.31×10^{-3}	1.48	3.31×10^{-3}	0.63	3.31×10^{-3}	0.09
I_{xy}	0	0	0	0	0	0	0
I_{xz}	0	0	0	0	0	0	0
I_{yz}	0	0	0	0	-0.01	0	0

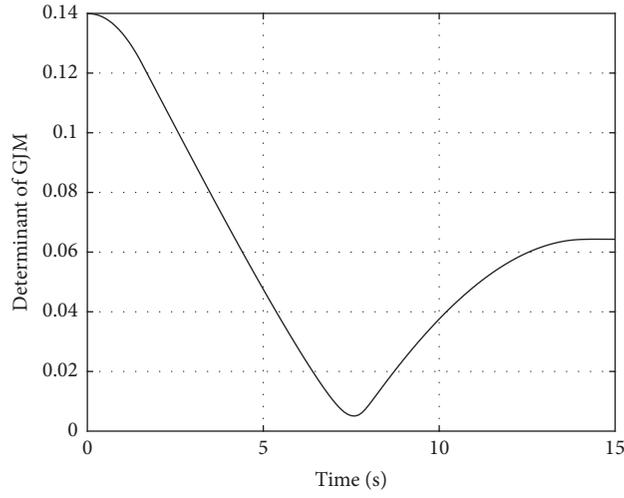


FIGURE 5: Determinant of GJM.

