

Research Article

Path Planning for a Space-Based Manipulator System Based on Quantum Genetic Algorithm

Zhengcang Chen^{1,2} and Weijia Zhou²

¹State Key Laboratory of Robotics, Shenyang Institute of Automation, The Chinese Academy of Sciences (CAS), Shenyang 110016, China

²University of Chinese Academy of Science, Beijing, China

Correspondence should be addressed to Zhengcang Chen; chenzhengcang@sia.cn

Received 17 November 2016; Revised 20 January 2017; Accepted 15 February 2017; Published 9 March 2017

Academic Editor: Shahram Payandeh

Copyright © 2017 Zhengcang Chen and Weijia Zhou. 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.

In this study, by considering a space-based, n -joint manipulator system as research object, a kinematic and a dynamic model are constructed and the system's nonholonomic property is discussed. In light of the nonholonomic property unique to space-based systems, a path planning method is introduced to ensure that when an end-effector moves to the desired position, a floating base achieves the expected pose. The trajectories of the joints are first parameterized using sinusoidal polynomial functions, and cost functions are defined by the pose deviation of the base and the positional error of the end-effector. At this stage, the path planning problem is converted into a target optimization problem, where the target is a function of the joints. We then adopt a quantum genetic algorithm (QGA) to solve this objective optimization problem to attain the optimized trajectories of the joints and then execute nonholonomic path planning. To test the proposed method, we carried out a simulation on a six-degree-of-freedom (DOF) space-based manipulator system (SBMS). The results showed that, compared to traditional genetic optimization algorithms, the QGA converges more rapidly and has a more accurate output.

1. Introduction

During space missions, astronauts face a harsh and life-threatening environment in space, especially in the course of extravehicular activities. A growing number of researchers think it inevitable that space manipulators will be used, autonomously or semiautonomously, in future missions to perform tasks either completely or incompletely [1–3]. With the rapid development of space technologies, a number of tasks can be implemented by space manipulator systems. At present, some manipulators are installed on a base whose position or attitude is controlled by reaction wheels or thrusters, taking a satellite-manipulator system, for instance. This kind of system has the following disadvantages: (1) the position and pose control system (PACS) consumes fuel, which shortens the lifespan of the entire space mission. (2) Gas emitted by the thruster might negatively impact the surrounding objects. (3) When the PACS is used, a transient response appears that may impact the dynamic performance of the system. (4) The pose control system can easily attain a saturated state.

In order to solve this problem, many researchers use space-based systems to replace the ones whose base is installed with PACS. However, there exists a highly dynamic coupling phenomenon between a floating base and a manipulator in a space-based system, especially when both have a similar mass. This affects the control precision of the end-effector of the manipulator because the space-based manipulator system (SBMS) is nonholonomic. This study focuses on using the nonholonomic property to plan the path of joints such that the end-effector and the floating base simultaneously arrive at the desired state.

Unlike fixed systems, the position and pose of the end-effector of a manipulator in space-based systems are dependent not only on the given joint angles, but also on the joint paths. In other words, for the same joint configuration, different joint paths correspond to different poses. The authors of [4, 5] studied the course of the development and the status of nonholonomic systems as well as their principles of dynamics and control methods. Rathee and Pathak [6] present path planning of dual arm free flying space

robot using smooth functions of time. Results of their works demonstrate that if the base mass and inertia are increased to a certain limit keeping in mind the singularities, then base disturbance can be decreased and the end effectors follow the required circular path. Zeng et al. [7] present a subsection algorithm for obstacle avoidance path planning of free-floating space manipulator for orbit observation. The approach could guarantee that the end-effector reaches the desired point and realize the obstacle avoidance. Kaigom et al. [8] propose a Constrained Particle Swarm Optimization algorithm to steer the end-effector to a target pose with minimized base disturbance. This approach is based on Reaction Null-Space concept. And its simulation results demonstrate the effectiveness of the planning method. Liu et al. [9] analyze the path planning of redundant, free-floating space manipulators with revolute joints, and 7 degrees of freedom. They provide a tractable planning method to avoid the pseudo-inverse of the Jacobian matrix. Nenchev [10] summarizes results based on the application of Reaction Null-Space approach. Via the null-space and pseudo-inverse mapping of the coupling inertia, the joint space is decomposed into two complementary orthogonal subspaces which is useful for motion analysis and control of various unfixed-base systems. Jia et al. [11] present a novel path planning method to avoid obstacle based on A* algorithm for space manipulators. In this article, a concept of “collision-free workspace” of a space manipulator is provided. Shi et al. [12] propose a nonholonomic motion planning methodology for free-floating space robot (FFSR) using swarm optimization algorithm. Also, an optimal control technique is applied for FFSR trajectory planning. Zhang [13] proposed a path planning method for a free-floating space robot in Cartesian space. Shi [14] proposed a quantum particle swarm optimization method to plan path for space robots, and Jia et al. [15] developed a straight-line trajectory optimization method for a redundant robot with nine degrees of freedom (DOF).

In this study, based on an analysis of nonholonomic properties, we first determine the coupling relationship between a base and a manipulator. We consider the pose of the base and the position of the end-effector as the optimized targets. We then parameterize the joint trajectories using sinusoidal polynomial functions, set ranges of value for the joint angles, rate, and acceleration, and convert the path planning problem into a target optimization problem. Finally, we make use of a QGA to resolve and obtain the optimal parameters as well as an expression of the joint trajectory. To test the effectiveness of the proposed method, we conducted a forward simulation to solve for the optimal target.

2. Kinematic Model of a Space-Based Manipulator System

According to the modeling theory of unrooted multibody systems, with respect to coordinate transformation, a new modeling approach is proposed.

