Robotica (2017) volume 35, pp. 861–875. © Cambridge University Press 2015
doi:10.1017/S0263574715000855

Nonholonomic motion planning for minimizing base disturbances of space manipulators based on multi-swarm PSO

Qiang Zhang†*, Lu Ji†, Dongsheng Zhou† and Xiaopeng Wei†‡

† *Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian, 116622, P. R. China. E-mail: zhoudongsheng@dlu.edu.cn*
‡ *College of Computer Science, Dalian University of Technology, Dalian, 116024, P. R. China. E-mail: xpwei@dlut.edu.cn*

(Accepted September 22, 2015. First published online: November 6, 2015)

SUMMARY

Because space manipulators must satisfy the law of conservation of momentum, any motion of a manipulator within a space-manipulator system disturbs the position and attitude of its free-floating base. In this study, the authors have designed a multi-swarm particle swarm optimization (PSO) algorithm to address the motion planning problem and so minimize base disturbances for 6-DOF space manipulators. First, the equation of kinematics for space manipulators in the form of a generalized Jacobian matrix (GJM) is introduced. Second, sinusoidal and polynomial functions are used to parameterize joint motion, and a quaternion representation is used to represent the attitude of the base. Moreover, by transforming the planning problem into an optimization problem, the objective function is analyzed and the proposed algorithm explained in detail. Finally, numerical simulation results are used to verify the validity of the proposed algorithm.

KEYWORDS: Space manipulators; Free-floating base; Multi-swarm PSO; Minimizing base disturbance; Nonholonomic motion planning.

1. Introduction

Recently, studies of space manipulators have been of particular interest to researchers because they offer safety and efficiency advantages over human astronauts. However, control of space manipulators presents additional difficulties beyond those encountered with fixed-base manipulators, such as the property of conservation of momentum during operation. For this reason, the motion of a manipulator can disturb the position and attitude of its base and cause the end-effector to miss its target.[1] This property makes the control problem for space manipulators a nonholonomic motion-planning problem.[2–4] Generally, researchers use thrusters, orthogonal reaction wheels or control moment gyroscopes to counteract such disturbances, but these methods consume fuel and may not exert enough torque to offset the disturbance. For these reasons, base disturbances should be minimized during the planning of the motion of space manipulators.

To address these problems, Dubowsky *et al.* proposed a method called the disturbance map (DM).[5,6] This map is based on the concept of virtual manipulators. They then extended their theory to the enhanced disturbance map (EDM).[7,8] Using maps, the EDM clearly illustrates the maximum and minimum base disturbances with regard to different orientations of motion of space manipulators. Nenchev defined a fixed attitude-restricted (FAR) Jacobian matrix to ensure that manipulator motion has no impact on base attitude.[9] Yoshida presented the zero reaction maneuver (ZRM) concept,[10] which makes it possible to plan the motion of space manipulators without base disturbance, even

* Corresponding author. E-mail: zhangq@dlu.edu.cn

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

though it acts only on certain manipulators with dynamic redundancy. Yoshida and Huang presented a dual-arm coordination project, and Yamada used a calculus of variations to plan the motion of joints and so change the attitude of the base arbitrarily.[1,11,12] Suzuki and Nakamura developed Yamada's method and then proposed an optimal spiral motion.[13] However, this method is constrained by dynamic singularity. Papadopoulos used a parameter optimization method but illustrated it only using 2-DOF space manipulators. Other methods have been proposed in previous works.[2,14–24]

In this paper, a planning method inspired by previous work is proposed to plan nonholonomic motion to minimize base disturbance for 6-DOF space manipulators. The paper is organized as follows: Section 2 derives the direct kinematic equations of the manipulator system. Section 3 parameterizes the joint motion and the system state. Section 4 transforms the planning problem to an optimization problem. Section 5 introduces the detailed steps of the multi-swarm PSO algorithm. Section 6 shows the results of computer simulation and Section 7 summarizes the paper.

