Subsection Evolution in GA for Trajectory Planning of a Space Manipulator

Cen Zeng¹, Xiaopeng Wei¹² and Qiang Zhang²*

1 School of Mechanical Engineering, Dalian University of Technology, Dalian, P.R. China
2 Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian, P.R. China
* Corresponding author E-mail: zhangq@dlu.edu.cn

Received 6 Nov 2012; Accepted 18 Mar 2013

DOI: 10.5772/56401

© 2013 Zeng et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract A free-floating space manipulator has outstanding advantages and wide application prospects compared with other categories. This paper discusses the concepts and categories of space robots and introduces the current trajectory-planning algorithm for a space robot and its ground simulation system in the main countries of the world. This paper also constructs a system model for the space manipulator system and gives the kinematic equation of such a manipulator system. A dynamic equation of the manipulator's joints is also developed by using Lagrange equations. Continuous Cartesian trajectory planning is also studied in this paper based on differential kinematic equations and momentum conservation equations. Finally, this paper presents the subsection evolution algorithm in GA to realize the trajectory tracking of a free-floating space manipulator and a simulation about the free-floating space manipulator and its corresponding analyses are given.

Keywords Subsection Evolution, Space Manipulator, Trajectory Planning, Genetic Algorithm

1. Introduction

Space has attracted special interest as a new application field of robotics. The space robot system, composed by a satellite and a manipulator attached to the spacecraft, is widely applied in complicated and dangerous space missions such as the capture and on-orbit services of a space target. In free-flying mode, by using the limited chemical fuel, the thruster jets of the spacecraft can compensate for the disturbance to the spacecraft caused by the movement of the manipulator. In order to conserve the precious fuel carried by the spacecraft, the thruster jets of the spacecraft should be turned off, thus the spacecraft is uncontrolled. Such a system is called a free-floating space manipulator (FFSM) system and the dynamic coupling effect between the manipulator and the spacecraft caused by the unfixed spacecraft attitude of the FFSM system is inevitable, so FFSM motions induce the disturbances to the spacecraft attitude [1].

A well-known space robot, the Shuttle Remote Manipulator System (SRMS or “Canadarm”) [2], was operated to assist the capture and berth of the satellite
from the shuttle by the astronaut. NASA missions STS-61, STS-82 and STS-103 repaired the Hubble Space Telescope using astronauts with the help of SRMS. Engineering Test Satellite VII (ETS-VII) from NASDA, Japan demonstrated that the space manipulator could capture a cooperative satellite whose attitude is stabilized during the demonstration via teleoperation from the ground-base control station [5, 4]. The space robotic technologies mentioned above demonstrated the usefulness of a space robot for space service. The space manipulator is a complex flexible system, the forward and inverse kinematics equations of the manipulator involve a large number of transcendental functions such as cosine, sine, inverse tangent and square root. In conventional methods, transcendental functions are calculated by the Taylor series expansion or LUT (Look-up Table). However, the above two approaches are not attractive solutions to the real-time control of the manipulator and the accuracy of the end effector [5,6].

Many researchers have focused on kinematics, the motion-planning problems of the space manipulator systems. Papadopoulos and Tortopidis presented a path planning methodology in joint space for the planar FFSM. The method is based on mapping the non-holonomic constraint to a space where it can be satisfied trivially[9,10]. The reachable workspace of the FFSM with zero-disturbance spacecraft attitude is analysed and discussed in detail in [10].

The dynamic reaction forces and moments due to the manipulator motion will disturb the space base, especially when the space robot is in a free-floating situation. The longer the motion time of the space manipulator is, the greater the disturbance to the base will be. Panfeng Huang and Jie Yan presented an interval algorithm for tracking trajectory of a space manipulator for capturing an uncontrolled spinning satellite, and handled the difficulties as the feasible motion realization in joint space [11]. Time-optimal manipulator control strategy of a free-floating space robot is presented. The optimization procedure by Ma and Watanabe [12] was applied to solve the simplified optimization problem. A time-optimal control strategy of a free-floating space robot was discussed considering an arbitrary prescribed path for the manipulator hand and the dynamic constraints on reaction torque by Tomohisa Oki and Hiroki Nakaniishi. The strategy was verified by numerical simulation with a free-floating space robot model equipped with a seven-DOF (degrees-of-freedom) manipulator and a system of three reaction wheels [13]. In [14], a manipulator path planning method for a space robot is discussed considering the limitations on the torque of reaction wheels and the manipulator path is determined by using an optimization technique.