2.1. Description of SBMS. An SBMS consists of an n -joints manipulator and a floating base. Its simplified schematic diagram is shown in Figure 1:

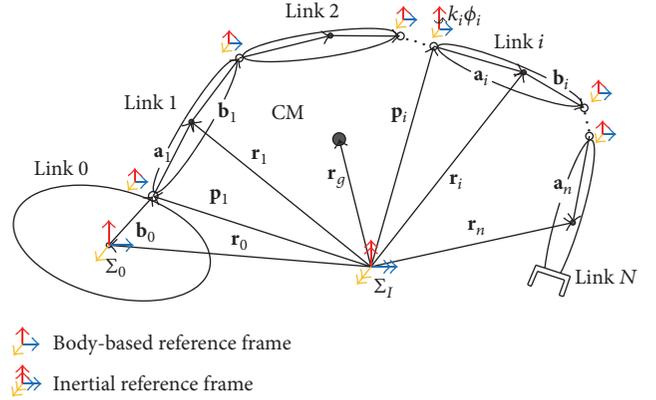


FIGURE 1: Structure diagram of a space-based manipulator system.

- (1) Each component of the system is rigid and numbered from 0 to N starting from the base.
- (2) For simplicity, each joint between link i and link $i + 1$ has one DOF.
- (3) The entire system is in microgravity without an external force exerting on it.
- (4) The system has a tree topology.
- (5) Σ_i and Σ_I are, respectively, the body-fixed reference frame and the inertial reference frame. In general, the origin of the inertial reference frame and the center of mass of the SBMS overlap to some extent in the initial state, and the direction of Σ_0 coincides with that of Σ_I .
- (6) \mathbf{a}_i represents the position vector relating joint i to the CM of link i , and \mathbf{b}_i represents that relating the CM of link i to the next joint.

The selection of coordinate systems is a precondition of modeling. A mixed coordinate system consisting of related and reference coordinates is chosen in this study. In particular, $\mathbf{x}_0 \in \mathbb{R}^3$ and $\theta \in \mathbb{R}^3$ represent the position and pose of the base, and $\phi \in \mathbb{R}^n$ represents the joint angles here.

2.2. Kinematic Modeling. We define dependent coordinates $\mathbf{Y}_{ci} = [\mathbf{x}_i^T, \theta_i^T]^T$, $i = 0, \dots, n$ (called the positional screw), mixed independent coordinates $\mathbf{y}_p = [\mathbf{x}_0^T, \theta_0^T, \phi^T]^T$, and the dependent velocity screw as $\dot{\mathbf{Y}}_i = [\mathbf{v}_i^T, \omega_i^T]^T$.

The constraint equation of an n -joint open-loop serial rigid body system is written as

$$\Phi(\mathbf{Y}_c) = 0 \in \mathbb{R}^m, \quad (1)$$

where $\mathbf{Y}_c = [\mathbf{Y}_{c0}^T, \dots, \mathbf{Y}_{cn}^T]^T \in \mathfrak{R}^{6n+6}$ and m represents the number of constraint equations. The velocity screw is not just a differential of positional screw. It is

$$\dot{\mathbf{Y}}_c = \mathbf{U}(\mathbf{Y}_c) \dot{\mathbf{Y}}. \quad (2)$$

When a different reference frame is selected, the expression of \mathbf{U} also differs: see [16].

We differentiate (1) and substitute (2) into it:

$$\Phi_Y \dot{Y} = 0, \quad (3)$$

where $\Phi_Y = \partial\Phi/\partial Y \in \mathfrak{R}^{m \times (6n+6)}$ represents the Jacobian of the constraint equations.

Since the DOF of a system is equal to the difference between the number of dependent coordinates and that of the constraint equations—that is, $f = (6n+6) - m$ —the kernel of Φ_Y is spanned by f linearly independent column vectors. The basic solutions of this vector space are defined as $\{\mathbf{S}_1, \dots, \mathbf{S}_f\}$. Hence, the dependent velocity screw can be expressed as

$$\dot{Y} = \mathbf{S}_1 t_1 + \mathbf{S}_2 t_2 + \dots + \mathbf{S}_f t_f, \quad (4)$$

where $S = [\mathbf{S}_1, \dots, \mathbf{S}_f] \in R^{(6n+6) \times (n+6)}$ is called the basic solution matrix. Let $\dot{y} = \mathbf{t} = [t_1, \dots, t_{n+6}]^T$, which is an independent velocity vector. The velocity screw of the n th link can be written as

$$\dot{Y}_n = \mathbf{B}_{n,0} \dot{Y}_0 + \mathbf{J}_n \dot{\phi}. \quad (5)$$

Since the system is in free-floating state, momentum is conserved and is assumed to be zero:

$$\dot{Y}_0 = \begin{bmatrix} \mathbf{J}_{Tb\phi} \\ \mathbf{J}_{Rb\phi} \end{bmatrix} = \begin{bmatrix} \frac{-\mathbf{J}_{T\omega}}{\omega} + \tilde{\mathbf{r}}_{g0}^T \mathbf{I}_s^{-1} \mathbf{I}_m \\ -\mathbf{I}_s^{-1} \mathbf{I}_m \end{bmatrix} \dot{\phi}. \quad (6)$$

Connecting (5) and (6), the velocity-level differential equation is written as

$$\dot{Y}_n = \mathbf{J}^*(\phi) \dot{\phi}, \quad (7)$$

where \mathbf{J}^* represents the generalized Jacobian matrix, the concrete representation of which is

$$\mathbf{J}^* = \left\{ \begin{bmatrix} \mathbf{E}_3 & \tilde{\mathbf{r}}_{n,0}^T \\ 0 & \mathbf{E}_3 \end{bmatrix} \begin{bmatrix} \frac{-\mathbf{J}_{T\omega}}{\omega} - \tilde{\mathbf{r}}_{g0}^T \mathbf{I}_s^{-1} \mathbf{I}_m \\ -\mathbf{I}_s^{-1} \mathbf{I}_m \end{bmatrix} + \mathbf{J}_n \right\},$$

