

Path Optimization of Free-floating Space Robot Based on Adaptive Genetic Algorithm

Junjie Zhang^{1,a*}, Shidong Yao¹, Bo Peng¹, Qingli Zhang¹ and Tao Zhang

¹ China Academy of Launch Vehicle Technology, *Beijing*, 100076, *China*

^aemail, xibeiair@163.com

Keywords: free-flying space robot; Adaptive Genetic Algorithm; joint space; path optimization

Abstract. In the free-floating mode, dynamic coupling exists between the manipulators and the spacecraft base of space robot, and the base attitude and position are easily disturbed by the motions of the manipulators. In this paper, firstly, the angular momentum conservation equation is analyzed. Then, the objective function is defined according to the change of the base attitude. Finally, the joint path is parameterized by using a cosine function and the coefficients of the cosine function are set as optimization parameters. Considering the Standard Genetic Algorithm is easily premature and has slow convergent speed, Adaptive Genetic Algorithm is adopted in this paper to search the optimal parameters. Simulation results show that the obtained joint path is smooth, the angular velocity has no sudden change and the base disturbance meets the demand, which prove the effectiveness of this method.

Introduction

Space robots generally consist of a spacecraft platform (satellite, space shuttle, etc) and one or more manipulators mounted on it. Space robots are under continued concern because of their advantage in on-orbit services, space debris removal and refueling that are inaccessible or too risky for astronauts to act. Compared to the robots on ground, space robots are not fixed at one point, resulting in dynamic coupling between the manipulators and their spacecraft platforms. The motions of the manipulators induce disturbances to the position and attitude of the spacecraft, especially when the ratio between manipulators and the spacecraft is large. Of course, these disturbances can be compensated by thruster jets or momentum wheels that mounted on the spacecraft, but thruster jets are very expensive and momentum wheels have maximum limit, so it is very important to reduce the disturbances to the spacecraft.

The linear momentum and angular momentum of space robots satisfy the conservation equations when no external forces or torques act on the system and when the total momentum of the system is zero, and because of the non-integrability of the angular momentum, the system exhibits nonholonomic behaviors[1], which increases the difficulty of control and trajectory planning of space robots. Vafa and Dubowsky[2] developed the *disturbance map* using the virtual arm modeling method, which displays the motion directions of the manipulators that generate maximum and minimum disturbances to the spacecraft in joint space. Y. Umetani and K. Yoshida[3] proposed the concept of *generalized jacobian matrix* which was used to plan the space robot trajectory, but because of the dynamic singularities the trajectory was not always achieved. E. Papadopoulos[4,5] employed polynomial functions to drive the system to a desired configuration in finite and prescribed time based on mapping the nonholonomic constraint to a space where it can be satisfied trivially. Wenfu Xu[6] studied the stability of the space robot with a balance arm, and presented the concept of *system centric equivalent manipulator*, also two typical applications using this method are given. Yan Li[7] set up the joint angular probability as input variables, and set the least energy cost as the main objective function, and finally the genetic algorithm was applied to get the optimized variables. Zhong Shi[8] used the high order polynomials to approach the trajectories of the joint angles, and set the coefficients of the interpolation polynomial as the optimization parameters. Then particle swarm optimization algorithm was applied to achieve the optimal trajectory. Hongwei Xia[9] obtained the joint trajectories that generate minimum disturbance to the spacecraft based on the chaos particle swarm optimization algorithm.

In this paper, the nonholonomic behaviors are used and only by controlling the joint angles, the manipulators joint angles and the spacecraft attitude can obtain the desired state simultaneously. Since *standard genetic algorithm* is easy to premature convergence, a new algorithm called *adaptive genetic algorithm* is used to get the optimal parameters. Simulation example states that compared to the standard genetic algorithm, adaptive genetic algorithm can converge to the global optimal solution faster. Furthermore, the planned trajectories are smooth and disturbances to the spacecraft are in line with expectations which demonstrates the effectiveness of the proposed method.

Kinematic Equations of the Space Robot System

A single-arm space robot system consists of the spacecraft platform and n-link manipulators, as shown in Fig 1. Assumptions: (1) ignoring the flexibilities of all arms and joints, so they can be taken as rigid; (2) the joints are rotary joints, and only have one degree of freedom; (3) ignoring external forces and moments (gravity, engine thrust, etc.).