This paper is organized as follow. Section 2 presents a model for the space manipulator system and gives the kinematic equation of such a manipulator system. A dynamic equation is also developed by using Lagrange equations. In Section 3, the procedure of the trajectory-planning algorithm in Cartesian space is explained. Also, the approach of GA developed by Panfeng Huang and Jie Yan et al. is demonstrated. In addition, the subsection evolution of GA for trajectory planning of FFSM is presented in detail. In Section 4, the simulation is shown compared to the desired trajectory. The final section summarizes the whole paper and gives some conclusions.

2. Kinematic Model, Dynamics Model And Motion Equation

A. Kinematic Model of Space Robot

In this paper, the authors assume a simple model of a space robot system, which is composed of a space base and a robotic manipulator arm mounted on the space base.

The space manipulator is installed on the space platform to fetch parts for astronauts, catch assemblies and so on. Fig. 1. depicts the structure and coordinates of the manipulator. This space manipulator is mainly composed of six modular rotary joints, one gripper and one vision camera, which is installed on the gripper. The central controller is placed on the space platform to perform kinematic control.

In this section, we select a manipulator with six degrees-of-freedom as our studied object.

A six-DOF space manipulator mounted on a spacecraft is used to demonstrate the analytical results. The six joints are all revolute joints. The structure and coordinate system of the space manipulator are shown in Fig. 1. The D-H parameters of the space manipulator are listed in Tab. 1.

![Figure 1. D-H coordinates of the space manipulator](image)

The camera on the gripper gets the pose and position of the target and forms the homogeneous transform matrix \( T_g \) defined in gripper coordinate.
Joint | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$
--- | --- | --- | --- | ---
1 | 0 | 0 | 0 | $\theta_1(0)$
2 | 0 | 90° | $l_1$ | $\theta_2(0)$
3 | $l_2$ | 90° | 0 | $\theta_3(90°)$
4 | $l_3$ | -90° | 0 | $\theta_4(-90°)$
5 | 0 | 0 | $l_4$ | $\theta_5(0)$
6 | $l_5$ | 90° | 0 | $\theta_6(90°)$

Table 1. D-H Parameters of the Manipulator

$$^{1}\mathbf{T}_0 = \begin{bmatrix}
T_x & T_y & T_z & T_p \\
0 & 0 & 0 & 1
\end{bmatrix}$$

In order to plan the desired set points of the Cartesian coordinate trajectory for the gripper to catch the target, $^{1}\mathbf{T}_0$ needs to be transformed into $^{6}\mathbf{T}_0$ and defined in the base coordinate. According to the Denavit-Hartenberg method, the transformation can be expressed in the chain product of six successive homogeneous matrices of $^{i+1}\mathbf{T}_i$. $^{i+1}\mathbf{T}_i$ is the homogeneous transformation matrix relating the $i$th coordinate frame to the $(i-1)$th coordinate frame.

Equation (2) contains a large set of trigonometric functions: $\sin \theta = s_i$ and $\cos \theta = c_i$. $^{i+1}\mathbf{T}_i$ can be calculated by following:

$$^{i+1}\mathbf{T}_i = \begin{bmatrix}
6n_x & 6n_y & 6n_z & 6a_x & 6a_y & 6a_z & 6p_x & 6p_y & 6p_z & 0 & 0 & 0 & 1
\end{bmatrix}$$

B. Inverse Kinematic for 6-DOF Manipulator

The inverse kinematics solution procedure for a six-DOF manipulator is indicated as follows:

$$\cos \theta_4 = \frac{l_{23}^2 + l_{45}^2 - d^2}{2l_{13} \times l_{14}}$$

$$\theta_4 = \arccos\left(\frac{l_{23}^2 + l_{45}^2 - d^2}{2l_{13} \times l_{14}}\right)$$