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{J}_{Ti} \\ \mathbf{J}_{Ri} \end{bmatrix} = [\mathbf{B}_{i,1} \boldsymbol{\eta}_1 \quad \dots \quad \mathbf{B}_{i,i-1} \boldsymbol{\eta}_{i-1} \quad \boldsymbol{\eta}_i \quad 0 \quad \dots \quad 0]$$

$$\in \mathfrak{R}^{6 \times n},$$

$$\mathbf{B}_{i,j} = \begin{bmatrix} \mathbf{E}_3 & \tilde{\mathbf{r}}_{i,j}^T \\ 0 & \mathbf{E}_3 \end{bmatrix},$$

$$\boldsymbol{\eta}_i = \begin{bmatrix} \mathbf{k}_i \times \mathbf{a}_i \\ \mathbf{k}_i \end{bmatrix},$$

$$\mathbf{J}_{T\omega} = \sum_{i=1}^n \mathbf{m}_i \mathbf{J}_{Ti},$$

$$\mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j,$$

$$\mathbf{I}_m = \sum_{i=1}^n (\mathbf{I}_i \mathbf{J}_{Ri} - \mathbf{m}_i \tilde{\mathbf{r}}_i^T \mathbf{J}_{Ti}) - \tilde{\mathbf{r}}_g \mathbf{J}_{T\omega},$$

$$\mathbf{I}_s = \mathbf{I}_0 + \sum_{i=1}^n (\mathbf{I}_i + \mathbf{m}_i \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_{i0}^T) - \omega \tilde{\mathbf{r}}_g \tilde{\mathbf{r}}_{0g}^T. \quad (8)$$

During nonholonomic path planning, the integral and iterative processes based on (6) and (7) can implement path following a floating base.

2.3. Analysis of Nonholonomic Properties. In order to better understand the coupling phenomenon between a floating base and a manipulator installed on it, an SBMS is here demonstrated as being nonholonomic [17].

Definition 1. A linear subspace spanned by one or more n -dimensional unsteady vectors is called a distribution.

Definition 2. If the result of a Lee bracket operation between two random vectors belonging to a distribution also belongs to it, the distribution is deemed involutive.

Frobenius' Theorem. The necessary and sufficient condition for the integrality of the Pfaffian constraint equations is that their solution space is an involutive distribution.

The state variables are defined as the angle velocity of the base and the joint angle velocity of the manipulator; that is, $\dot{\mathbf{x}}_z = [\boldsymbol{\omega}_0^T, \dot{\boldsymbol{\phi}}^T]^T$. The input value is defined as $\mathbf{u} = \dot{\boldsymbol{\phi}}$. From (2) and (6),

$$\dot{\mathbf{x}}_z = \mathbf{K} \mathbf{u}, \quad (9)$$

where the coefficient matrix \mathbf{K} can be written as $\mathbf{K} = \begin{bmatrix} \mathbf{U}^{-1}(\boldsymbol{\theta}_0)_{Rb\phi} \\ \mathbf{E}_n \end{bmatrix}$ and $\boldsymbol{\theta}_0$ represents the pose angle.

A generalized velocity distribution Δ consists of n columns of \mathbf{K} ; that is, $\Delta = \text{span}(\mathbf{K}_1, \dots, \mathbf{K}_n)$, where $\mathbf{K}_i = (k_{1i}, k_{2i}, k_{3i}, \mathbf{e}_i^T)^T$ and \mathbf{e}_i^T represents the standard unit vector. The result of the Lee bracket between \mathbf{K}_i and \mathbf{K}_j is

$$[\mathbf{K}_i, \mathbf{K}_j] = [k_{1ij}, k_{2ij}, k_{3ij}, \mathbf{0}^T]^T \notin \Delta. \quad (10)$$

Hence, Δ is not involutive. According to the Frobenius theorem, SBMS has been shown to be a nonholonomic system.

3. Transformation of Path Planning Problem to Objective Optimization Problem

3.1. Solving the Objective Function. As we know, the whole system is composed of a floating base, a macroarm, and a dexterous robotic hand. During objective capturing process, the macroarm and dexterous hand are fixed together. The main topic of this text is to make the dexterous hand (or a special point of the macroarm) arrive a desired position by planning the joint path of the macroarm. As for how and with

what attitude to capture an objective by the dexterous hand, it is not included in the topics. Also, attitude of the floating base has an important influence on communication quality and solar conversion efficiency. So, the focus of this study is to plan joint trajectory in order to enable the posture of the base and the position of the end-effector arrive at their desirable states simultaneously. The relevant variables include the quaternion of the base and the position vector of the end link, which are written as $\mathbf{X} = [\mathbf{Q}_b^T, \mathbf{P}_e^T]^T \in \mathbf{R}^6$. \mathbf{X}_0 , \mathbf{X}_d , and \mathbf{X}_f are, respectively, the initial state, the desired state, and the final state.

Steps of path planning are as follows:

- (1) Transformation of state variables ($\mathbf{X} \rightarrow \mathbf{X}' = [\mathbf{Q}_b^T, \boldsymbol{\phi}^T]^T$). End-effector positional vector with respect to the base-linked reference frame is only dependent on joint angles vector $\boldsymbol{\phi}$; that is, ${}^0\mathbf{P}_e^T = f_1(\boldsymbol{\phi})$, while the homogenous transformation matrix from inertial reference frame to base-linked one is related to attitude of the base and the base's positional vector \mathbf{r}_b ; that is, ${}^I\mathbf{A}_0 = f_2(\mathbf{Q}_b^T, \mathbf{r}_b)$. Assume that $\mathbf{r}_b = \mathbf{r}_{b0}$ (a constant) at $t = 0$. According to momentum conservation law, vector \mathbf{r}_b is related to history of joint angles; that is, $\mathbf{r}_b = \mathbf{r}_b(\boldsymbol{\phi})$. So state variable is transformed by $\mathbf{P}_e^T = {}^I\mathbf{A}_0 \cdot {}^0\mathbf{P}_e^T = f_3(\mathbf{Q}_b^T, \boldsymbol{\phi})$.
- (2) Parameterized joint trajectory using sinusoidal polynomial functions.
- (3) Design of objective function based on the deviation between the final state and the desire state.

The destination of path planning is as follows: the planned joint trajectory $\boldsymbol{\phi}(t)$ can render true the condition whereby the final state variables can infinitely approach the desired variables; that is, $\|\mathbf{X}'_d - \mathbf{X}'_f\| \rightarrow 0$.