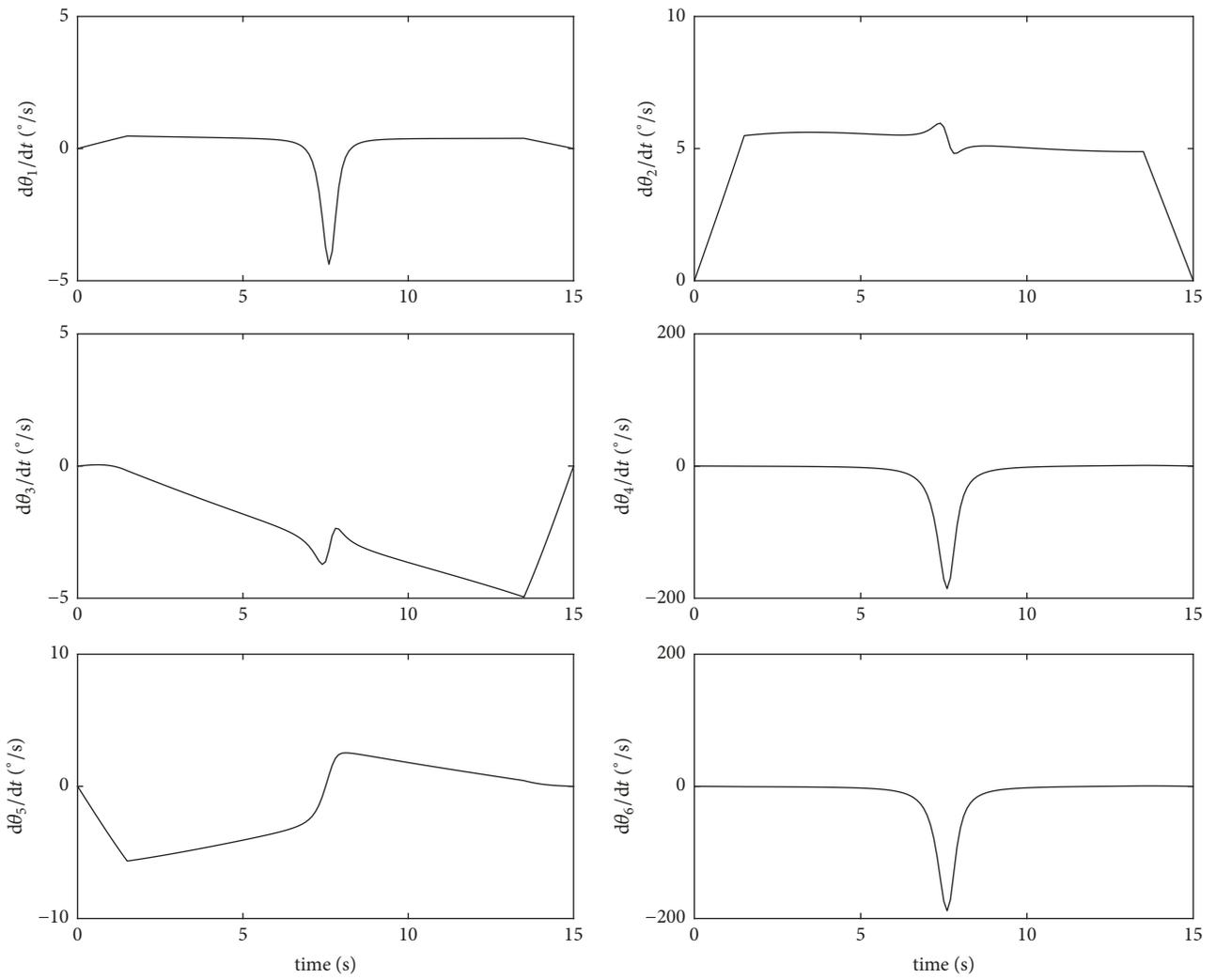


FIGURE 6: The joint velocity path planned by inverse kinematics.

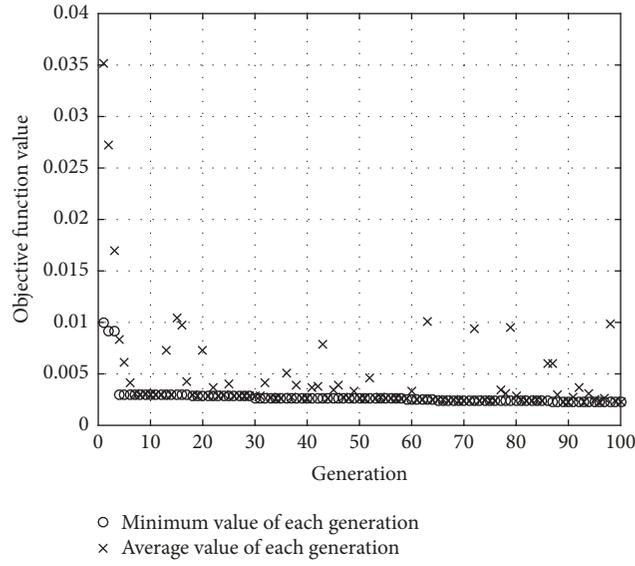


FIGURE 7: Each generation value of optimization.

