Global Minimum-Jerk Trajectory Planning of Space Manipulator

Panfeng Huang, Yangsheng Xu, and Bin Liang

Abstract: A novel approach based on genetic algorithms (GA) is developed to find a global minimum-jerk trajectory of a space robotic manipulator in joint space. The jerk, the third derivative of position of desired joint trajectory, adversely affects the efficiency of the control algorithms and stabilization of whole space robot system and therefore should be minimized. On the other hand, the importance of minimizing the jerk is to reduce the vibrations of manipulator. In this formulation, a global genetic-approach determines the trajectory by minimizing the maximum jerk in joint space. The planning procedure is performed with respect to all constraints, such as joint angle constraints, joint velocity constraints, joint angular acceleration and torque constraints, and so on. We use an genetic algorithm to search the optimal joint inter-knot parameters in order to realize the minimum jerk. These joint inter-knot parameters mainly include joint angle and joint angular velocities. The simulation result shows that GA-based minimum-jerk trajectory planning method has satisfactory performance and real significance in engineering.

Keywords: Genetic algorithms, minimum jerk, space manipulator, trajectory planning.

1. INTRODUCTION

Space robots are playing more and more important roles in current space operation. As well known, the space robots are highly nonlinear, coupled multi-bodies system with nonlinear constraints. Moreover, the dynamics coupling between the manipulator and the base (spacecraft or satellite) will affect the performance of manipulator [1]. For these reasons, it is very difficult to realize the optimal control and provided methods result in impractically complicated algorithms. A simpler approach is to plan a robot's trajectory along which an optimal result can be obtained, then control the robot to track this trajectory. Therefore, the optimum control can be realized by following two steps: optimum trajectory planning for off-line processing; followed by on-line trajectory track. In this paper, we will focus on optimizing motion trajectory of space manipulator according to the defined optimum index to realize minimum-jerk trajectory planning.

The jerk, the third derivative of position of desired joint trajectory, adversely affects the efficiency of the control algorithms and stabilization of whole space robot system. Because of the dynamic coupling between the robotic manipulator and the base, the jerk of robotic manipulator will affect the stabilization of the base, especially, when the space robot is in free-floating situation. If the travelling time of robotic manipulator is determined by the operative mission, an interesting problem is to how to realize the smooth motion of robotic manipulator, that is to plan minimum-jerk trajectory. The key problem of minimizing the jerk in trajectory planning is that the joint position errors decrease when the jerk decreases which was attested by Kyriakopoulos and Saridis [2]. On the other hand, minimum jerk trajectory planning is desirable to expand the robot life-span. Therefore, we propose to find a global optimum method for minimum jerk trajectory planning.

There are many studies on trajectory planning or path planning of space robots. Agrawal and Xu [3] addressed the global optimum path planning for redundant space manipulator. They considered the linear and angular momentum as constraint conditions, then using Lagrange multiplier technique to change the optimum problems subject to constraint conditions to non-constraint problems. Then, using differential and algebraic equations to solve the objective functions. Dubowsky and Torres [4] proposed a method called the Enhanced Disturbance Map (EDM) to plan the space manipulator motions so that the

Manuscript received July 14, 2005; revised March 31, 2006; accepted April 26, 2006. Recommended by Editorial Board member Sangdeok Park under the direction of Editor Jae-Bok Song. This work is partially supported by Chinese High-Tech Program in Aerospace.

Panfeng Huang is with the College of Astronautics, Northwestern Polytechnical University, China (e-mail: pfhuang@nwpu.edu.cn).

Yangsheng Xu is with the Department of Automation and Computer-Aided Engineering, the Chinese University of Hong Kong, Hong Kong, China (e-mail: ysxu@acae.cuhk.edu.hk).

Bin Liang is with the Shenzhen Space Technology Center, Harbin Institute of Technology, Shenzhen, China (e-mail: bliang@robotsat.com).

disturbance to the space base is relatively minimized. Their technique aims at understanding the complicated problem and developing the algorithm to reduce the disturbance. Papadopoulos [5] exhibited the nonholonomic behavior of free-floating space manipulator by planning the path of space manipulator. They proposed a path planning method in Cartesian space to avoid dynamically singular configurations. Yoshida and Hashizume [6] utilized the ETS-VII as an example to address a new concept called Zero Reaction Maneuver (ZRM) that is proven particularly useful to remove the velocity limit of manipulation due to the reaction constraint and the time loss due to wait for the attitude recovery. Moreover, they found that the existence of ZRM is very limited for a 6 DOF manipulator, besides the kinematically redundant arm. However, the redundant manipulator gives rise to the computational cost and complicated kinematics problems. Hence, it is not best way to use the redundant manipulator from the view of engineering point. Fernandes, Gurvits, and Li [7] presented a method for near optimum attitude control of space manipulator using internal motion. This formulation considered a two point boundary value problem but it did not consider the problem of the end-effector following a path. As a result, the holonomic and nonholonomic momentum conservation constraints lie in the null space of the Jacobian Matrix. This may not be true if the end-effector must follow certain trajectory.