3.2. Parameterization of Joint Angles. Using the sinusoidal function to parameterize the joint angles can better constrain their scope. To ensure that the velocity and acceleration curves are smooth, we define

$$\phi_i(t) = a_i \sin(b_{17}t^7 + \dots + b_{11}t + b_{10}) + c_i. \quad (11)$$

There are six constraint conditions: $\phi_i(0) = \phi_0$, $\dot{\phi}_i(0) = 0$, $\ddot{\phi}_i(0) = 0$, $\phi_i(t_f) = \phi_{id}$, $\dot{\phi}_i(t_f) = 0$, and $\ddot{\phi}_i(t_f) = 0$. The range of ϕ_i is $\phi_i \in [\phi_i | \phi_{imin} < \phi_i < \phi_{imax}]$.

So, $a_i = (\phi_{imax} - \phi_{imin})/2$ and $c_i = (\phi_{imax} + \phi_{imin})/2$.

Then, joint angle velocity and acceleration can be written as

$$\begin{aligned} \dot{\phi}_i(t) &= a_i \cos(b_{17}t^7 + \dots + b_{11}t + b_{10}) \\ &\quad \cdot (7b_{17}t^6 + \dots + b_{11}), \\ \ddot{\phi}_i(t) &= -a_i \sin(b_{17}t^7 + \dots + b_{11}t + b_{10}) \\ &\quad \cdot (7b_{17}t^6 + \dots + b_{11})^2 \end{aligned}$$

$$\begin{aligned} &+ a_i \cos(b_{17}t^7 + \dots + b_{11}t + b_{10}) \\ &\quad \cdot (42b_{17}t^5 + \dots + 2b_{12}). \end{aligned} \quad (12)$$

Equation (11) contains eight unknown parameters: $b_{10} \sim b_{17}$. With six constraint conditions, six of the eight unknowns can be eliminated. Let $\mathbf{b}_i = [b_{17}, b_{16}]$ and $\mathbf{b} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$. The deviation of the state variables can be written as

$$\Delta\mathbf{X}' = \mathbf{X}'_d - \mathbf{X}'_f = \left([\Delta\mathbf{Q}_b^T, \Delta\boldsymbol{\phi}^T]^T \right). \quad (13)$$

Then, the objective function is defined as

$$f(\mathbf{b}) = \frac{\text{norm}(\Delta\mathbf{Q}_b^T)}{K_Q} + \frac{\text{norm}(\Delta\boldsymbol{\phi}^T)}{K_\phi}. \quad (14)$$

At this time, the path planning problem in SBMS has been transformed into an optimization problem of nonlinear function $f(\mathbf{b})$. We choose the QGA to solve it.

4. Using QGA to Solve Optimization Problem

The quantum genetic algorithm [18] is a method that combines quantum computation and a genetic algorithm, which is a probabilistic evolutionary algorithm.

4.1. Theory of QGA. The genetic algorithm is a common method to solve complicated optimization problems. In 1998, Dr. Bagley proposed the concept of the "genetic algorithm" in his doctoral dissertation and developed a self-adaptation genetic algorithm [19]. The basic idea underlying this algorithm is to learn the jungle law in biological evolution and selection, as well as the crossover or mutation of chromosomes. It is robust and has a wide range of applicability. However, its iterations is much, its convergent rate is low, and it is easily caught in a local optimum.

On the contrary, quantum computation involves solving an NP problem in classical computation using superposition, entanglement, and the interference of quantum-state information units. In 1994, Shor first proposed the quantum algorithm [20].

The QGA is a genetic algorithm based on the expression of a quantum-state vector. The probability amplitude expression of the quantum bit is applied to a chromosome's encoding process, because of which the chromosome can be expressed through the superposition of quantum states whose update process is conducted by means of a quantum rotate gate. Target optimization is thus completed.

The quantum bit is a two-state system in a superposed state, such as $|\varphi\rangle = \alpha|0\rangle\beta|1\rangle$, where (α, β) is a set of amplitude constants that satisfies $|\alpha|^2 + |\beta|^2 = 1$. States $|0\rangle$ and $|1\rangle$, respectively, represent the spin-up and spin-down states. That is, a quantum bit contains information concerning States $|0\rangle$ and $|1\rangle$.

The QGA is updated through the quantum rotate gate, which is expressed as $\mathbf{U}(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}$. The principle

of update is $\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}_{\text{update}} = \mathbf{U}(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$, where $[\alpha_i, \beta_i]^T$ and $[\alpha_i, \beta_i]^T_{\text{update}}$ represent probability amplitudes before and after the update, respectively, and θ_i represents the spin angle.

Thus, the key differences between the quantum genetic algorithm and traditional genetic algorithm are that the former has more abundant population diversity and better parallel calculating abilities with larger search range and convergence rate. This is why its adaptation and efficiency is much higher.

4.2. Brief Introduction to QGA Performance

- (1) Compared to the genetic algorithm, QGA has several new features.
- (2) It uses the global searching method, which enlarges the search scope and increases search efficiency.
- (3) It effectively resists premature phenomena and has better convergence and precision.
- (4) With increase in the number of iterations, the global search capability decreases and local search ability increases, which can cause the system to converge to the global optimum.
- (5) The quantum system is a complex nonlinear system that can adapt to practical issues.

4.3. *Path Planning Based on QGA.* The path planning process of an SBMS based on the QGA is as follows:

- (1) Set known parameters, such as mass, inertia, length, and the bound range of the joint angles. Input the initial state variable, the desired state variable, and the planning time.
- (2) Parameterize the joint trajectory, and determine the type and number of parameters.
- (3) Solve the final state variable. Compute the difference between the final state and the desired state.
- (4) Define parameters mentioned in (2) as chromosomes, and initialize them randomly to obtain a population composed of n quantum bits.
- (5) Measure the population and encode it.
- (6) Design the objective function based on state difference.
- (7) Solve the fitness of each individual and record the results.
- (8) Judge whether the iterative process has end. The basis is the fitness constraint condition and the maximum number of iterations.
- (9) Update individuals by quantum rotate gate, and perform the circuit operation along (5)~(8) until it jumps from the loop.

The corresponding flowchart is shown in Figure 2.