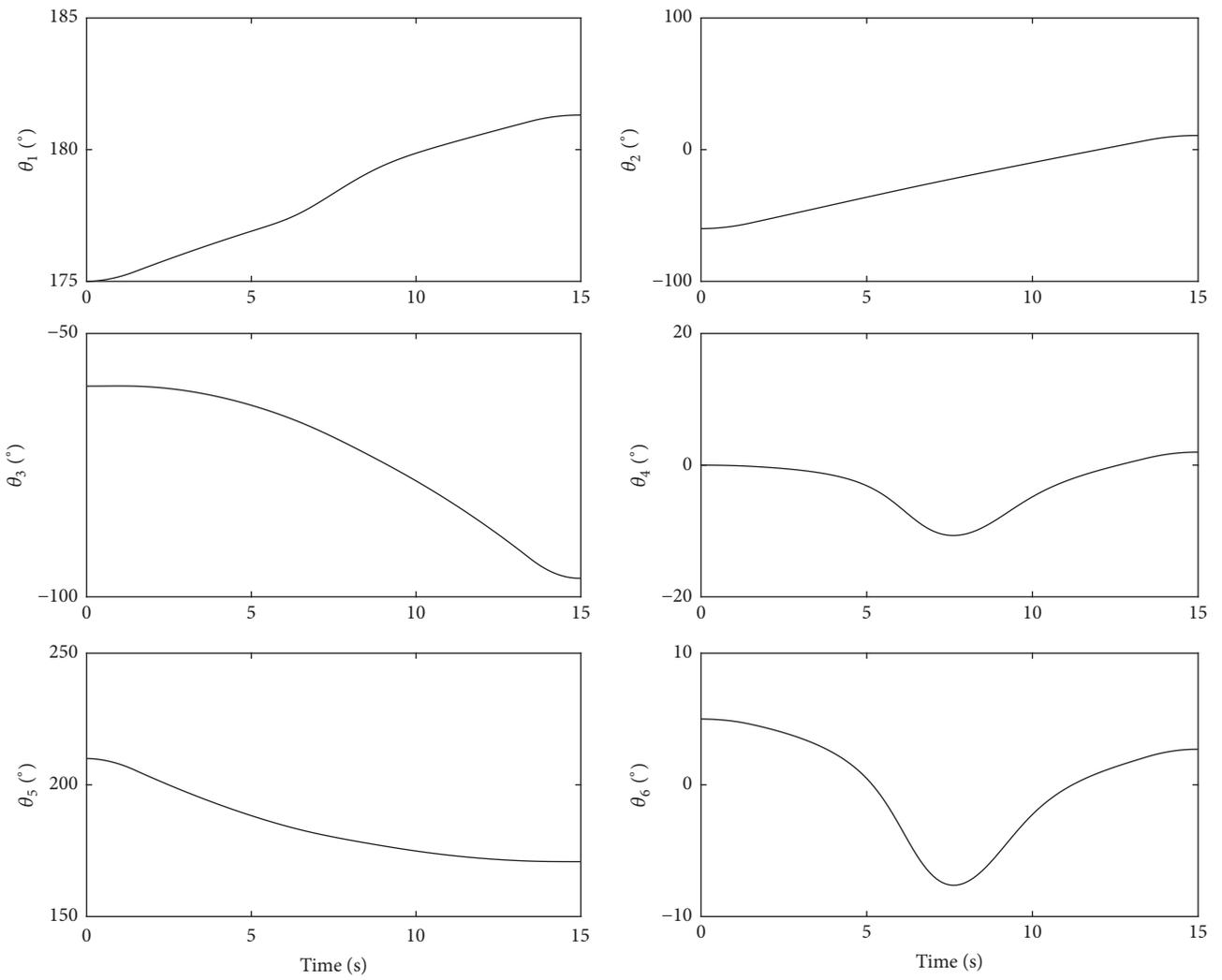


FIGURE 8: The joint position path planned by singularity-robust algorithm.