We obtain the equation from (4) as:

$$(s_{45}l_5 + s_{4}l_4)^2 + (c_{45}l_5 + c_{4}l_4 + l_{23})^2 = 6 \times p_x^2 + 6 \times p_y^2 + (6 \times p_z - l_1)^2$$

Then:

$$\theta_4 = \arccos\left(\frac{s_{45}l_5 + s_{4}l_4}{c_{45}l_5 + c_{4}l_4 + l_{23}}\right) - \theta_4$$

We could also obtain the expressions as follows:

$$\theta_2 = \arcsin\left(\frac{c_{45}l_5 + c_{4}l_4 + l_{23}}{\sqrt{(p_x - l_1)^2 + p_y^2}}\right)$$

$$\theta_1 = \arccos\left(\frac{s_{45}l_4 - c_{45}l_5 - c_{4}l_4}{\sqrt{(p_x - a_{l_5})^2 + (p_y - a_{l_5})^2}}\right)$$

$$\theta_3 = \arctan\left(\frac{c_{4}X + p_x - a_{l_5}c_{4}}{s_{4}l_4^4(a_z - s_{4}l_5)}\right)$$

C. Dynamics Model of Space Robot

Some important research achievements in space robot dynamics and control were collected by Xu and Kanade.

Figure 3. Dynamics Model of Space Manipulator System.
As shown in Fig. 3., the dynamics model assumed can be treated as a set of \( n + 1 \) rigid bodies connected with \( n \) joints that form a tree configuration, each joint numbered in a series of 1 to \( n \). Two coordinate systems are defined. One is the inertial coordinate system \( \Sigma \) in the orbit, the other is the base coordinate system \( \Sigma_0 \) attached on the base body with its origin at the centroid of the base. All vectors in this paper are expressed in terms of coordinate \( \Sigma_0 \). Three appropriate parameters such as Roll, Pitch and Yaw are described for the attitude of the base and end-effector. Ellery \[18\] defined the notations as follows:

\[
\begin{align*}
\Sigma_i & \quad \text{--- the } i_{th} \text{ link frame} \\
\Sigma_e & \quad \text{--- the manipulator end-effector frame} \\
\Sigma_b & \quad \text{--- the inertial frame} \\
\mathbf{r}_i & \quad \text{--- the position of the mass centre of link } i \text{ with respect to the inertial coordinates} \\
\mathbf{r}_{\Sigma} & \quad \text{--- the position of the mass centre of the entire system with respect to the inertial coordinates} \\
\mathbf{p}_e & \quad \text{--- the position of the end-effector with respect to the inertial coordinates} \\
\mathbf{l}_i & \quad \text{--- the vector from joint } i \text{ to joint } i + 1 \\
\mathbf{a}_i & \quad \text{--- the vector from joint } i \text{ to the mass centre of link } i \\
\mathbf{b}_i & \quad \text{--- the vector from the mass centre of link } i \text{ to joint } i + 1 \\
m_i & \quad \text{--- the mass of each rigid link } i \\
M_i & \quad \text{--- the total mass of the system} \\
\mathbf{I}_i & \quad \text{--- the inertia matrix of link } i \text{ with respect to the mass centre} \\
\omega_i & \quad \text{--- the angular velocity of the link } i \\
\end{align*}
\]

Where \( i = 0, 1, 2, \ldots, n \)