According to the previous researches, there are a few researches on the optimum trajectory planning according to a detailed optimum index. In the meanwhile, these topics are interesting and important to realize the optimum control and smooth motion of space robot system. Therefore, it is necessary to investigate the minimum-jerk based trajectory planning of space robot. In general, the optimum trajectory planning mainly includes optimum trajectory planning in Cartesian space (OTPCS) and optimum trajectory planning in Joint space (OTPJS). OTPCS is to plan the optimum trajectory along the assigned geometric path. The main objective of OTPCS is to avoid the obstacle successfully. For our research topic, OTPCS is not necessary here. However, OTPJS is key problem. In comparison to OTPCS, OTPJS is more difficult than OTPCS because the optimum trajectory should be found in joint space. Moreover, OTPJS provides an opportunity to fully use the capability of a manipulator. Especially, when the space robotic manipulator has some inherent physical constraints and environmental constraints, it is an important problem to optimize the motion trajectory of robotic manipulator. In this paper, we present a new approach based on a genetic algorithm to deal with OTPJS problem in a more general and practical style.

Genetic Algorithms (GA) is population-based, stochastic, and global search methods. Their performance is better than that of some classical techniques and they have been successfully used in the path planning of industrial robotic manipulator [8]. An optimal solution is quite difficult to achieve using traditional methods for multi-parameters system. However, GA has these search abilities that can provide the possibility to find optimal solutions. In this paper, to obtain the minimum jerk trajectory, we formalize the proposed trajectory as a global constrained minimax optimization problem using GA. According to the trajectory planning strategies introduced in the following section, we divide whole trajectory into several trajectory segments, the path point connecting with two segments is called knot point. Our proposed method is to search optimum parameters of each knot point, such as, joint angle and joint angular velocity, then to realize minimum jerk trajectory planning. We will use the GA to search the optimal parameters of each knot point globally.

This paper is organized as follows. In Section 2, we firstly describe the fundamental knowledge about modelling and trajectory planning strategies of space robot system, then address the problem about minimum jerk trajectory planning. Some basic concepts and kernels of genetic algorithms are simply introduced in Section 3. In Section 4, a new algorithm to solve the optimal trajectory planning problem based on the GA is proposed. In Section 5, the simulation study and result are shown to illustrate the effectiveness of our proposed method. Final Section summarizes the whole paper and give some conclusions.

2. MODELLING AND TRAJECTORY PLANNING

2.1. Modelling and equations of motion

Consider a free-flying space robot shown in Fig. 1, composed of robotic manipulator connected by revolution joints and a spacecraft, the manipulator mounted on the spacecraft. As shown in Fig. 1, we define two coordinate systems, one is the inertial coordinate system in the orbit, the other is the spacecraft coordinate system Σ_0 attached on the spacecraft body with its origin at the centroid of the spacecraft. The COM is the center of total system mass, all vectors in this paper are expressed in terms of coordinate Σ_I. For whole space robot system, the external force or torque on the space base F_b, which can be generated by jet thrusters or reaction wheels, and F_e can be assumed zero before the end-effector contacts the objective. Therefore the linear and angular momenta of whole system are conservative when $F_b = 0$, the motion of system is governed by only inertial force/torque of the manipulator joint τ.

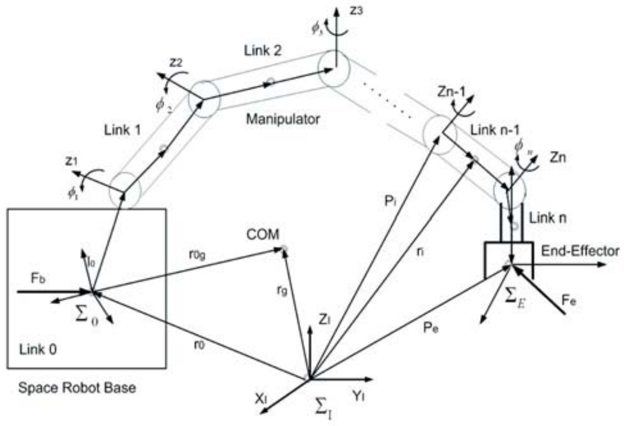

Fig. 1. Dynamics model of space robot system.

Thus, it is easy to derive the kinematics and dynamics equations of motion as follows.

The kinematic equation of a space robot in velocity level is to represent the relationship between the end-effector and joint velocity of manipulator. The equation is expressed as follows [9]:

$$x_e = J_m\dot{\phi} + J_b\dot{x}_b\,. \tag{1}$$

The dynamics equation of the space robot system is expressed in the following form [9,10]:

$$\begin{bmatrix} H_b & H_{bm} \\ H_{bm}^T & H_m \end{bmatrix}\begin{bmatrix} \ddot{x}_b \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} c_b \\ c_m \end{bmatrix} = \begin{bmatrix} 0 \\ \tau \end{bmatrix}. \tag{2}$$

The detailed definition and symbols in all equations and Fig. 1 mentioned above can be obtained from papers [10]. According to the above assumption, we can obtain the following momentum equation from (2).

$$H_b\dot{x}_b + H_{bm}\dot{\phi} = \begin{bmatrix} L \\ P \end{bmatrix} = const \tag{3}$$

Assume $\begin{bmatrix} L \\ P \end{bmatrix}$ = 0 for simplification, thus, from (3), we obtain

$$\dot{x}_b = -H_b^{-1}H_{bm}\dot{\phi}\,. \tag{4}$$

Substituting (4) into (1), we get

$$\dot{x}_e = J_g\dot{\phi} \;\text{ or }\; \dot{\phi} = J_g^{-1}\dot{x}_e\,, \tag{5}$$

where $J_g = J_m - J_bH_b^{-1}H_{bm}$, the matrix $\boldsymbol{J_g}$ is called Generalized Jacobian Matrix (GJM) or Space Jacobian Matrix (SJG). GJM is used to calculate the joint angular velocity and end-effector velocity. Moreover, it is also used to check whether the space manipulator system Σ_I causes the dynamics singularities. When the determinant of GJM is equal to zero or the GJM loses full rank, the manipulator appears the dynamics singularities. In addition, the GJM can be used to design controller usually.