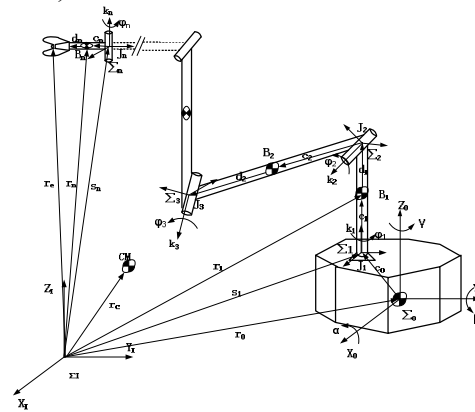


Fig1. The General Model of a Space Robot System

In Fig 1, some symbols and variables are defined as follows:

$\Sigma_0, \Sigma_i (i = 0, 1 \dots n)$	inertia frame and i th body fixed frame
$B_0, B_i (i = 1, 2, \dots, n)$	the spacecraft and i th link of the manipulator
CM	the center of the space robot system
$k_i (i = 1, 2, \dots, n)$	unit vector representing the rotation direction of i th joint
$j_i (i = 1, 2, \dots, n)$	rotation angle of i th joint
$r_i (i = 0, 1 \dots n)$	position vector from the origin of inertia frame to the center of i th link
$s_i (i = 0, 1 \dots n)$	position vector from the origin of inertia frame to the i th joint
$c_i (i = 0, 1 \dots n)$	position vector from the i th joint to the center of i th link
$d_i (i = 0, 1 \dots n)$	position vector from the center of i th link to the $i+1$ th joint

In free-floating mode, according to the proposed assumptions, linear momentum and angular momentum of space robot systems satisfy the conservation law, and the relationship between the spacecraft and the joint angular velocity of the manipulator can be expressed as Eq 1[3].

$$\begin{bmatrix} ME_3 & M\%_{0g}^T \\ M\%_{0c}^T & J_w \end{bmatrix} \begin{bmatrix} v_0 \\ w_0 \end{bmatrix} + \begin{bmatrix} J_{Tw} \\ J_{wf} \end{bmatrix} \Phi = 0. \quad (1)$$

where M is the total mass of the space robot system, v_0 and w_0 denote the linear velocity and angular velocity of the spacecraft, respectively, and $\Phi = [j_1, j_2, \dots, j_n]^T$ denotes the angular velocity of the joints. The other symbols are defined as Eq 2 to Eq 7.

$$J_w = \sum_{i=1}^n (I_i + m_i \%_{0i}^T \%_{0i}) + I_0. \quad (2)$$

$$J_{wf} = \sum_{i=1}^n (I_i J_{Ri} + m_i J_{Ti}). \quad (3)$$

$$J_{Tw} = \sum_{i=1}^n m_i J_{Ti}. \quad (4)$$

$$J_{Ri} = [k_1, k_2, \dots, k_n, 0, \dots, 0]. \quad (5)$$

$$J_{Ti} = [k_1 \times (r_i - p_1), k_2 \times (r_i - p_2), \dots, k_i \times (r_i - p_i), 0, \dots, 0]. \quad (6)$$

$$r_{0c} = r_c - r_0 \quad r_{0i} = r_i - r_0. \quad (7)$$

The Trajectory Planning in Joint Space

Choice of Objective Function. Depending on space missions, the evaluation to the space robot trajectory planning is different. Taking the communication, observation, etc. before and after the motion of the space robot into account, the attitude change of the space robot must be limited. So the objective function can be written as Eq 8.

$$J = \frac{\|dQ\|}{K_Q}. \quad (8)$$

where $\|dQ\| = \sqrt{(dQ)^T dQ}$ denotes the attitude change before and after the motion of the space robot, and K_Q denotes the attitude control accuracy.

As shown in Eq 8, if $J \leq 1$, then it can be taken that the result has achieved convergent requirements.