D. Kinematic and Dynamic Equation of Space Robot

Wenfu Xu and Yu Liu et al. have collected some important research on kinematic equations \[22\]. \( \mathbf{r}_i \) and \( \mathbf{p}_e \) are defined as the position vector points pointing toward the centre of the \( \mathbf{I}_b \) link with respect to coordinates \( \Sigma_i \) and \( \Sigma_b \) respectively and let \( \mathbf{r}_i \) be the position vector pointing toward the centre of the space base with respect to inertial coordinates \( \Sigma_b \). Let \( \mathbf{v}_i \) and \( \omega_i \) be linear and angular velocities of the \( \mathbf{I}_b \) link with respect to \( \Sigma_b \) and let \( \mathbf{v}_e \) and \( \omega_e \) be that with respect to \( \Sigma_0 \), where

\[
\mathbf{r}_i = \mathbf{r}_0 + \mathbf{b}_0 + \sum_{k=1}^{i} (\mathbf{a}_k + \mathbf{b}_k) \quad \mathbf{a}_i = \mathbf{r}_0 + \mathbf{b}_0 + \sum_{k=1}^{n} (\mathbf{a}_k + \mathbf{b}_k) 
\]

Then we can obtain:

\[
\begin{bmatrix}
\mathbf{v}_e \\
\omega_e
\end{bmatrix} = \mathbf{J}_b \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + \mathbf{J}_m \dot{\mathbf{q}}
\]

\[
\mathbf{J}_b = \begin{bmatrix}
\mathbf{E}_3 \\
\mathbf{0}
\end{bmatrix} \in \mathbb{R}^{6 \times d} \quad \text{is the Jacobian matrix of the centroid of the space robotic base and } \mathbf{J}_m = \begin{bmatrix}
\mathbf{J}_w \\
\mathbf{J}_{\Sigma_0}
\end{bmatrix} \in \mathbb{R}^{6 \times d} \quad \text{is the Jacobian matrix of the space manipulator.}
\]

The linear and the angular momentum conservation of the space manipulator are as follows:

\[
P = \sum_{i=0}^{n} m_i \mathbf{r}_i, \quad L = \sum_{i=0}^{n} (I_i \mathbf{w}_i + \mathbf{r}_i \times m_i \mathbf{r}_i) = 0
\]

Then we can obtain that:

\[
P = \begin{bmatrix}
\mathbf{M} \\
\mathbf{E}_6 \mathbf{g}
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + J_{TM} \dot{\mathbf{q}} = 0
\]

\[
L_0 = \begin{bmatrix}
\mathbf{M} \\
\mathbf{E}_6 \mathbf{g}
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + \mathbf{I}_p \dot{\mathbf{\omega}_p} = 0
\]

Where \( \mathbf{M} = \sum_{i=0}^{n} m_i \) is total mass of the entire system.

\[
J_{TM} = \sum_{i=0}^{n} (m_i J_{I_i}) \in \mathbb{R}^{3 \times n}, \quad \mathbf{r}_e = \frac{\sum_{i=0}^{n} m_i \mathbf{r}_i}{n + 1} \quad \text{and we define }
\]

\[
\mathbf{r}_o = \mathbf{r}_e - \mathbf{r}_0. \quad \text{We also define } I_w = \sum_{i=0}^{n} (I_i + m_i \mathbf{r}_i^2 \mathbf{r}_o) + I_0 \quad \text{and } I_p = \sum_{i=1}^{n} (I_{J_{ai}} + m_i \mathbf{r}_i^2 \mathbf{r}_o) 
\]

Then we can obtain the momentum conservation equations of space robotics:

\[
\begin{bmatrix}
P \\
L_0
\end{bmatrix} = H_b \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + H_p \dot{\mathbf{\omega}_p} = 0
\]

Where \( H_b = \begin{bmatrix}
\mathbf{M}_3 \\
\mathbf{M}_o \mathbf{g}
\end{bmatrix}, \quad H_p = \begin{bmatrix}
\mathbf{M} \\
\mathbf{E}_6 \mathbf{g}
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + J_{TM} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix}
\]

Then we obtain the motion equation of the space robot:

\[
\begin{bmatrix}
\mathbf{v}_e \\
\omega_e
\end{bmatrix} = -(J_b H_b^{-1} H_p + J_m) \dot{\mathbf{q}} = \mathbf{J}_b (\mathbf{v}_0, \dot{\mathbf{q}}, \dot{\mathbf{\omega}_m}, I_0) \dot{\mathbf{\omega}_m}
\]