2.2. Trajectory planning method

In general, we use the high order polynomial function to generate the trajectory of robotic manipulator. However, the path must be smooth and continuous in order to ensure the motion stabilization of manipulator, that is, the first and second differential of the polynomial with respect to time must be smooth curves according to continuity constraints. The number of inter-knots and the positions of them should be defined before trajectory planning. In this paper, the number of inter-knots is defined manually, and the parameters of inter-knots are optimized using GA.

Here, we define the point-to-point trajectory planning as a simple theme to address our optimum problem. The trajectory refers to a time history of position, velocity, and acceleration for each joint. Suppose that the point-to-point trajectory is connected by several segments with continuous acceleration at each inter-knot. The position of each inter-knot is supposed to be unknown in the following section of the paper. Of course, the inter-knot can also be given as particular points that should be passed through. This is useful especially when there is an obstacle in the working area.

Given an open chain space manipulator with n degree of freedom (DOF) revolute joints. Let q_i, i=1,…,n, denote the joint variables. Thus, the robotic manipulator trajectories consist of a finite sequence of position, velocity and acceleration of each joint. According to the motion equation of robotic manipulator, we can calculate the sequence of torque values. To obtain the minimum jerk based motion trajectory, the problem is to search some optimal inter-knots where the position and velocity of joint satisfy the constraint conditions. Therefore, the problem is changed to optimize the multiple parameters of joint trajectories.

In order to satisfy the initial and final conditions as well as continuity constraints. we use $\overbrace{4-4,\ldots,4-5}^{m}$ iterative trajectory planning strategy [11], in which m quad polynomials and one fifth order polynomial are used to inter-knot $m+2$ points. Let's define that there are m_p inter-knots between the initial and the final points. Thus, the initial point and m_p inter-knot, a quad polynomial is used to describe these segments as follows:

$$Q_{ij}(t) = a_{ij0} + a_{ij1}t + a_{ij2}t^2 + a_{ij3}t^3 + a_{ij4}t^4, \quad i = 0,\ldots,m-1,\; j = 1,\ldots,n. \tag{6}$$

According to the position trajectory of joint, Q_{ij} and definition of jerk, we can obtain the jerk function with respect to time.

$$Jerk_{ij} = Q_{ij}^{3}(t) = 6a_{ij3} + 24a_{ij4}t, \tag{7}$$

where $Q_{ij}^{3}(t)$ represents the third derivative of $Q_{ij}(t)$. The coefficients are given as follows

$$a_{ij0} = q_{ij}, \tag{8}$$

$$a_{ij1} = v_{ij}T_{ij}, \tag{9}$$

$$a_{ij2} = \frac{1}{2}a_{ij}T_{ij}^{2}, \tag{10}$$

$$a_{ij3} = 4(q_{(i+1,j)} - q_{ij}) - v_{(i+1,j)}T_{ij} - 3v_{ij}T_{ij} - a_{ij}T_{ij}^{2}, \tag{11}$$

$$a_{ij4} = v_{(i+1,j)} - 3(q_{(i+1,j)} - q_{ij}) - 2v_{ij}T_{ij} + \frac{1}{2}a_{ij}T_{ij}^{2}. \tag{12}$$

The acceleration $a_{(i+1,j)}$ in each inter-knot can be obtained as:

$$a_{(i+1,j)} = \frac{12a_{ij4} + 6a_{ij3} + 2a_{ij2}}{T_{ij}^{2}}. \tag{13}$$

The segment between the m inter-knot and final point can be described by a five order polynomial as follows:

$$Q_{mj}(t) = b_{mj0} + b_{mj1}t + b_{mj2}t^{2} + b_{mj3}t^{3} + b_{mj4}t^{4} + b_{mj5}t^{5}, \quad j = 1,\ldots,n. \tag{14}$$

Thus the jerk with respect to time in this segment can be obtained:

$$Jerk_{mj} = Q_{mj}^{(3)}(t) = 6b_{mj3} + 24b_{mj4}t + 60b_{mj5}t^{2}, \tag{15}$$

where the coefficients b_{mjk} in (14) are derived as follows:

$$b_{mj0} = q_{mj}, \tag{16}$$

$$b_{mj1} = v_{mj}T_{mj}, \tag{17}$$

$$b_{mj2} = \frac{1}{2}a_{mj}T_{ij}^{2}, \tag{18}$$

$$b_{mj3} = 10c_{1} - 4c_{2} + \frac{1}{2}c_{3}, \tag{19}$$

$$b_{mj4} = -15c_{1} + 7c_{2} - c_{3}, \tag{20}$$

$$b_{mj5} = 6c_{1} - 3c_{2} + \frac{1}{2}c_{3}, \tag{21}$$

where

$$c_{1} = q_{mj} - b_{ij2} - b_{ij1} - b_{ij0}, \tag{22}$$

$$c_{2} = v_{mj}T_{ij} - 2b_{ij2} - b_{ij1}, \tag{23}$$

$$c_{3} = a_{mj}T_{ij}^{2} - 2b_{ij2}. \tag{24}$$

In quad and five order polynomial equations, we normalize time t as $t = \frac{\tau - \tau_{i-1}}{\tau_i - \tau_{i-1}} = \frac{\tau - \tau_{i-1}}{T_i}$. Thus, normalized time variable $t \in [0,1]$; τ is the real time in seconds, τ_i is real time at the end of *i*th trajectory segment. T_i is the real time required to travel through the *i*th segment $T_i = \tau_i - \tau_{i-1}$.