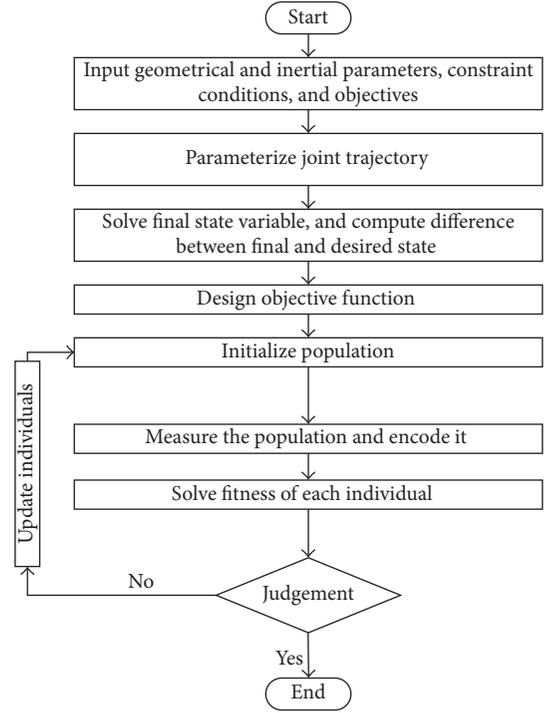


FIGURE 2: Flowchart of QGA.

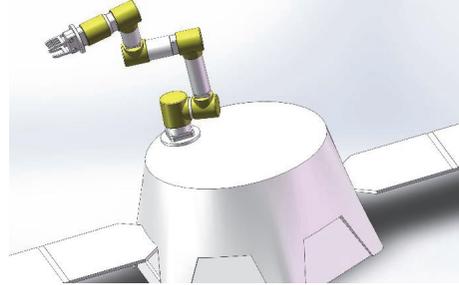


FIGURE 3: A space-based, 6-DOF manipulator system.

5. Simulation Test

In this section, we consider a 6-DOF SBMS as an example, the 3D model of which is shown in Figure 3, for a simulation test to test the proposed path planning method. We also compared it with the traditional genetic algorithm.

The geometrical and inertial parameters of the system are shown in Figure 3 and Table 1. The D-H model is shown in Figure 4: it depicts the deployable state, also called the zero state. The D-H parameters are shown in Table 2. The initial and desired poses of the base are represented by the quaternion as follows:

$$\begin{aligned} \mathbf{Q}_{b0}^T &= [1, 0, 0, 0], \\ \mathbf{Q}_{bd}^T &= [0.9727, -0.1877, 0.0782, 0.112]. \end{aligned} \quad (15)$$

TABLE 1: Geometrical and inertial parameters of a system.

Number	Mass/kg	Length/m		Inertia tensor/kg*m ²					
				I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
0	100.2	—	—	10.12	10.05	9.88	0	0	0
1	10.1	\mathbf{a}_1	0.554	1.012	0.099	0.102	0	0	0
		\mathbf{b}_1	0.554						
2	10.3	\mathbf{a}_2	0.554	1.012	0.099	0.102	0	0	0
		\mathbf{b}_2	0.554						
3	9.5	\mathbf{a}_3	0.554	1.012	0.099	0.102	0	0	0
		\mathbf{b}_3	0.554						
4	9.8	\mathbf{a}_4	0.554	1.012	0.099	0.102	0	0	0
		\mathbf{b}_4	0.554						
5	9.9	\mathbf{a}_5	0.489	1.01	0.110	0.103	0	0	0
		\mathbf{b}_5	0.489						
6	7.5	\mathbf{a}_6	0.412	0.808	0.1	0.099	0	0	0
		\mathbf{b}_6	0.412						

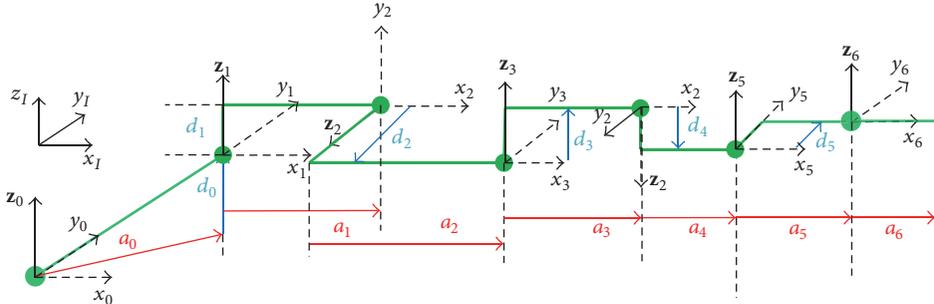


FIGURE 4: D-H frames of space-based systems.

TABLE 2: D-H parameters of a space-based manipulator system.

Link number	θ_i/rad	α_i/rad	a_i/m	d_i/m
0	0	0	1.3	0
1	$\pi/2$	0	1.02	0.433
2	$-\pi/2$	0	1.02	0.433
3	π	0	1.02	0.433
4	$-\pi$	0	1.02	0.433
5	0	$124/180*\pi$	0.81	0.548
6	0	0	0.808	0

The initial and desired positions of the end-effector are

$$\begin{aligned} \mathbf{P}_{e0}^T &= [5.5, 1.05, 0], \\ \mathbf{P}_{ed}^T &= [4.936, 2.7622, -0.1242]. \end{aligned} \quad (16)$$

According to the two preconditions above, we transformed them into another form that incorporated joint configuration. Based on the requests (positional error of end-effector $k_{pe} \leq 0.05$ m and attitude error of the base $k_{rb} \leq 5^\circ$) of positional precision of the floating base and the attitude precision of the end-effector, we then defined the positional coefficient $k_1 = 0.4$, $k_{pe} = 0.02$, and attitude coefficient

$k_2 = 0.0175$. According to formula (12) and the boundary values of it, the value ranges for each parameter were set as

$$\begin{aligned} b_{17} &\in [-3 \times 10^{-3}, 3 \times 10^{-3}], \\ b_{16} &\in [-3 \times 10^{-3}, 3 \times 10^{-3}]. \end{aligned} \quad (17)$$

The optimization of the objective functions was executed by the QGA and the genetic algorithm. The results, including the objective values and optimal parameters, are given below.