Where \( \mathbf{J}_b \) is the generalized Jacobian matrix of the space robot.

\[
\mathbf{L} = \frac{1}{2} \begin{bmatrix}
\mathbf{v}_0^T, \omega_0^T \\
\mathbf{m}_q
\end{bmatrix} \begin{bmatrix}
\mathbf{H}_p \\
\mathbf{H}_m
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix} + \frac{1}{2} \mathbf{E}_6 \begin{bmatrix}
\mathbf{v}_0^T, \omega_0^T, \dot{\mathbf{\omega}_p}^T
\end{bmatrix} \begin{bmatrix}
\mathbf{H}_p \\
\mathbf{H}_m
\end{bmatrix} \begin{bmatrix}
\mathbf{v}_0 \\
\omega_0
\end{bmatrix}
\]

Where \( \mathbf{H}_m = \sum_{i=1}^{n} \begin{bmatrix}
\mathbf{J}_{J_{ai}} \\
\mathbf{J}_{\Sigma_0}
\end{bmatrix} \) \quad \text{and } \mathbf{H} = \begin{bmatrix}
\mathbf{H}_p \\
\mathbf{H}_m
\end{bmatrix} \in \mathbb{R}^{6(n+6) \times (n+6)}
\]

We define generalized coordinates \( q = [v_0^T, \omega_0^T, \Theta^T]^T \) and generalized force \( \tau = [F_b^T]_m + [J_b^T]_m F_e \) as the forces on the base and the manipulator of the space robot, \( F_b \) is the external force on the end of manipulator.

We assume that \( C(q, \dot{q}) = [c_b, c_m, c_e] \) and \( c_b, c_m, c_e \in \mathbb{R}^6 \) are, respectively, nonlinear forces with respect to the motion of the base and the manipulator (Coriolis force and centripetal force).

Then we can obtain the Lagrange dynamic Equation:

\[
\begin{bmatrix}
H_S & H_p & \dot{V}_0 \\
H_p^T & H_m & \dot{c}_b \\
V_0^T & c_m & \dot{c}_m \\
\end{bmatrix}
\begin{bmatrix}
\dot{V}_0 \\
\dot{c}_b \\
\dot{c}_m \\
\end{bmatrix}
= \begin{bmatrix}
F_b \\
F_m \\
F_e \\
\end{bmatrix}
\]

It has been found that free-floating systems are subject to unpredictable dynamic singularities in their manipulator workspaces owing to the attitude motion of the platform at which point they become unstable \([19, 20]\). Indeed, dynamic singularities are characteristic of free-floating systems and are functions of the mass and inertia of the composite spacecraft/robot system. These singular joint configurations cannot be mapped into unique points in the workspace since the generalized Jacobian is a dynamic function rather than being purely kinematic and spacecraft attitude coordinates do not map uniquely to end-effector coordinates.

The bidirectional search mapped between the original state and the final desired configuration using the generalized Jacobian to avoid excessive joint torques is used for obstacle avoidance or for reorientation of attitude. A Moore-Penrose pseudo-inverse version of the generalized Jacobian \( J^{-1}_b = (J_b^T J_b)^{-1} J_b^T \) was used to overcome the dynamic singularities problem by using the redundant degrees of freedom \([21]\). It allowed changes in spacecraft attitude while keeping the generalized position of the end-effector fixed with respect to the inertial reference frame to provide greater flexibility of operation.

3. Subsection Evolution in Trajectory-Planning

A. Trajectory Planning in Cartesian Space

Trajectory planning in Cartesian space is always described as follows:

**Step 1.** Set the continuous trajectory of the manipulator end-effector in Cartesian space as \( p_e(t), \phi_e(t) \).

**Step 2.** Plan the linear velocity of the manipulator end-effector by \( v_e(t) = \dot{p}_e(t) \).

**Step 3.** Plan the angular velocity of the manipulator end-effector by \( \omega_e(t) = r \cdot \phi_e(t) \), where \( r = \frac{e_0}{e_0} \), \( e_0 \) is the direction deviation of the manipulator end-effector.