As formulated above, the total parameters to be determined are the joint angles, angular velocities of all inter-knots ($2n \times m$). All these parameters can be determined and optimized using the GA.

2.3. Minimum-jerk trajectory planning

The trajectory planning problem is generally defined here as the point to point problem, i.e., that of determining the time history of the robot joints and spacecraft state (position and orientation) in order to move the end-effector of the robot from a given initial state to a final state in inertial space. However, such planning trajectory only ensures that the end-effector of robot move to the desired state. Whereas, this trajectory must be optimized in order to satisfy kinematic and dynamic constraints and reduce the vibrations. Especially, for space robot system, optimizing the motion trajectory becomes a more and more important problem in order to minimize to the disturbance because of the vibration of robotic manipulator. Moreover, the disturbance is with respect to the vibrations of manipulator. Therefore, minimum jerk trajectory planning is a significant research topic.

The global minimum-jerk trajectory planning problem may be stated as follows:

Minimize

$$\Gamma = \sum_{i=0}^{m} \max \left\| Jerk_{ij}(q,\dot{q},t) \right\|, i = 1,\ldots,m, j = 1,\ldots,n, \tag{25}$$

where $Jerk_{ij}$ represents the jerk of *j*th joint at *i*th inter-knot point, which can be computed by calculating three derivative of position of desired trajectory. Moreover, the class of trajectories in *n* dimensional joint space, Q_t. Q_t is a high order polynomial to represent the trajectory of joint. The trajectories satisfy the constraints as follows:

(1) Initial and final condition:

$$\begin{aligned} &Q_0 = q_0, \ \dot{Q}_0 = v_0, \ \ddot{Q}_0 = a_0, \\ &Q_{t_f} = q_{t_f}, \ \dot{Q}_{t_f} = v_{t_f}, \ \ddot{Q}_{t_f} = a_{t_f}. \end{aligned} \tag{26}$$

(2) Kinematics constraints:

$$\begin{aligned} &q_{\min}^{i} \le Q(t)^{i} \le q_{\max}^{i}, \\ &v_{\min}^{i} \le \dot{Q}(t)^{i} \le v_{\max}^{i}, \\ &a_{\min}^{i} \le \ddot{Q}(t)^{i} \le a_{\max}^{i}, \\ &i = 1,\ldots,n,\ t \in [0, t_f]. \end{aligned} \tag{27}$$

(3) Dynamics constraints:

$$\begin{aligned} &\tau_{\min}^{i} \le \tau^{i} \le \tau_{\max}^{i},\ i = 1,\ldots,n, \\ &f_{b\min} \le f_b(t) \le f_{b\max}, \end{aligned} \tag{28}$$

where $\tau_{\min}^{i}$ and $\tau_{\max}^{i}$ are the lower and upper computed torque of *i*th joint, and $f_{b\min}$ and $f_{b\max}$ are lower and upper permitted disturbance force and momenta to the spacecraft, respectively. $f_b(t)$ represents the computed dynamics disturbance force and moment to the spacecraft.

(4) Smooth motion constraints:

The key problem of optimal trajectory planning is to search the optimal parameters of inter-knots. Therefore, the position, velocity and acceleration of joint at any inter-knot should satisfy the following continuity constraints in order to keep the smooth motion of manipulator.

$$\begin{cases} Q_i(t_i) = q_i \\ Q_{i+1}(t_i) = q_i \\ \dot{Q}_i(t_i) = \dot{Q}_{i+1}(t_i) \\ \ddot{Q}_i(t_i) = \ddot{Q}_{i+1}(t_i) \end{cases} \tag{29}$$

3. GENETIC ALGORITHMS

3.1. A brief review of genetic algorithm

Genetic algorithm (GA) is a kind of heuristic global searching algorithm based on the mechanics of natural selection and natural genetics by Holland and his students, Goldberg [12]. The global search characteristics of GA can be found from reference [13], we will not verify it here. The global solutions can be found for both linear and nonlinear formulations. A genetic algorithm has the following basic characteristics: (a) it works in representational space after encoding the parameters by alphabets or real values; (b) it searches from a population of points instead of a single point; (c) The optimal solution searching process is independent of the form of the objective function; (d) it uses probabilistic operation rules. Based on motivations mentioned above, we use GA as our global optimal algorithm.

Generally speaking, a GA starts to evolve by generating (usually in a random way) an initial population of chromosomes. Then, the value of a function called fitness function is evaluated for each chromosome of the population. After this, a set of genetic operators (selection, crossover and mutation) are used in succession to create a new population of chromosomes for the next generation. The process of evaluation and creation of new successive generations is repeated until the satisfaction of a convenient termination condition. The basic optimal processes of GA can mainly be illustrated in Fig. 2. In the following section, we will introduce the kernels of GA.