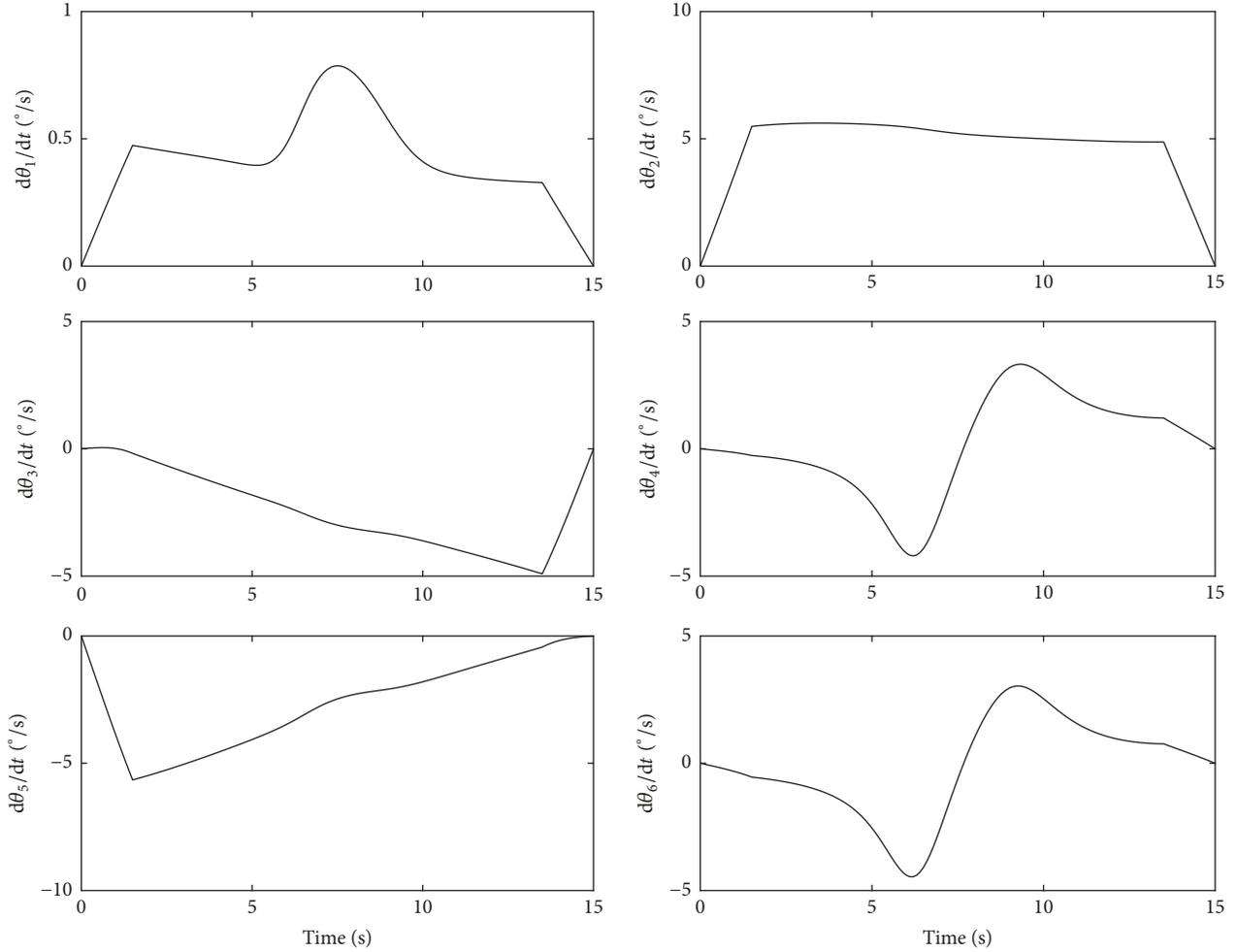


FIGURE 9: The joint velocity path planned by singularity-robust algorithm based on SVD.

joint angle is 175° , -60° , -60° , 0° , 210° , and 5° . In accordance with the practical sampling period, time step length Δt is 250 ms in the simulation. In order to make the motion more smooth, the trapezoidal velocity planning is adopted to the joint motion, as shown in Figure 4.

In the calculation of the path planning, the GJM will come across singularity issue. The determinant value of the GJM is close to zero in 7.5 seconds, as shown in Figure 5. If the calculation through the inverse of GJM is still adopted, the joint velocity will become very large (Figure 6), which is unacceptable in practical application, so it is necessary to employ the singular avoidance method.

In the algorithm mentioned in Section 3.2, the parameters of the objective function need to be determined. In (17) and (19), we determine that $w_1 = 0.5$, $w_2 = 0.5$, $\theta_{+i,\text{limit}} = 6^\circ/\text{s}$, and $\theta_{-i,\text{limit}} = -6^\circ/\text{s}$. Through the GA optimization, the value of objective function is improved in each generation, which is shown as in Figure 7. The optimization results are $\lambda_{\text{max}} = 0.0571$ $\varepsilon = 0.0677$.

The angle and angular velocity of each joint planned by the adaptive singularity-robust algorithm are shown in Figures 8 and 9, respectively. It can be seen that the joints'

velocity is obviously restricted in the singularity region of GJM when the adaptive and optimized damping is introduced in the calculation. The angular velocity of each joint is smooth and the range of each joint velocity is within $6^\circ/\text{s}$.