The Trajectory Planning Method. The trajectory planning in space joint is in fact planning a function of time, the time derivative of the first order and second-order to describe the movement of joints. To ensure the smooth movement of joints, trajectories and their derivatives must be continuous, 4-order or more than 4-order polynomial can meet these requirements [5]. Taking the ranges of the joint angles into account, cosine function can be used to parameterize the joint angles, as shown in Eq 9.

$$j_i(t) = \Delta_{i1} \cos(m_{i7}t^7 + m_{i6}t^6 + m_{i5}t^5 + m_{i4}t^4 + m_{i3}t^3 + m_{i2}t^2 + m_{i1}t + m_{i0}) + \Delta_{i2}. \quad (9)$$

where $m_{i0} \dots m_{i7}$ denotes the unknown coefficients in cosine functions; Δ_{i1} and Δ_{i2} can be defined as shown in Eq 10.

$$\Delta_{i1} = \frac{j_{i\max} - j_{i\min}}{2}, \quad \Delta_{i2} = \frac{j_{i\max} + j_{i\min}}{2}. \quad (10)$$

where $j_{i\max}$ and $j_{i\min}$ denote the upper limit and lower limit of the ranges of the joint angles, respectively.

Then the angular velocity and angular acceleration are the first and second order time derivatives of the joint angle shown in Eq 9, respectively, and are shown as Eq 11 and Eq 12.

$$\dot{j}_i(t) = -\Delta_{i1} \sin(m_{i7}t^7 + m_{i6}t^6 + m_{i5}t^5 + m_{i4}t^4 + m_{i3}t^3 + m_{i2}t^2 + m_{i1}t + m_{i0}) * (7m_{i7}t^6 + 6m_{i6}t^5 + 5m_{i5}t^4 + 4m_{i4}t^3 + 3m_{i3}t^2 + 2m_{i2}t + m_{i1}) \quad (11)$$

$$\begin{aligned} \ddot{j}_i(t) = & -\Delta_{i1} \cos(m_{i7}t^7 + m_{i6}t^6 + m_{i5}t^5 + m_{i4}t^4 + m_{i3}t^3 + m_{i2}t^2 + m_{i1}t + m_{i0}) * (7m_{i7}t^6 + 6m_{i6}t^5 + 5m_{i5}t^4 + 4m_{i4}t^3 + 3m_{i3}t^2 \\ & + 2m_{i2}t + m_{i1})^2 - \Delta_{i1} \sin(m_{i7}t^7 + m_{i6}t^6 + m_{i5}t^5 + m_{i4}t^4 + m_{i3}t^3 + m_{i2}t^2 + m_{i1}t + m_{i0}) * (42m_{i7}t^5 + 30m_{i6}t^4 + 20m_{i5}t^3 + 12m_{i4}t^2 \\ & + 6m_{i3}t + 2m_{i2}) \end{aligned}$$

(12) Considering the conditions of the joints at initial and terminal states, some equations can be obtained as shown in Eq 13.

$$\cdot \quad (12)$$

$$\Phi(t_0) = \Phi_0, \dot{\Phi}(t_0) = 0, \ddot{\Phi}(t_0) = 0, \Phi(t_f) = \Phi_d, \dot{\Phi}(t_f) = 0, \ddot{\Phi}(t_f) = 0 \quad (13)$$

where $\Phi_0 = [j_{10}, j_{20}, \dots, j_{n0}]$.

Then the coefficients in cosine function can be easily derived from Eq9, Eq11, Eq12 and Eq13, as shown in Eq14 to Eq18.

$$m_{i0} = \arccos\left(\frac{j_{i0} - \Delta_{i2}}{\Delta_{i1}}\right) \quad (14)$$

$$m_{i1} = m_{i2} = 0 \quad (15)$$

$$m_{i3} = \frac{1}{t_f^3} (10p - 10(\arccos(\frac{j_{i0} - \Delta_{i2}}{\Delta_{i1}}) - \arccos(\frac{j_{id} - \Delta_{i2}}{\Delta_{i1}}))) - 3m_{i7}t_f^4 - m_{i6}t_f^3 \quad (16)$$

$$m_{i4} = \frac{1}{t_f^4} (-15p + 15(\arccos(\frac{j_{i0} - \Delta_{i2}}{\Delta_{i1}}) - \arccos(\frac{j_{id} - \Delta_{i2}}{\Delta_{i1}}))) + 8m_{i7}t_f^3 + 3m_{i6}t_f^2$$

$$(17) \quad m_{i5} = -\frac{1}{t_f^5} (-6p + 6(\arccos(\frac{j_{i0} - \Delta_{i2}}{\Delta_{i1}}) - \arccos(\frac{j_{id} - \Delta_{i2}}{\Delta_{i1}}))) - 6m_{i7}t_f^2 - 3m_{i6}t_f$$