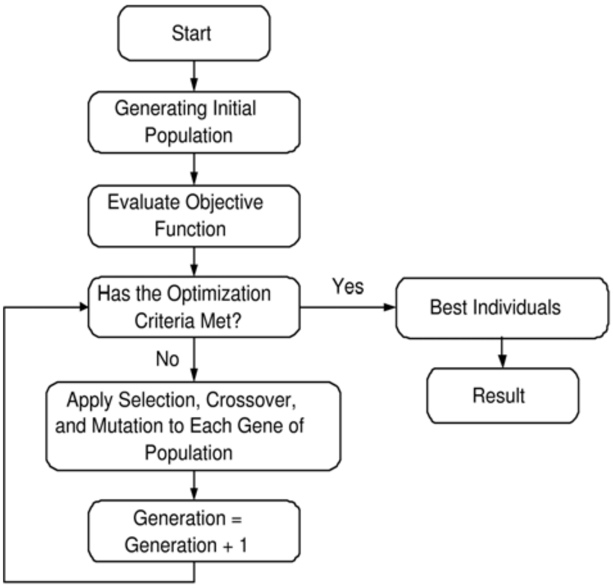
Start
Generating Initial Population
Evaluate Objective Function
Has the Optimization Criteria Met?
Yes
No
Best Individuals
Result
Apply Selection, Crossover, and Mutation to Each Gene of Population
Generation = Generation + 1

Fig. 2. Optimization procedure using genetic algorithm.

3.2. Genetic operators and control parameters

The kernels of GA are its genetic operator and control parameters form Fig. 2. The basic operators of GA mainly include selection, crossover, mutation, and inversion, we will simply introduce these operators in the following section.

Selection: The individual chromosomes are selected based on the binary tournament selection strategy. According to this strategy, two chromosomes are picked at random from the population and that with the higher fitness value is copied into a mating pool (i.e., it survives and reproduces its structure into the new population). This process is repeated until the mating pool is full.

Crossover: A uniform crossover operator was used. This operator works as follows: two parent chromosomes are selected based on the crossover probability. For the pair of the selected parents a template or mask chromosome is randomly generated. The bit-value at each position of the template specifies the bit-value of the corresponding position of the child chromosome. Specifically, where there is “1” in the template, the corresponding gene from the first parent passes its value to the child, otherwise the second parent passes its bit-value to the child. The process is repeated with the parents exchanged to produce the second child. Therefore, offspring contain a mixture of genes from each parent.

Mutation: Mutation is a random alteration of the value of a string position. In binary coding, this means changing a 1 to 0 and vice versa. In GA, its probability of occurrence is generally kept small, as a higher occurrence rate would lead to a loss of important data. GA, with 100% mutation rate becomes random search in the solution space.

Inversion: Inversion is a reordering operator applied on the bits of a single chromosome. It works by reversing the order of genes between two randomly chosen positions within the selected chromosome.

The last critical aspect in designing a GA is the selection of the suitable settings for the GA's control parameters. Unfortunately, there is no formal way to define the appropriate parameters' settings. Traditionally, this is achieved experimentally. A description of each one of the control parameters is given below:

Population size: Determines how many chromosomes, i.e., how much genetic material, are available during the genetic search. A too small population size decreases the ability of the GA to adequately cover the search space. A too large population size significantly increases the time needed by the GA to evaluate the chromosomes and thus results in an ineffective search.

Crossover rate: Specifies the frequency with which the crossover operator is applied to the individual's chromosomes in a new generation. A too low crossover rate causes the introduction of fewer new individual's into the population and may lead to search stagnation since the process of reproduction tends to dominate. A too high rate leads to a very fast exploration of the search space but the GA performance may be degraded as strong individuals are discarded very fast before reproducing their structure.

Mutation rate: Specifies the probability that a gene's value of a newly created chromosome will be changed. Mutation governs the introduction of new unexplored areas in the search. A high mutation rate increases the diversity in the population but introduces excessive randomness in the search. Conversely, a too low mutation rate reduces the diversity and leads to sub-optimal solution.

Generation gap: This parameter specifies the proportion of the individuals in the population which are replaced by the offspring in each generation. Usually, a generation gap of one is use, i.e., the whole population is replaced in each generation.

GA is inevitably slower than calculus based methods in the problem domain where they can be used. However, all calculus based methods are lack of either robustness or global optimum, so that GA is useful for highly complicated, nonlinear and discontinuous problems as well as combination problem.

4. MINIMUM JERK ALGORITHMS USING GA

In this section, we use the genetic algorithms to solve the optimal trajectory planning described by the cost function (25) with dynamics model under initial and final conditions and constraints. It is important to select the appropriate trajectory planning strategy for optimum trajectory planning in joint space. Firstly, some inter-knots are selected to generate the trajectory segments. In each trajectory segment, a suitable polynomial will be used to obtain the local trajectory. The inter-knot parameters, such as, position and velocity of each trajectory segment will be code and optimized using GA. Thus, the whole trajectory will be planned iteratively and its performance can be calculated according to the fitness function in which cost function and constraints are taken into account. The parameters of inter-knots of each trajectory segment are optimized by the GA. Meantime, the path are evolved into optimum trajectory connecting initial and final points in the joint space.

From above section, we obtain that the coefficients of polynomials depend on trajectory parameters, q_{ij}, v_{ij}, and a_{ij} if the travel time t_f of every trajectory segment is fixed. Therefore, these parameters will be evolved by genetic algorithm. They should be encoded because GA works in the representation space. Here, we use binary numbers to code the parameters.

The target of optimizing trajectory planning problem in our research is to obtain minimum jerk with satisfactory joint position, velocity, acceleration, torque, and continuity constraints. Therefore, we can obtain new algorithm procedure to optimize trajectory using genetic algorithm as follows:

Step 1: Choose the number of inter-knots m, then determine trajectory planning strategies for each trajectory segment according to description method in Section 2.2. Encode the parameters of every inter-knot using the chromosomes of GA.