**Step 4.** Compute the velocity of the joint on the inverse kinematics equation by the angular and linear velocity of the manipulator end-effector \( \dot{\theta}_d(t) = \left( J_b(t) \right)^{-1} \left[ \omega_e(t) \right] \).

Where \( J_b^{-1} \) is the inverse damping minimum variance of the generalized Jacobian matrix of the robot. It can be presented as: \( \dot{\theta}_d(t) = (J_b)^{-1} \left( J_b^T J_b \right)^{-1} \omega_e(t) \). Then \( \dot{\theta}_d(t + \Delta t) = \dot{\theta}(t) + \dot{\theta}_d(t) \Delta t \), where \( \Delta t \) is the joint control period and the damping factor \( 0 < \lambda \leq 1 \).

**Step 5.** Compute the angular velocity of the space manipulator base attitude by (7), then obtain the base attitude \( \Phi_0 \) by numerical integration.

Trajectory planning in Cartesian Space calculates the movement of six joint angles by an inverse generalized Jacobian matrix of the space robot and the expected trajectory of the manipulator end-effector. It has some disadvantages such as reliance on the inverse velocity kinematics equation and is influenced by the singular value of the Jacobian matrix.

B. Genetic Algorithms-Based Minimum Torque Path Planning for a Space Manipulator

In the space environment, the power supply for a space manipulator is limited because of the lower ratio of the solar array. Panfeng Huang and Jie Yan presented a minimum torque path-planning scheme for a space manipulator, which used a Genetic Algorithm (GA) to minimize the objective function. The planning procedure was performed in the joint space and with respect to all constraints, such as joint angle constraints, joint velocity constraints and torque constraints. GA was used to search for the optimal joint inter-knot parameters in order to realize the minimum-torque.

![Figure 4](https://www.intechopen.com)
The basic optimal processes of GA are illustrated in Fig. 4. The algorithm procedure in [23] can optimize the trajectory using the genetic algorithm as follows:

**Step 1.** Choose the number of inter-knots \( m \), then determine the path planning strategies for each path segment. Encode the parameters of every inter-knot using the chromosomes of GA.

**Step 2.** Define the fitness function according to the 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, \( \text{pop}^1 = b_{s1}^1 | | b_{s2}^1 | | ... | | b_{sp1}^1 \).

**Step 4.** Let \( l = 1 \).

**Step 5.** Decode each binary string \( b_{sl}^l \) into parameter

\[
\text{pop}^l = q_{ij}^l | | v_{ij}^l | | l = 1, ..., m, j = 1, ..., n .
\]

Use 4-4,...,4-5 iterative path planning strategy [23] to plan the path \( Q^* \) with \( \text{pop}^l \).

**Step 6.** Compute the maximum joint angle, joint angular velocity, joint angular acceleration, \( q_{ij}^{k_{\text{max}}} \) and \( v_{ij}^{k_{\text{max}}} \) with respect to the path \( Q^* \).

**Step 7.** Compute every joint torque using inverse dynamics according to the maximal position, velocity and acceleration of the joint, \( q_{ij}^{k_{\text{max}}} \) and \( v_{ij}^{k_{\text{max}}} \).

**Step 8.** Let \( l = l + 1 \) and return to Step 5 until \( l = n_p \).

**Step 9.** Summarize the absolute values of all torques from \( l = 1 \) to \( l = n_p \) according to fitness function. Collect the minimum fitness function value.

**Step 10.** Let \( k = k + 1 \), generate a new population \( \text{pop}^{k} \) using reproduction, crossover and mutation operators. Return to Step 4 until \( k = n_g \).

**Step 11.** Obtain the minimum fitness function value, thus, the parameters in this situation as optimal values. Get the optimum trajectory \( Q^* \).

Firstly, some inter-knots were selected to generate the trajectory segments. In each trajectory segment, a suitable polynomial is used to obtain the local trajectory. Thus, the whole trajectory is 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 the inter-knots of each trajectory segment were optimized by the GAs. Meanwhile, the path is evolved into the optimum trajectory connecting initial and the final points in the joint space.