So, the unknown coefficients in cosine function are only m_{i7} and m_{i6} . If we define $[m_{17}, m_{16}, m_{27}, m_{26}, \dots, m_{n7}, m_{n6}]$, then dQ can be written as $dQ = dQ(m)$. The trajectory planning method of space robot then can be described that, first using the cosine function to parameterize the joint angles, and then employing the optimization algorithm to seek for the optimal m , and last obtaining joint trajectories, angular velocities and angular accelerations using Eq 9, Eq 11 and Eq 12.

Trajectory Planning Based on Adaptive Genetic Algorithm

Genetic algorithm was founded based on Darwin's theory of evolution and Mendelism, which imitated the process of evolution and referred to random statistical theory. Genetic algorithm can be viewed as an evolutionary process wherein a population of solutions evolves over a sequence of generations, and it is hoped that during this evolutionary process the fitness of the population can come near to the optimal value. However, the standard genetic algorithm usually has slow convergent speed and often has premature phenomenon in engineering applications.

In those parameters that affect the performance of genetic algorithm, the cross probability P_c and mutation probability P_m are the most important. During the computing process of standard genetic algorithm, P_c and P_m are fixed. The higher the value of P_c , the quicker are the new solutions introduced into the population, however, exorbitant value of P_c may exploit the solutions which have relatively high fitness values. The lower the value of P_m , the harder are the encodings changed, however, too large value of P_m may transform the genetic algorithm to a purely random search algorithm. To overcome the above-stated problem, in this paper, a new optimal technique called adaptive genetic algorithm is introduced. The main idea of adaptive genetic algorithm is that P_c and P_m can adaptively in response to the fitness values of the solutions, that is, giving lower values of P_c and P_m for those solutions whose fitness values are higher than mean fitness values and giving higher values of P_c and

P_m for those solutions whose fitness values are lower than mean fitness values. Then the expressions for P_c and P_m can be written as in Eq 19 and Eq 20.

$$P_c = \begin{cases} k_1 - \frac{(k_1 - k_2)(f_{\max} - f_c)}{f_{\max} - f_{avg}} & f_c \geq f_{avg} \\ k_1 & f_c < f_{avg} \end{cases} \quad (19)$$

$$P_m = \begin{cases} k_3 - \frac{(k_3 - k_4)(f_{\max} - f_m)}{f_{\max} - f_{avg}} & f_m \geq f_{avg} \\ k_3 & f_m < f_{avg} \end{cases} \quad (20)$$

In order to have a faster convergent speed, the classic roulette method and elite preserving strategy were introduced, and considering that the unknown parameters in m were as many as twelve, two-point crossover method was introduced.

The procedures to plan the trajectory of space robot using adaptive genetic algorithm are shown as follows:

Step 1. Encode the parameters in m with binary code as the initial population and let generation number $gen=1$.

Step 2. Define the fitness function as shown in Eq 21.

$$F = 1 / J \quad (21)$$

where J is the objective function defined in Eq 8.

Step 3. Generating a new population using reproduction, crossover and mutation operators according to the fitness values computed in step 2.

Step 4. Let $gen=gen+1$, and if population number gen is bigger than the maximum population number then jumps to step 5, else jumps to step 2.

Step 5. Output the best solution m .

Simulation Example

In order to verify the effectiveness of the above-stated trajectory planning method, a six-DOF space robot is introduced, and the mathematical model is given, and at last Matlab is used to get the final output. The dynamical parameters of the space robot are shown in Table 1.