In order to verify the operability of the calculation results, the planned motion of the free-floating space robot is simulated by software ADAMS (Figure 10). The simulation shows that the position and attitude of the floating carrier are changed with the motion of manipulator, and the Cartesian trajectory tracked by the end-effector is a straight line, which is consistent with the desired trajectory. The tracking errors of position and attitude in three directions are shown in Figure 11. The tracking errors change with the singularity value, and the maximum tracking error is introduced when the singularity value is minimal. The objective function values of DLS method and our method are 0.0030 and 0.0022, respectively. Compared with the DLS method, the error is decreased about 26.7%. The position errors of x and z direction are obviously reduced, and the errors of other directions are basically the same. This shows that in the simulation task presented in this paper the tracking errors of x and z direction

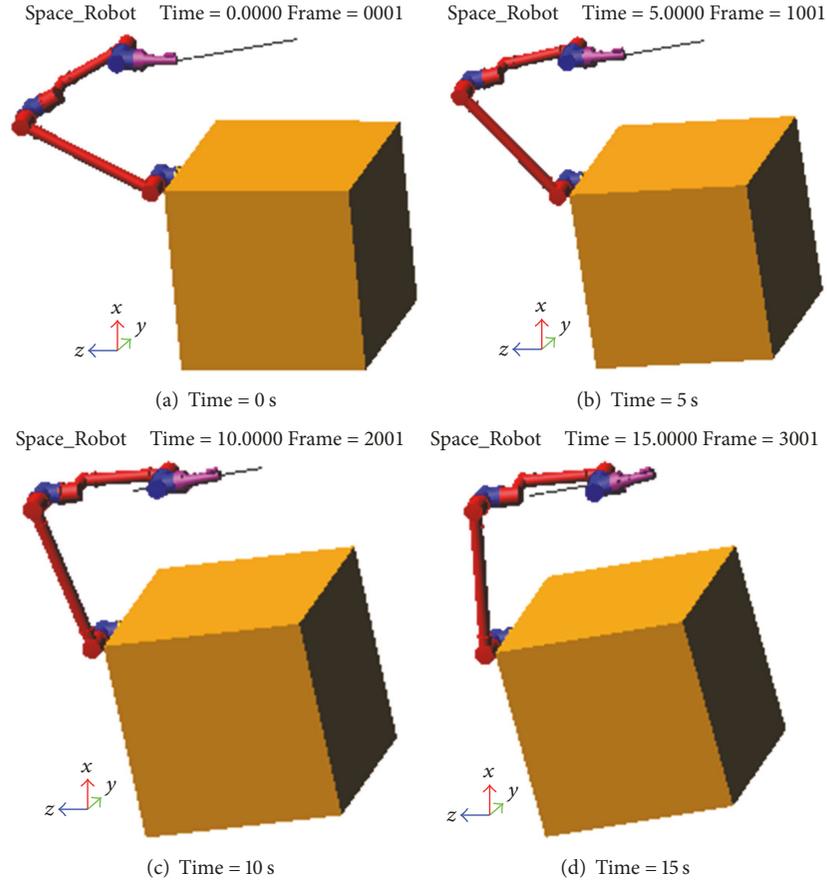


FIGURE 10: 3D dynamic simulation to the Cartesian path planning for space robot.

have a great influence on the value of optimization objective function, which have a good potential to be optimized. In other directions, the proportion of the errors in the objective function is small, so the effect of optimization is not obvious. If it requires a very strict accuracy in a certain direction, we can achieve the requirements by changing the weight coefficient in the objection function.

5. Conclusions

A space robot can be employed to detect the spacecraft surface through accurate tracking the Cartesian trajectory. It is a typical nonholonomic system, whose Cartesian trajectory planning can only be obtained through the velocity level kinematics. Therefore, in the path planning based on the inverse GJM, the singularity issue will possibly be encountered. To solve this issue and improve the Cartesian tracking accuracy, the following work is conducted:

(1) The path planning method of Cartesian trajectory planning is given based on the velocity level inverse kinematics model.