2. Modeling of Space/Free-Floating Manipulators

Symbols and notation

$\boldsymbol{r}_0 \in \boldsymbol{R}^3(m)$	position vector of the center of mass (CM) of the base with respect to inertial coordinates;
$\boldsymbol{r}_i \in \boldsymbol{R}^3\ (\{i = 1, ..., n\})\,(m)$	position vector of the CM of link ii with respect to inertial coordinates;
$\boldsymbol{r}_g \in \boldsymbol{R}^3(m)$	position vector of the system CM with respect to inertial coordinates;
$\boldsymbol{b}_0 \in \boldsymbol{R}^3(m)$	position vector from the CM of base to joint 1 with respect to inertial coordinates;
$\boldsymbol{p}_i \in \boldsymbol{R}^3\ (\{i = 1, ..., n\})\,(m)$	position vector of joint ii with respect to inertial coordinates;
$\boldsymbol{p}_e \in \boldsymbol{R}^3(m)$	position vector of the end-effector with respect to inertial coordinates;
$\boldsymbol{v}_0 \in \boldsymbol{R}^3(m/s)$	linear velocity of the base at a given time;
$\boldsymbol{v}_e \in \boldsymbol{R}^3(m/s)$	linear velocity of the end-effector at a given time;
$\boldsymbol{\omega}_0 \in \boldsymbol{R}^3(\text{rad}/s)$	angular velocity of the base at a given time;
$\boldsymbol{\omega}_e \in \boldsymbol{R}^3(\text{rad}/s)$	angular velocity of the end-effector at a given time;
$\boldsymbol{k}_i \in \boldsymbol{R}^3\ (i = 1, ..., n)$	unit direction vector of the z-axis of frame ii, $\Sigma_i\ (i = 1, ..., n)$;
$\theta_i(\text{rad})$	joint angle ii;
$\boldsymbol{\Theta} \in \boldsymbol{R}^n(\text{rad})$	joint angle vector;
$\boldsymbol{I}_i \in \boldsymbol{R}^{3\times3}(kg \cdot m^2)$	inertia matrix of link ii with respect to itself CM (all of links are regarded as cylinder);
$\boldsymbol{X}$	status of the total system(include attitude and position);
$\boldsymbol{Q}_b$	attitude of the base;
$\boldsymbol{P}_b(m)$	position of the base.

The path planning problem for free-floating space robots in Cartesian space cannot be solved using position-level kinematic equations as for fixed-base manipulators because the position and orientation of the end-effector depend not only on the geometrical configuration of the manipulators, but also on the inertial properties of the system, which change according to its configuration. This means that the history of postural changes must be considered to solve the motion-planning problem. To deal with this problem, Umetani, and Yoshida have proposed the concept of a GJM,[25] which expresses velocity-level kinematic equations for space manipulators rather than position-level equations.

The characteristic equation of the manipulators is as follows:

$$\boldsymbol{p}_e = \boldsymbol{r}_0 + \boldsymbol{b}_0 + \sum_{j=1}^{n} \left(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\right). \tag{2.1}$$

The end-effector velocity can be obtained by differentiating Eq. (2.1) with respect to time:

$$\boldsymbol{v}_e = \dot{\boldsymbol{p}}_e = \boldsymbol{v}_0 + \boldsymbol{\omega}_0 \times \left(\boldsymbol{p}_e - \boldsymbol{r}_0\right) + \sum_{i=1}^{n} \left[\boldsymbol{k}_i \times \left(\boldsymbol{p}_e - \boldsymbol{p}_i\right)\right] \cdot \dot{\boldsymbol{\theta}}_i. \tag{2.2}$$

The angular velocity of the end-effector can also be written in a similar form:

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_0 + \sum_{i=1}^{n} \boldsymbol{k}_i \dot{\boldsymbol{\theta}}_i. \tag{2.3}$$

The differential form of the kinematic equation for a free-floating manipulator system is as follows:

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}_b \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \boldsymbol{J}_m \dot{\boldsymbol{\Theta}}, \tag{2.4}$$

where

$$\boldsymbol{J}_m = \begin{bmatrix} \boldsymbol{k}_1 \times (\boldsymbol{p}_e - \boldsymbol{p}_1) \dots \boldsymbol{k}_n \times (\boldsymbol{p}_e - \boldsymbol{p}_n) \\ \boldsymbol{k}_1 \qquad \dots \qquad \boldsymbol{k}_n \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{J}_b = \begin{bmatrix} \boldsymbol{E} & -\tilde{\boldsymbol{p}}_{0e} \\ 0 & \boldsymbol{E} \end{bmatrix}, \quad (\boldsymbol{p}_{0e} = \boldsymbol{p}_e - \boldsymbol{r}_0). \tag{2.6}$$

In which $\boldsymbol{J}_b$, $\boldsymbol{J}_m$ are the Jacobian matrices of the manipulators and the base respectively. In addition to its initial zero, the momentum of the whole system will remain invariant in inertial space during the operation because neither external forces nor torques act on the base and the manipulators. Therefore, the equation of momentum conservation is

$$\boldsymbol{I}_b \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \boldsymbol{I}_m \dot{\boldsymbol{\Theta}} = 0, \tag{2.7}$$

where

$$\boldsymbol{I}_b = \begin{bmatrix} M \cdot \boldsymbol{E} & M \cdot \tilde{\boldsymbol{r}}_{0g}^T \\ M \cdot \tilde{\boldsymbol{r}}_{0g} & \boldsymbol{I}_M \end{bmatrix}, \quad (\boldsymbol{r}_{0g} = \boldsymbol{r}_g - \boldsymbol{r}_0) \tag{2.8}$$

$$\boldsymbol{I}_{bm} = \begin{bmatrix} \boldsymbol{J}_{TM} \\ \boldsymbol{I}_{M\phi} \end{bmatrix} \tag{2.9}$$

are the inertia matrix of base and the coupling inertia matrix between base and manipulator respectively. The items of above equations are as follows:

$$\boldsymbol{I}_M = \sum_{i=1}^{n} \left(\boldsymbol{I}_i + m_i \cdot \tilde{\boldsymbol{r}}_{0i}^T \cdot \tilde{\boldsymbol{r}}_{0i}\right) + \boldsymbol{I}_0 \in \boldsymbol{R}^{3\times 3}, \quad (\boldsymbol{r}_{0i} = \boldsymbol{r}_i - \boldsymbol{r}_0) \tag{2.10}$$