In [23], GA was used to solve the optimal trajectory planning described by the cost function with a dynamic model under initial and final conditions and constraints. The whole trajectory was divided into several trajectory segments. The path point connecting with two segments was called the knot point. The optimal algorithm was to search for the optimum parameters of each knot point, such as joint angle and joint angular velocity, then to realize minimum torque trajectory planning. The coefficients of the polynomials depended on the trajectory parameters, \( q_{ij} \) and \( v_{ij} \) if the travel time \( t_i \) of every trajectory segment was fixed [24]. Therefore, these parameters will be evolved by the genetic algorithm.

**C. Trajectory Planning on subsection evolution using GA**

In this section, the steps of the trajectory planning for FFSM is presented. Firstly, we set a continuous trajectory of end-effector and divide the whole trajectory into a variety of subsections. We call the path points as inter-knots which cut the trajectory into sections. Then we encode the parameters of every subsection and define GA parameters and also fitness function according to the kinetic energy. Then we obtain the movement, velocity and acceleration of the joints by calculating the population. The approach could search for the optimum parameters of the subsection to realize minimum energy planning. The new algorithm procedure to optimize the trajectory using subsection evolution by a genetic algorithm is as follows:

**Step 1.** Set the continuous trajectory of the manipulator \( p_{e}(t) \), then set the \( n \) inflection points as the crucial segmentation points to divide the trajectory into \( n-1 \) parts of the whole trajectory. Choose the number \( m-1 \) to divide each subsection by inter-knots, then the number \( m, n \) of initial inter-knots can be obtained. Draw a circle with the centre at each initial inter-knot and with a radius equal to \( \tau \). Obtain the inter-knots by proportional spacing. Encode the parameters of every subsection using chromosomes of GA and generate the initial population. The initial inter-knots, which construct
different subsections, are connected to form a continuous trajectory.

**Step 2.** Define fitness function according to kinetic energy.

The whole kinetic energy of the space system is already expressed:

\[
L = \frac{1}{2} \begin{bmatrix} v_{0}^T, \omega_{0}^T, q \end{bmatrix} \begin{bmatrix} H_0 & H_0 & H_m \\ H_0 & H_0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{0} \\ \omega_{0} \\ q \end{bmatrix} = \frac{1}{2} \begin{bmatrix} v_{0}^T, \omega_{0}^T, \Theta^T \end{bmatrix} H \begin{bmatrix} v_{0} \\ \omega_{0} \\ \Theta \end{bmatrix} \quad (9)
\]

\[
f(x) = \begin{cases} C_{\text{max}} - \frac{1}{2} \begin{bmatrix} v_{0}^T, \omega_{0}^T, \Theta^T \end{bmatrix} H \begin{bmatrix} v_{0} \\ \omega_{0} \\ \Theta \end{bmatrix} & \text{when } g(x) < C_{\text{max}} \quad (11) \\ 0 & \text{others} \end{cases}
\]

Where, \( C_{\text{max}} \) is the total energy consumed by the system.

**Step 3.** Define GA parameters.

**Mutation:**

- Add an inter-knot
- Reduce an inter-knot
- Move an inter-knot

**Figure 6.** Mutation operator

Every new inter-knot is generated by the circle with the centre at each initial inter-knot and with a radius equal to \( r_{i} \). Every new inter-knot creates a new trajectory after every generation. The mutation rate \( p_{m} = 0.05 \).

**Crossover:**

As shown in Fig. 6, two trajectories exchange two inter-knots to create new trajectories for crossover. The crossover parameters \( M_{i} \) are determined by the crossover rate \( p_{c} \). We set \( p_{c} = 0.4 \) in this paper.

**Reproduction:**

We determine the roulette wheel method, which is the base of GA \(^{22}\) and set the reproduction rate \( p_{r} = 0.2 \).