(2) Based on the SVD method, the singular avoidance characteristics of GJM are illustrated. The DLS method is improved by introducing an adaptive damping factor which changed with the singularity. For the method introducing a damping, which is adaptively adjusted by λ_{\max} and ϵ , in the

singular region, the tracking accuracy is not influenced by the singularity-robust algorithm.

(3) In order to improve the tracking accuracy of the singularity-robust algorithm, the objective function including λ_{\max} and ϵ is established, which is optimized by GA.

(4) Through the simulation of a 6-DOF free-floating space robot, the optimized parameters are obtained. Compared with the DLS method, the tracking accuracy of the end-effector is increased by 26.7%.

Data Availability

The modeling data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Grant 51675136, Natural Science Foundation of Heilongjiang Province, China, Grant E2017032, China Postdoctoral Science Foundation (CPSF)

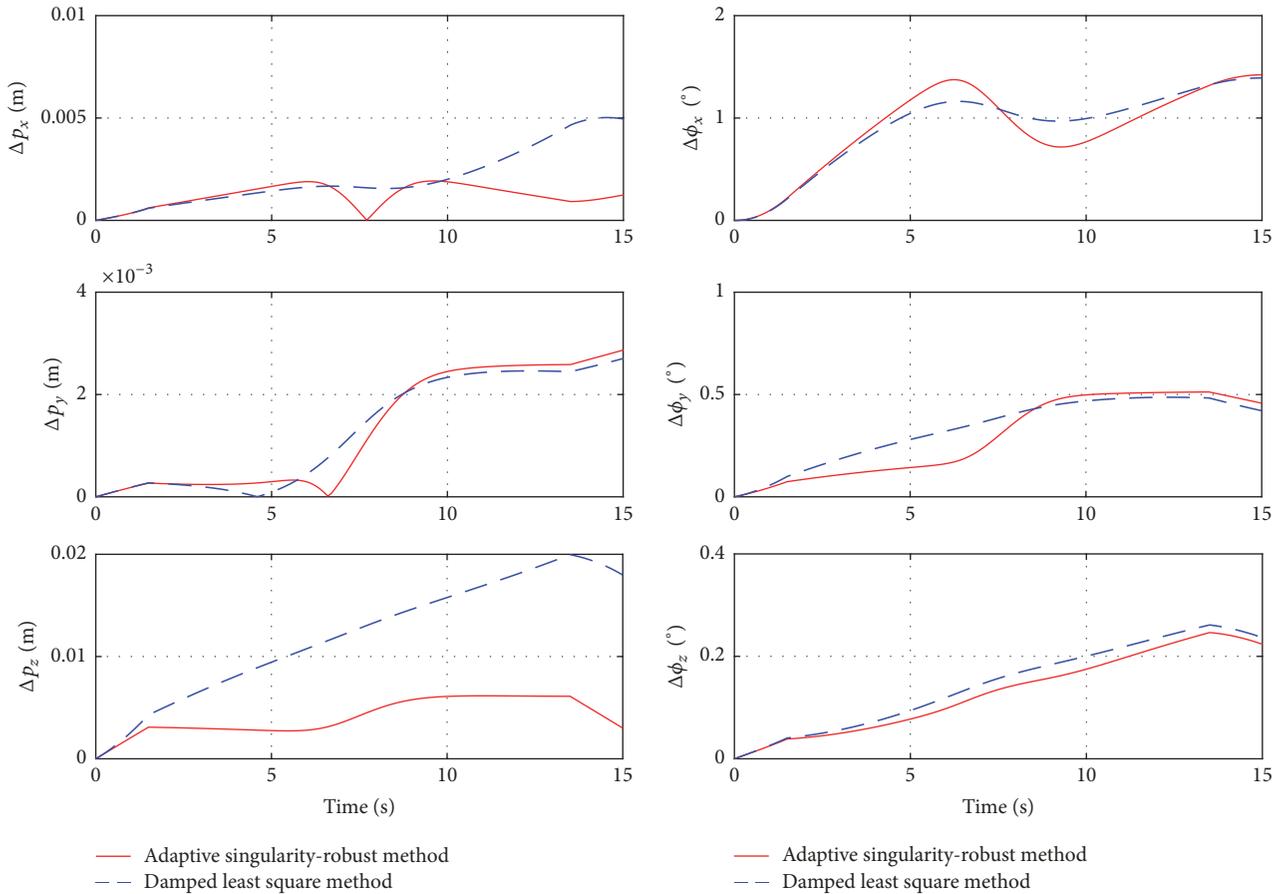


FIGURE 11: The tracking errors comparisons.

Grant 2014M551231, and Heilongjiang Postdoctoral Fund Grant LBH-Z12131.

References

- [1] J. Shi, Y. Cai, and Y. Li, "Research summarizing of on-orbit testing," *Ordnance Industry Automation*, vol. 30, no. 6, pp. 59–62, 2011.
- [2] R. Madison, "Micro-satellite based, on-orbit servicing work at the Air Force Research Laboratory," in *Proceedings of the 2000 IEEE Aerospace Conference Proceedings*, pp. 215–226, Big Sky, MT, USA.
- [3] S. E. Fredrickson, S. Duran, and J. D. Mitchell, "Mini AERCam inspection robot for human space missions," in *Proceedings of the A Collection of Technical Papers - AIAA Space 2004 Conference and Exposition*, pp. 283–291, San Diego, CA, USA, September 2004.
- [4] J. Wu, Y. Zhang, J. Cui, and J. Tan, "Carrier attitude adjustment strategy of a space robot for on-orbit detection," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, Article ID 62246, 2016.
- [5] Y. Umetani and K. Yoshida, "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 303–314, 1989.
- [6] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Springer, New York, NY, USA, 2006.
- [7] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 1986.