Step 2: Define fitness function according to cost function and constraints. Define GA parameters, such as population size n_p, generation number n_g, crossover probability p_c and effective gene number n_e. Let generation number n_g=1.

Step 3: Randomly generate a population of binary strings, $pop^k = bs_l^k \mid 1 = 1,...,n_p$.

Step 4: Let l=1.

Step 5: Decode each binary string bs_l^k into parameter $pop_l^k = \{q_{ij}, v_{ij}, \mid i=1,...,m, j=1,...,n\}$. Using 4-4-,...,-4-5 iterative trajectory planning strategy to plan the trajectory Q_{ij}^k with pop_l^k according to (6)-(24).

Step 6: Compute the maximum joint angle, joint angular velocity, q_{ijmax}, v_{ijmax} in the constraint conditions, such as angle constraints, angular velocity limits, acceleration limits, with respect to the trajectory Q_{ij}^{k}.

Step 7: Compute the absolute values of each joint jerk and choose the maximal jerk at each inter-knot point according to maximal position and velocity of joint, q_{ijmax}, v_{ijmax}.

Step 8: Let $l=l+1$, and go to Step 5, until $l=n_p$.

Step 9: Summarize the maximal values of all jerks from $l=0$ to $l=n_p$ according to fitness function. Collect the minimum fitness function value.

Step 10: Let $k=k+1$, generate a new population pop^k using reproduction, crossover and mutation operators. Go to Step 4, until $k=n_g$

Step 11: Obtain the minimum fitness function value, thus, the parameters in this situation is optimal values. Get the optimum trajectory Q_{ij}^{*}.

5. SIMULATION STUDY

In this section, In order to verify the performance of minimum jerk trajectory planning method using GA mentioned above. Let's consider an example to better understand the optimum algorithms. A model of a planar 2 DOF free-flying space robot is shown in Fig. 3. The parameters of the space robot are shown in Table 1. For a real space robot system, the joint angle, angular velocity, acceleration, and torque of the manipulator should have constraint values, we can define the constraint conditions of the model of space robot as follows.

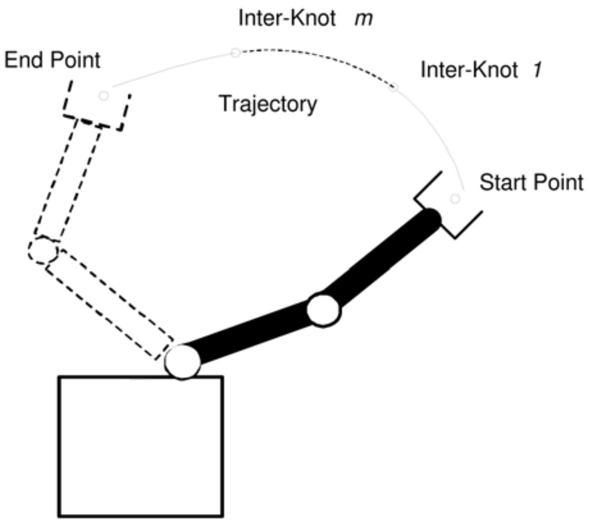

Fig. 3. Model of 2 DOF planar space robot system.

Table 1. Parameters of space robot system.

	Base (Link0)	Space Manipulator	
		Link1	Link2
Mass (Kg)	40	4	3
Length (m)	1	1	1
I (Kg·m^2)	6.67	0.33	0.25

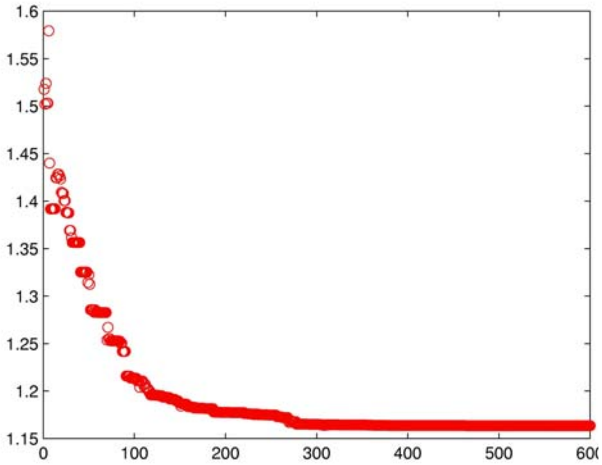

Fig. 4. Average value of objective function (GA optimization) versus number of generation.

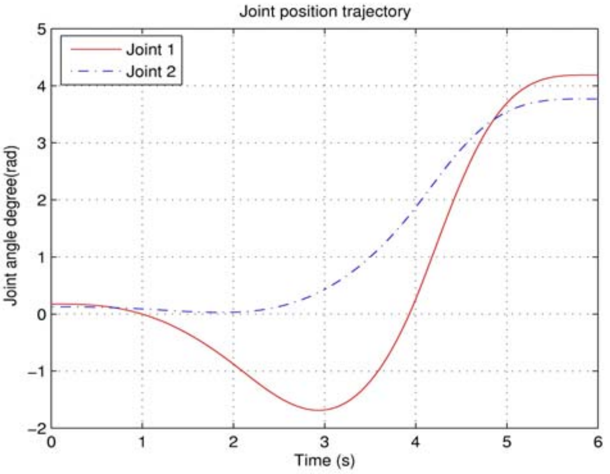

Fig. 5. Joint position trajectory versus time.