**Step 4.** Let generation number \( \text{Gen} = 1 \)

**Step 5.** Obtain the movement, velocity and acceleration of the joints by calculating the population. Solve the generalized Jacobian matrix of the space robot \( J_{g} \) by the performance of the joints and the base to the inertial coordinate system. Calculate the velocity of every joint and the base by the inverse kinematics equation of velocity. Obtain the fitness function value.

**Figure 7.** Crossover operator

**Step 6.** Let \( \text{Gen} = \text{Gen} + 1 \)

**Step 7.** Generate a new population using reproduction, crossover and mutation operators. Go to Step 4.

**Step 8.** Obtain the minimum fitness function value. Calculate the error \( \zeta_{e} \) of the end of the space manipulator to judge whether or not it’s within the allowable error \( \zeta_{c} \). Go to Step 4. If \( \zeta_{e} > \zeta_{c} \), the parameters in this situation are optimal values. Obtain the optimum trajectory.

**4. Simulation**

The detailed derivation of dynamic equations of space robots can be obtained from Section 2. The parameters of the space robot are shown in Table 2. 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 a space robot as follows:

\[
v_{\text{imax}} = 3 \text{rad/s}, i = 1,2,\ldots,6
\]

\[
a_{\text{imax}} = 5 \text{rad/s}^2, i = 1,2,\ldots,6
\]

<table>
<thead>
<tr>
<th>Link</th>
<th>( I_{y}(m) )</th>
<th>( m_{i}(kg) )</th>
<th>( I_{xx} )</th>
<th>( I_{yy} )</th>
<th>( I_{zz} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sat</td>
<td>0.2</td>
<td>400</td>
<td>30</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>Link1</td>
<td>0.3</td>
<td>5</td>
<td>0.105</td>
<td>0.105</td>
<td>0.05</td>
</tr>
<tr>
<td>Link2</td>
<td>0.6</td>
<td>5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.075</td>
</tr>
<tr>
<td>Link3</td>
<td>0.8</td>
<td>6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.15</td>
</tr>
<tr>
<td>Link4</td>
<td>0.3</td>
<td>4</td>
<td>0.15</td>
<td>0.15</td>
<td>0.1</td>
</tr>
<tr>
<td>Link5</td>
<td>0.3</td>
<td>3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.05</td>
</tr>
<tr>
<td>Link6</td>
<td>0.1</td>
<td>2</td>
<td>0.05</td>
<td>0.05</td>
<td>0.02</td>
</tr>
</tbody>
</table>

**Table 2.** Parameters of Space Robot System
According to the simulation result, Fig. 8 shows the trajectory of the manipulator end-effector. The plot shows that it starts from the initial position at $t = 0$ seconds and reaches the final position at $t = 40$ seconds. It also shows the linear velocity and acceleration of the manipulator end-effector.

The goal of the simulation is to verify the performance of GA optimization.

As shown in Fig. 8, the planned path is smoother and more applicable for the control of the manipulator. The structure of the algorithm is more flexible for solving the general problems in these kinds of algorithms using inter-knots if introducing the other parameters such as the corner points by generating inter-knots. The error between the actual trajectory and the desired trajectory can be kept in an allowed range by introducing the error parameter into the algorithm. The number of inter-knots is chosen manually before by research. The approach in this paper generates the inter-knots in a particular way to reduce time by selecting the points. (In Fig. 8, the unit of the Y axis is cm.)

![Figure 8. Trajectory of end-effector](image)

![Figure 9. The simulation of space manipulator](image)

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

5. Conclusion

In this paper, a new subsection evolution algorithm for trajectory planning is developed. The method is based on genetic algorithms, which can search most satisfactory parameters of subsections to generate the optimal path. The optimal path obtained is fit for high velocity and high precision dynamic control.

6. Acknowledgement

This work is supported by the Program for Changjiang Scholars and Innovative Research Team in University (No.IRT1109), the Program for Liaoning Science and Technology Research in University (No.LS2010008), the Program for Liaoning Innovative Research Team in University (No.LT2011018), Natural Science Foundation of Liaoning Province (201102008), the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University and by "Liaoning BaiQianWan Talents Program(2010921010, 2011921009)".

7. References