- [8] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.
- [9] D. Oetomo and M. H. Ang Jr., "Singularity robust algorithm in serial manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 1, pp. 122–134, 2009.
- [10] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 14, no. 1, pp. 43–68, 1995.
- [11] M. G. Carmichael, D. Liu, and K. J. Waldron, "A framework for singularity-robust manipulator control during physical human-robot interaction," *International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 861–876, 2017.
- [12] H.-X. Cui, K. Feng, H.-L. Li, and J.-H. Han, "Singularity avoidance of 6R decoupled manipulator using improved Gaussian distribution damped reciprocal algorithm," *Industrial Robot: An International Journal*, vol. 44, no. 3, pp. 324–332, 2017.
- [13] W. Xu, J. Zhang, B. Liang, and B. Li, "Singularity analysis and avoidance for robot manipulators with nonspherical wrists,"

- IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 277–290, 2016.
- [14] Y.-L. Lai, C.-C. Liao, and Z.-G. Chao, “Inverse kinematics for a novel hybrid parallel–serial five-axis machine tool,” *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 63–79, 2018.
- [15] E. Papadopoulos and S. Dubowsky, “Dynamic singularities in the control of free-floating space manipulators,” *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 115, no. 1, pp. 44–52, 1993.
- [16] E. Papadopoulos, “Path Planning For Space Manipulators Exhibiting Nonholonomic Behavior,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 669–675, Raleigh, NC, USA, 1992.
- [17] R. Lampariello and K. Deutrich, “Simplified path planning for free-floating robots,” DLR Internal Report 515-99-04, 1999.
- [18] S. Pandey and S. K. Agrawal, “Path Planning of Free-Floating Prismatic-Jointed Manipulators,” *Multibody System Dynamics*, vol. 1, no. 1, pp. 127–140, 1997.
- [19] W. Xu, L. Li, B. Liang et al., “Workspace analysis of space 3R robot,” *Journal of Astronautics*, vol. 28, no. 5, pp. 1389–1394, 2007.
- [20] M.-H. Jin, C. Zhou, Y.-C. Liu, Z.-Q. Liu, and H. Liu, “Reaction torque control of redundant free-floating space robot,” *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 295–306, 2017.