Table 1. Dynamical parameters of the space robot

	Mass [kg]	Length [m]		Moment of inertia [kg.m ²]		
		a_i	b_i	Ixx	Iyy	Izz
spacecraft	1700	1.75	1.75	1434	1434	1735
Link 1	5	0.125	0.125	0.0292	0.0292	0.00625
Link 2	50	1.25	1.25	0.0625	26.1	26.1
Link 3	50	1.25	1.25	0.0625	26.1	26.1
Link 4	10	0.25	0.25	0.0125	0.215	0.215
Link 5	5	0.125	0.125	0.00625	0.0292	0.0292
Link 6	5	0.125	0.125	0.0292	0.0292	0.00625

Supposed that the initial joint angles and terminal joint angles are as shown in Eq 22.

$$\Phi_0 = [0^0 \ 50^0 \ -20^0 \ 10^0 \ 75^0 \ 30^0], \quad \Phi_d = [10^0 \ -5^0 \ 10^0 \ 40^0 \ 50^0 \ 0^0]. \quad (22)$$

The ranges of joint angles are shown in Table 2.

Table 2. The ranges of joint angles

Joint	Ranges of joint angles [°]
1	[-100,100]
2	[-120,200]
3	[-150,150]
4	[-150,200]
5	[-100,100]
6	[-200,200]

The initial attitude angles of the spacecraft are $\Psi(0)=[5^0, 5^0, 5^0]^T$, and the terminal attitude angles of the spacecraft are $\Psi(t_f)=[0^0, 0^0, 0^0]^T$, and the planned time is $t_f = 5s$. The adaptive genetic algorithm parameters are set as: initial population number npop=50, generation number nger=200, crossover probability and mutation probability are defined as in Eq 19 and Eq 20.

The standard genetic algorithm and adaptive genetic algorithm are used to seek for the optimal solutions using the above-stated technique, respectively. The results are shown is Eq23 and Eq24, including the objective function values and the resulted m .

$$\begin{cases} J_{SGA} = 0.012 \\ m_{SGA} = [0.09770 \ 0.2001 \ 0.5079 \ -0.9648 \ -0.9019 \ -0.06389 \ 0.1892 \ -0.9434 \\ -0.9998 \ -0.9736 \ 0.3391 \ 0.7420] \times 10^{-3} \end{cases} \quad (23)$$

$$\begin{cases} J_{AGA} = 0.0011 \\ m_{AGA} = [0.08546 \ -0.3300 \ 0.3928 \ 0.6595 \ -0.9288 \ 0.9575 \ -0.7852 \ 0.9115 \\ -0.9138 \ -0.05954 \ -0.9572 \ -0.8650] \times 10^{-3} \end{cases} \quad (24)$$

The best fitness values of each generation of the standard genetic algorithm is shown as Fig 2, and the best fitness values of each generation of the adaptive genetic algorithm is shown as Fig 3.

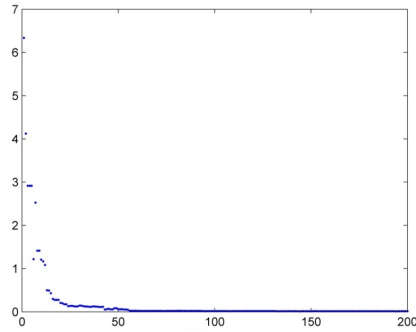


Fig 2. The best fitness values of each generation of the standard genetic algorithm

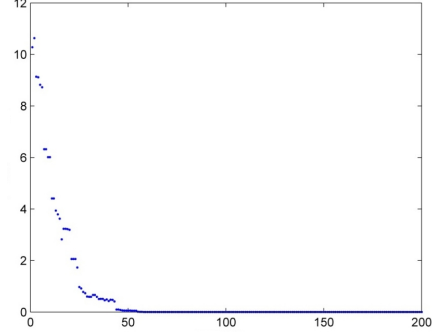


Fig 3. The best fitness values of each generation of the adaptive genetic algorithm

As shown in Fig 3, it is at the 45th generation that the result comes near to the convergent value, however, as shown in Fig 2, it is at the 60th generation that the result comes near to the convergent value. At the end of the computing process, the objective value of the standard genetic algorithm is 0.012, and the objective value of the adaptive genetic algorithm is 0.0011. Thus, the simulation results shows that, compared to the standard genetic algorithm, the adaptive genetic algorithm have a faster convergent speed and the global searching performance is much better.

Taking the resulted m into Eq 9 and Eq 11, we can obtain the joint angles and the joint angular velocities, as shown in Fig 4 and Fig 5, respectively. And the variation of the spacecraft attitude angles is shown in Fig 6.