Results of the Genetic Algorithm. The optimized parameters were $\mathbf{b}_{GA} = [0.448, 1.973, 2.309, -1.274, 2.327, 1.887, -0.215, 0.855, 0.786, 0.518, -1.888, 0.714] \times 10^{-4}$. The corresponding objective value was $J_{GA} = 0.385$.

Figure 5 shows that the simulation process converged after 152 iterations.

Results of QGA. The optimized parameters were $\mathbf{b}_{QGA} = [-1.9319, -1.7996, -2.0789, 1.6905, 0.46052, -2.1949, 1.3569, -0.42342, 0.98057, -0.28985, 1.877, -0.4173] \times 10^{-4}$. The corresponding objective value was $J_{QGA} = -2.2 \times 10^{-4}$.

Figure 6 shows that the simulation converged after 80 iterations. Compared to the GA, the QGA converged much

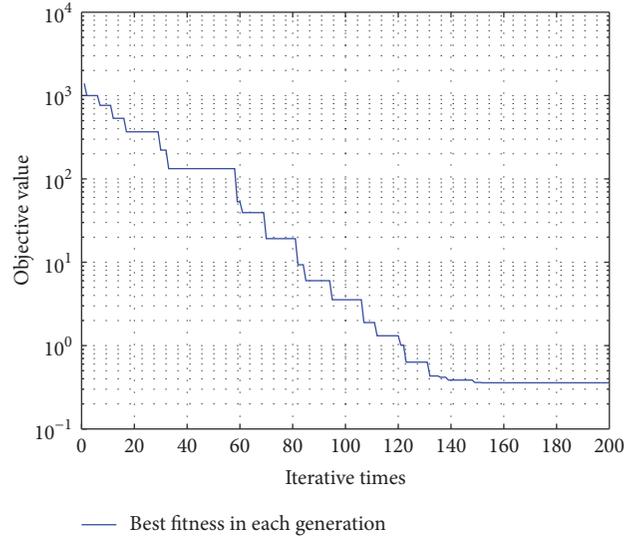


FIGURE 5: Fitness curve of global optimal points for GA.

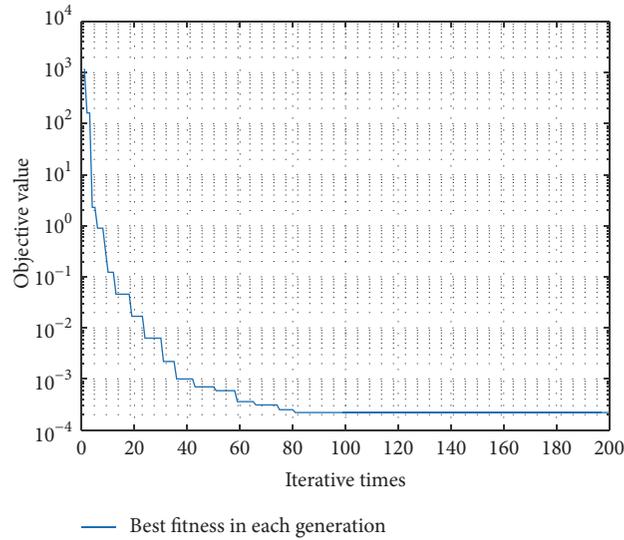


FIGURE 6: Fitness curve of global optimal points of QGA.

more rapidly with higher precision. Moreover, step width of ladder-shaped curve shown in Figure 5 is larger than that shown in Figure 6. This implies that the genetic algorithm can easily fall into a local optimum. Hence, the QGA is superior to the genetic algorithm.

We then solved the optimized joint trajectory according to the parameters obtained from the QGA, as shown in Figure 7. Finally, we used the joint trajectories in Figure 7 as inputs to the motion of the control joint and observed whether the base and the end-effector arrived at the desired state simultaneously. Figure 8 shows the RPY angle curves of the floating base. Positional curves of the floating base and the end-effector are, respectively, described in Figures 9 and 10. Figure 11 shows the distance between the actual and desired positions of the end-effector. The deviation as 2.78×10^{-2} ,

which was acceptable. From Figure 7, we see that the curves are smooth, and this is satisfactory for the control of the joints.

6. Conclusion

In this study, we built a kinematical model for a SBMS and analyzed its nonholonomic properties. Moreover, we also parameterized the joint trajectory and the design objective function and hence transformed the path planning problem into a nonlinear objective optimization problem. We used the genetic algorithm and the QGA to solve above problem. The results showed that the proposed method can obtain a superior solution compared to prevalent systems and converges much more rapidly.