In the simulation study, we plan a point to point trajectory in joint space. The manipulator starts from q_{1s} = pi/18, q_{2s} = pi/25 in joint space, and the end point, q_{1e} = 4 × pi/3, q_{2e} = 6 × pi/5. The initial and final velocities, and accelerations are taken to be zero. According to our optimal objective, we need use the GA to search the best inter-knot points in the constraint conditions. These optimized parameters mainly include joint angle, joint angular velocities at each inter-knot point. To simplify the complicated computation, two inter-knots and two second execution time for each segment are chosen. We can use the trajectory planning method mentioned above to plan three segment trajectories. The termination conditions of 600 generation are used.

According to the simulation result, Fig. 4 shows the average value of objective function (GA optimization) versus number of generation. A continuous decrease in average of objective function is also indicative of a smooth convergence to a solution. Fig. 5 shows the joint position trajectory of joint q_1 and q_2. The plot

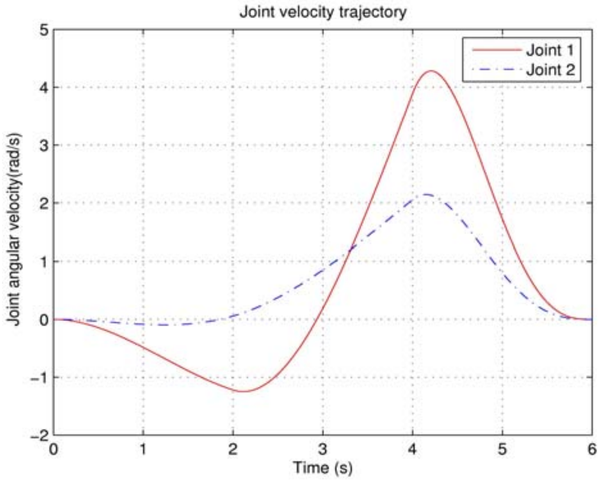

Fig. 6. Joint angular velocity trajectory versus time.

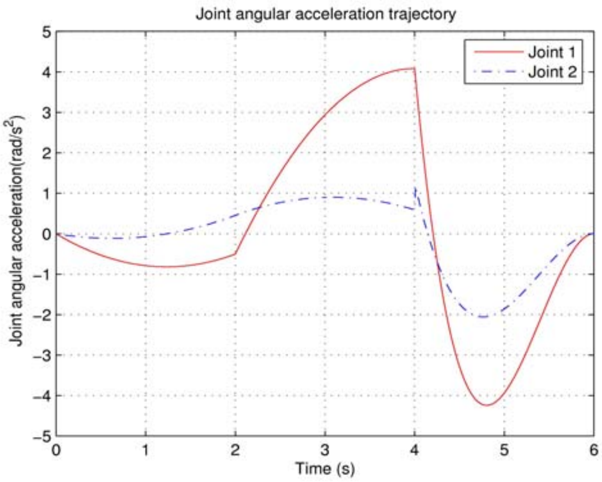

Fig. 7. Joint angular acceleration trajectory versus time.

shows that both q_1 and q_2 start from initial position at t = 0s, and reach the final position at t = 6s. Fig. 6 shows the joint angular velocities trajectory whose values are limited in the constraints of joint angular velocity. Fig. 7 shows the joint angular acceleration after optimization. Moreover, the acceleration values are limited in the constraint conditions. Fig. 8 shows the joint torque when the manipulator tracking the planned trajectory. Fig. 9 shows the joint jerk after optimizing the trajectory. In order to compare the minimized jerk after optimum with the normal jerk no optimized. We have compared the optimal jerk with non optimal jerk, and we find that the maximum jerk optimized is smaller than the non optimal jerk. Thus, the space manipulator can realize minimum-jerk motion from initial point to final point and exert the capability fully.

The goal of simulation is to verify the performance of GA optimization. From the simulation result, the parameters of two inter-knot points can be obtained as follows.

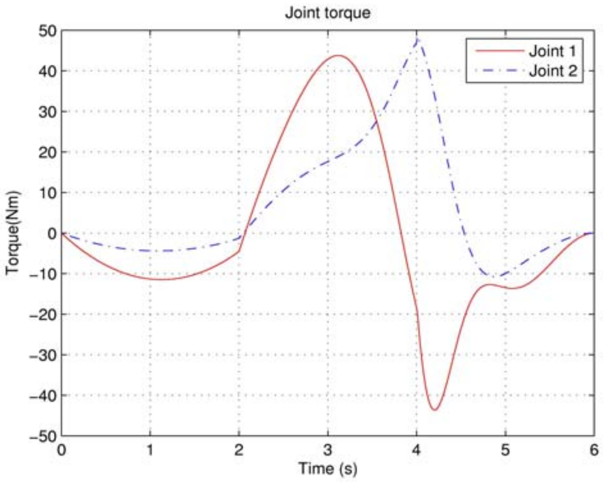

Fig. 8. Joint torque versus time.