As shown in Fig 4, the planned trajectories of joint angles are very smooth, and during the motion process the joint angles are limited into the ranges that are allowed, and after the motion the joint angles reached the desired angles. As shown in Fig 5, the joint angular velocities have no sudden change and the changes of the spacecraft attitude angles are relatively small in x and z directions, and

although the change of attitude angle is relatively big in y direction, however, it also reaches the desired angle at the terminal time.

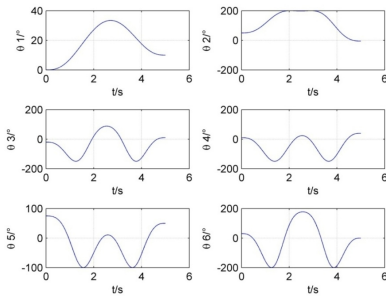


Fig 4. Joint angles vs planned time

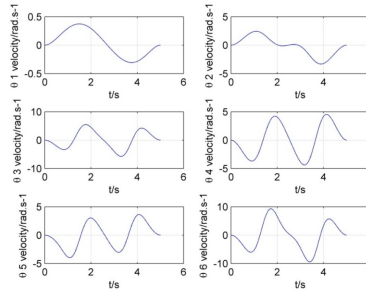


Fig 5. Joint angular velocities vs planned time

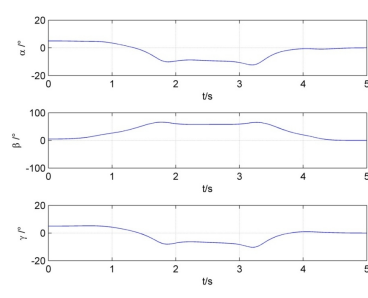


Fig 6. the variation of the spacecraft attitude angles

Summary

(1) By making use of the nonholonomic behaviors of the space robot and using the cosine functions to parameter the joint trajectories, the trajectory optimization problem has transformed into the coefficients optimization problem of cosine functions.

(2) Using the adaptive genetic algorithm to get the optimal results which overcomes the disadvantages of premature and slow convergent speed.

(3) The trajectory planning method is given, and simulation example has verified the effectiveness of the above-stated method.

Reference

- [1] Yoshihiko Nakamura, Ranjan Mukherjee. Nonholonomic. Motion Planning of Free-Flying Space Robots via Bi-Directional Approach[J]. IEEE Transactions on Robotics and Automation, 1991,7(4):505-514
- [2] Z. Vafa, S. Dubowsky. On the Dynamics of Space Manipulator Using the Virtual Manipulator, with Applications to Path Planning[J]. The Journal of the Astronautical Sciences, 1988,38(4):441-472, 1990
- [3] Kazuya Yoshida, Yoji Umetani. Control of Space Manipulators with Generalized Jacobian Matrix. Space Robotics: Dynamics and Control, The Kluwer International Series in Engineering and Computer Science, 1993,188: 165-204
- [4] Ioannis Tortopidis, Evangelos Papadopoulos. On point-to-point motion planning for underactuated space manipulator systems[J]. Robotics and Autonomous Systems (2006), doi:10.1016/j.robot.2006.07.003
- [5] Evangelos Papadopoulos, Ioannis Tortopidis, Kostas Nanos. Smooth Planning for Free-floating Space Robots Using Polynomials[C]. Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain, April 2005
- [6] Wenfu XU, Xueqian Wang, Qiang Xue, Bin Liang. Study on Trajectory Planning of Dual-arm Space Robot Keeping the Base Stabilized[J]. ACTA AUTOMATICA SINICA, 2013, 1
- [7] Yan LI, Yuanwen Cai. The Motion Planning for Free Flying Space Manipulator Based on Genetic Algorithm[J]. Aerospace Control, 2012, 1
- [8] Zhong Shi, Yongzhi Wang, Qinglei Hu. A polynomial Interpolation Based Particle Swarm Optimization Algorithm for Trajectory Planning of Free-Floating Space Robot[J]. Journal of Astronautics, 2011, 32(7)
- [9] Hongwei Xia, Yanbin Zhai, Guangcheng Ya Ma, Deng, Changhong Wang. Path planning algorithm of space manipulator based on chaos particle swarm optimization algorithm[J]. Journal of Chinese Inertial Technology, 2012, 22(2)
- [10] M. Srinivas, L. M. Patnaik. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms[J]. IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4)