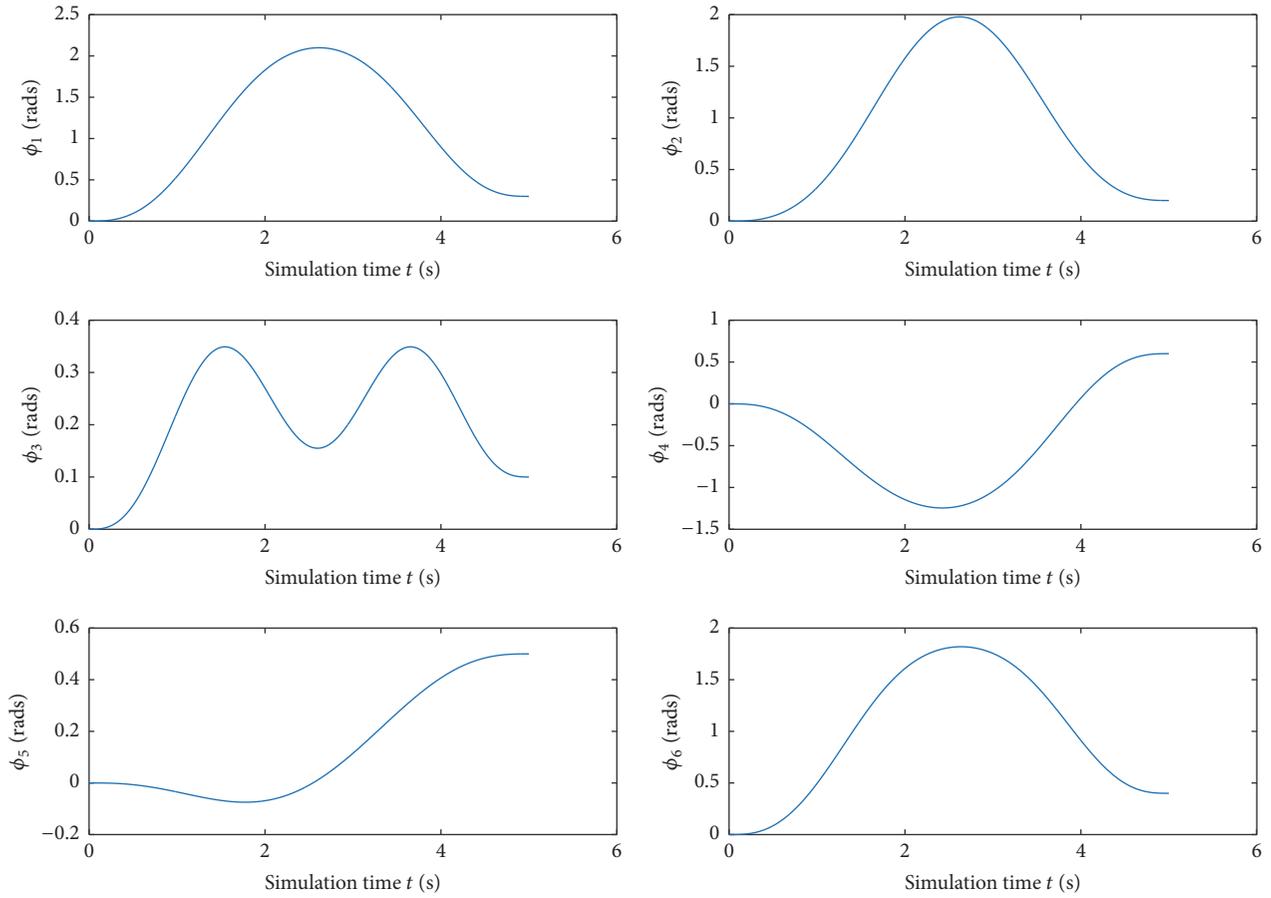


FIGURE 7: Optimal joint curves.

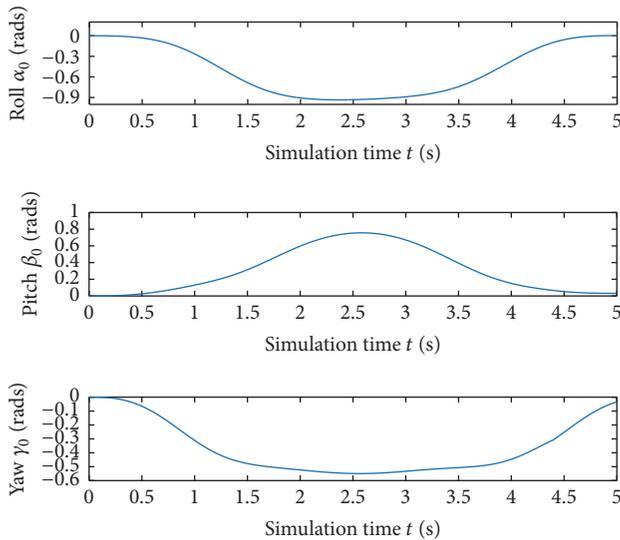


FIGURE 8: Pose error curves of the floating base.

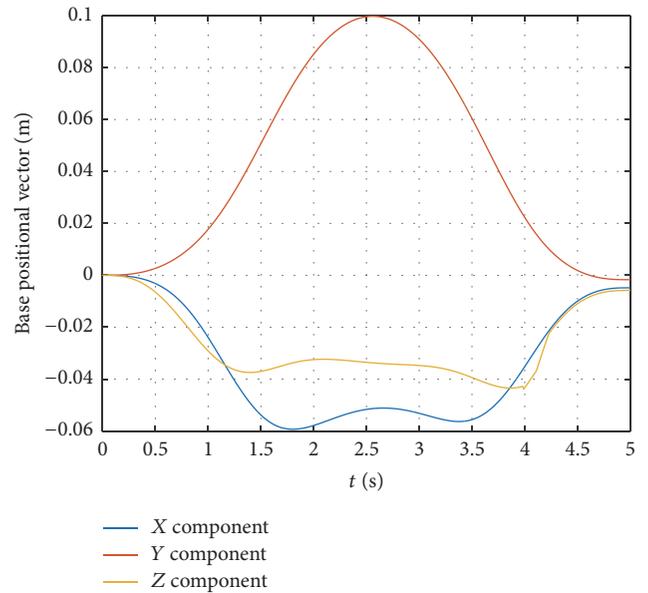


FIGURE 9: Positional curves of the floating base.

We also obtained joint trajectories based on the optimized parameters and drew pose curves of the base and the position curve of the end-effector. The results indicated that the proposed method can arrive at the desired state.

Our method can efficiently reduce the energy consumption and time in a coordinated control process between a floating base and a manipulator.

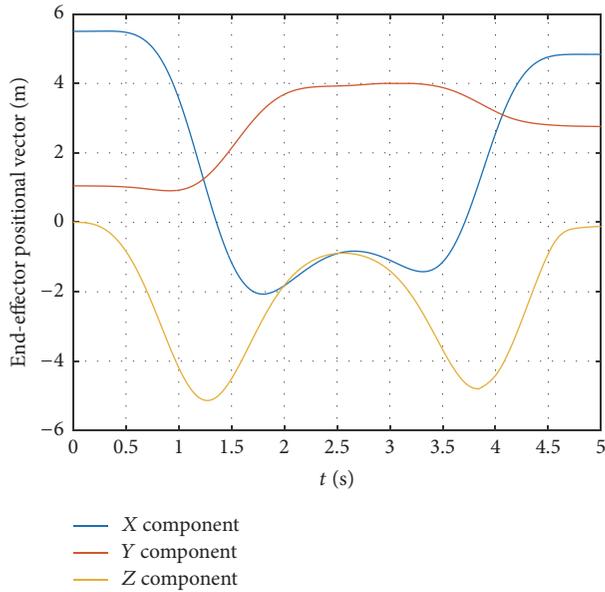


FIGURE 10: Positional curves of the end-effector in inertial reference frame.