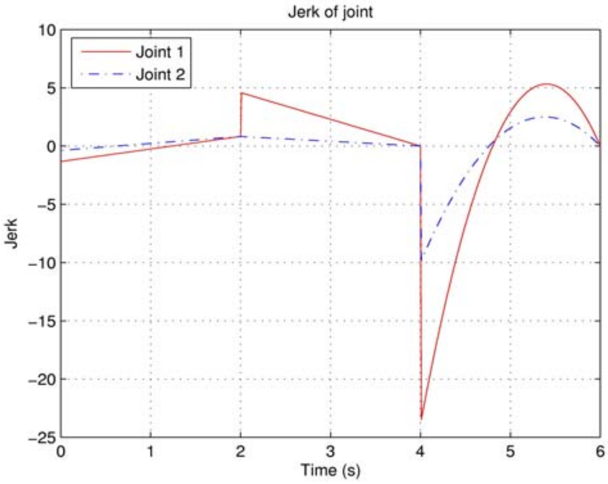

Fig. 9. Joint jerk versus time.

$$\theta_{1mp1} = -0.8779, \quad \theta_{2mp1} = 0.0302,$$
$$\dot{\theta}_{1mp1} = -1.2203, \quad \dot{\theta}_{2mp1} = 0.0557,$$
$$\theta_{1mp2} = 0.2613, \quad \theta_{2mp2} = 1.8693,$$
$$\dot{\theta}_{1mp2} = 3.8885, \quad \dot{\theta}_{2mp2} = 2.0571.$$

Because the number of inter-knots is chosen manually, it is necessary to study how many inter-knots are optimal for optimization. Obviously, the more inter-knots, the more problem complicated, which certainly cost more computation time. Thus, choosing the smallest number of inter-knot is optimal. However, optimal precision may increase when adding the number of inter-knot, which will be verified in the future work.

6. CONCLUSION

In this paper, a new scheme for global minimum-jerk trajectory planning method is developed. The method is based on the genetic algorithms which can globally search most satisfactory parameters of inter-knot to generate the optimal motion trajectory. The optimal trajectory obtained is fitful for high velocity and high precision dynamic control. The performance for the two DOF planar space robot is good and suggest its potential application to real space robot system.

As well known, the natural characteristics of space robot is how to minimize the disturbance to the base due to the motion of manipulator. Therefore, it is an significant research topic to optimize the motion path of manipulator during its operation in order to minimize the disturbance. On the other hand, when the manipulator captures the object, the collision between the end-effector and objects is existent consequentially. So optimizing the approach trajectory of end-effector can reduce or avoid the collision fully. These two problems are importance for satellite service by space robot in the future.

REFERENCES

[1] P. F. Huang, Y. S. Xu, and B. Liang, "Dynamic balance control of multi-arm free-floating space robots," *International Journal of Advanced Robotic Systems*, vol. 2, no. 2, pp. 117-125, 2005.

[2] K. J. Kyriakopoulos and G. N. Saridis, "Minimum-jerk path generation," *Proc. of IEEE International Conference on Robotics and Automation*, Philadelphia, PA, pp. 364-369, 1988.

[3] O. P. Agrawal and Y. Xu, "On the global optimum path planning for redundant space manipulator," *IEEE Trans. on System, Man, and Cybernetics,* vol. 24, no. 9, pp. 1306-1316, September 1994.

[4] S. Dubowsky and M. A. Torres, "Path planning for space manipulator to minimize spacecraft attitude disturbances," *Proc. of IEEE International Conference on Robotics and Automation*, 1991.

[5] E. Papadopouls, "Path planning for space manipulators exhibiting nonholonomic behavior," *Proc. of IEEE International Conference Intelligent Robots and Systems* 1992.

[6] K. Yoshida and K. Hashizume, "Zero reaction maneuver: Flight velification with ETS-VII space robot and extension to kinematically redundant arm," *Proc. of IEEE International Conference on Robotics and Automation*, 2001.

[7] C. Fernandes, L. Gurvits, and Z. X. Li, "Foundation of nonholonmic motion planning," *Technical Report No. 577*, Dept. of Computer Science, New York University, August 1991.

[8] S. D. Sun, A. S. Morris, and A. M. S. Zalzala, "Trajectory generation for redundant manipulator using virus evolutionary genetic algorithm," *Robotica,* vol. 14, pp. 227-234, 1996

[9] Y. Xu and T. Kanade, *Space Robotics: Dynamics and Control*, Kluwer Academic Publishers, November 1992.

[10] P. F. Huang, Y. S. Xu, and B. Liang, "Capturing uncontrolled spinning satellite by a space manipulator," *Proc. of the Second ISATED International Multi-Conference Automation, Control, and Application(ACIT-ACA)*, Novosibirsk, Russia, 2005.

[11] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics*: *Control, Sensing, Vision, and Intelligence*, McGraw-Hill, Inc., 1987.

[12] J. H. Holland, *Adaptation in Natural and Artificial System*, The University of Michigan Press, Ann Arbor, MI, 1975.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

Panfeng Huang received the B.S. and M.S. degrees from Northwestern Polytechnical University in 1998, 2001, respectively, and the Ph.D. degree from the Chinese University of Hong Kong in the area of Automation and Robotics in 2005. He is currently an Associate Professor of College of Astronautics and Vice Director of Research Center for Intelligent Robotics at the Northwestern Polytechnical University. His research interests include modelling, kinematics, dynamics, trajectory planning, and control of space robots.

Yangsheng Xu received the B.S. and M.S. degrees from Zhejiang University, and Ph.D. from the University of Pennsylvania in the area of Robotics. He was a Faculty Member in Robotics Institute, School of Computer Science at Carnegie Mellon University from 1989-1999. He is currently a Professor of Automation and Computer-aided Engineering at the Chinese University of Hong Kong. His research interests are in the areas of robotics, interface, dynamics and control, and micro-spacecraft.

Bin Liang received the Ph.D. degree in Robotics from the Tsinghua University, China. He is currently a Professor at Harbin Institute of Technology and Director of Shenzhen space technology center. He has been working in areas of design and control of robots, and intelligent spacecraft systems.