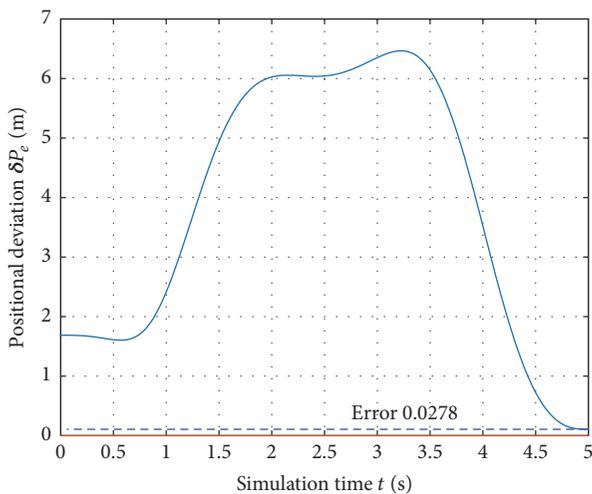


FIGURE 11: Position error curve of end feature point.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

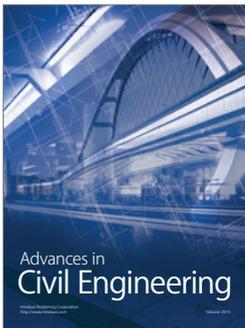
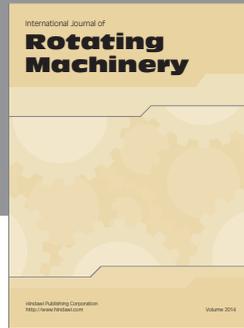
Acknowledgments

Support for this work was provided by the National Natural Science Foundation of China (Grant no. 51505470) and the Dr. Start-Up Fund in Liaoning Province (20141152).

References

- [1] Y. Wang, L. Sun, W. Yan, and J. Liu, "An analytic and optimal inverse kinematic solution for a 7-DOF space manipulator," *Jiqiren/Robot*, vol. 36, no. 5, pp. 592–599, 2014.
- [2] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.
- [3] Y. Liu, Q. Jia, G. Chen, and H. Sun, "Load maximization trajectory optimization for free-floating space robot using multi-objective particle swarm optimization algorithm," *Robot*, vol. 36, no. 4, pp. 402–410, 2014.
- [4] A. M. Bloch, J. E. Marsden, and D. V. Zenkov, "Nonholonomic dynamics," *Notices of the American Mathematical Society*, vol. 52, no. 3, pp. 324–329, 2005.
- [5] H. Li and R. Li, "Zero-disturbance control of free-floating space manipulators using integral-type sliding mode control," *Mathematical Problems in Engineering*, vol. 2015, Article ID 308305, 7 pages, 2015.
- [6] R. Rathee and P. M. Pathak, "Dual arm free flying space robot trajectory planning using polynomial," *Journal of Robotics*, vol. 2015, Article ID 324831, 11 pages, 2015.
- [7] C. Zeng, Q. Song, and Z. Li, "Obstacle avoidance path planning of free-floating space manipulator for on-orbit observation," in *Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering*, p. 3, Kitakyushu, Japan, July 2016.
- [8] E. G. Kaigom, T. J. Jung, and J. Roßmann, "Optimal motion planning of a space robot with base disturbance minimization," in *Proceedings of the 11th Symposium on Advanced Space Technologies in Robotics and Automation*, pp. 1–6, April 2011.
- [9] X. Liu, H. Baoyin, and X. Ma, "Optimal path planning of redundant free-floating revolute-jointed space manipulators with seven links," *Multibody System Dynamics*, vol. 29, no. 1, pp. 41–56, 2013.
- [10] D. N. Nenchev, "Reaction null space of a multibody system with applications in robotics," *Mechanical Sciences*, vol. 4, no. 1, pp. 97–112, 2013.
- [11] J. Hanxu and Z. Shuangqi, "Path planning for space manipulator to avoid obstacle based on A* algorithm," *Journal of Mechanical Engineering*, vol. 13, p. 17, 2010.
- [12] Z. Shi, Y.-Z. Wang, and Q.-L. Hu, "A polynomial interpolation based particle swarm optimization algorithm for trajectory planning of free-floating space robot," *Journal of Astronautics*, vol. 32, no. 7, pp. 1516–1521, 2011.
- [13] F. Zhang, Y. Fu, and S. Wang, "A path planning method for free-method space robot in Cartesian space," *Robot/Jiqiren*, vol. 31, p. 6, 2009.
- [14] Y. Shi, B. Liang, X. Wang, and W. Xu, "Cartesian non-holonomic path planning of space robot based on quantum-behaved particle swarm optimization algorithm," *Journal of Mechanical Engineering*, vol. 47, no. 23, pp. 65–73, 2011.
- [15] Q.-X. Jia, M. Chu, H.-X. Sun, and L. Hong, "Research on the optimal algorithm for trajectory planning of a 9-DOF hyper-redundant robot," *Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications*, vol. 31, no. 2, pp. 20–25, 2008.
- [16] P. C. Hughes, *Spacecraft Attitude Dynamics*, Courier, 2012.
- [17] Z. Li and J. F. Canny, *Nonholonomic Motion Planning*, vol. 192, Springer, 2012.
- [18] C.-Y. Liang, H. Bai, M.-J. Cai, and W.-X. Lu, "Advances in quantum genetic algorithm," *Jisuanji Yingyong Yanjiu*, vol. 29, pp. 2401–2405, 2012.

- [19] J. Bagley, "The behavior of adaptative systems wich emply genetic and correlation algorithms. Doctoral diseertation, University of Michigan," *Dissertation Abstracts International*, vol. 29, no. 12, 1967.
- [20] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, November 1994.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