$$\boldsymbol{J}_{TM} = \sum_{i=1}^{n} (m_i \cdot \boldsymbol{J}_{Ti}) \in \boldsymbol{R}^{3\times n} \tag{2.11}$$

$$\boldsymbol{I}_{M\phi} = \sum_{i=1}^{n} (\boldsymbol{I}_i \cdot \boldsymbol{J}_{Ri} + m_i \cdot \tilde{\boldsymbol{r}}_{0i} \cdot \boldsymbol{J}_{Ti}) \in \boldsymbol{R}^{3\times n} \tag{2.12}$$

$$\boldsymbol{J}_{Ti} = \left[\boldsymbol{k}_1 \times (\boldsymbol{r}_i - \boldsymbol{p}_1), \boldsymbol{k}_2 \times (\boldsymbol{r}_i - \boldsymbol{p}_2), \dots, \boldsymbol{k}_i \times (\boldsymbol{r}_i - \boldsymbol{p}_i), 0, \dots, 0\right] \in \boldsymbol{R}^{3\times n} \tag{2.13}$$

$$\boldsymbol{J}_{Ri} = [\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_i, 0, \dots, 0] \in \boldsymbol{R}^{3\times n}. \tag{2.14}$$

According to Eq. (2.7), $\boldsymbol{v}_0$, $\boldsymbol{\omega}_0$ can be solved for as

$$\begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} = -\boldsymbol{I}_b^{-1} \boldsymbol{I}_{bm} \dot{\boldsymbol{\Theta}} = \begin{bmatrix} \boldsymbol{J}_{vb} \\ \boldsymbol{J}_{\omega b} \end{bmatrix} \dot{\boldsymbol{\Theta}}. \tag{2.15}$$

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

This equation can be divided into a linear velocity part and an angular velocity part as follows:

$$\boldsymbol{\omega}_0 = \boldsymbol{J}_{\omega b}\dot{\boldsymbol{\Theta}}. \tag{2.16}$$

Finally, substituting Eq. (2.15) into Eq. (2.4) yields the desired result

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \left[\boldsymbol{J}_m - \boldsymbol{J}_b \boldsymbol{I}_b^{-1} \boldsymbol{I}_{bm}\right] \dot{\boldsymbol{\Theta}} = \boldsymbol{J}^* \left(\Psi_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i\right) \dot{\boldsymbol{\Theta}}. \tag{2.17}$$

$\boldsymbol{J}^* \left(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i\right)$ is the so-called GJM, which represents the kinematic features of the whole space-manipulator system. It is determined not only by the geometric parameters of each link, but also by the inertial properties of the overall system.

3. Description of the Motion-Planning Problem Based on Direct Kinematics

Parameterization of joint motion

To produce a smooth path, polynomial functions have been used to describe joint motion. The degree of the function must be greater than or equal to the number of constraints posed on the position, velocity and acceleration of the joint. Moreover, every joint should be limited within a certain boundary; in practice, sinusoidal functions can be used to address this problem. Therefore, in this work, each joint angle was parameterized using a sinusoidal seventh-order polynomial function:

$$\theta_i\left(t\right) = \Delta_{i1} \sin\left(a_{i7}t^7 + a_{i6}t^6 + a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t + a_{i0}\right) + \Delta_{i2}. \tag{3.1}$$

Here, the following is true:

$$\Delta_{i1} = \frac{\theta_{i\,\max} - \theta_{i\,\min}}{2}, \qquad \Delta_{i2} = \frac{\theta_{i\,\max} + \theta_{i\,\min}}{2}, \tag{3.2}$$

$a_{i0} - a_{i7}$ are the coefficients of the function and $\Delta_{i1}, \Delta_{i2}\Delta_{f1}$ and Δ_{i2} are used to limit every joint within its own angle constraint $\theta_i \in [\theta_{i\,\min,}\theta_{i\,\max,}]$. The corresponding angular velocities and accelerations are $\dot{\theta}_i\left(t\right), \ddot{\theta}_i\left(t\right)$.

Obviously, $\theta_i, \dot{\theta}_i, \ddot{\theta}_i$ should satisfy the following requirements:

$$\begin{gathered} \boldsymbol{\Theta}\left(t_0\right) = \boldsymbol{\Theta}_0, \dot{\boldsymbol{\Theta}}\left(t_0\right) = 0, \ddot{\boldsymbol{\Theta}}\left(t_0\right) = 0, \\ \boldsymbol{\Theta}\left(t_f\right) = \boldsymbol{\Theta}_d, \dot{\boldsymbol{\Theta}}\left(t_f\right) = 0, \ddot{\boldsymbol{\Theta}}\left(t_f\right) = 0. \end{gathered} \tag{3.3}$$

Here, t_0 and $t_f t_0, t_f$ are the initial and terminal times respectively and $\boldsymbol{\Theta}_d$ is calculated according to the Theory of Screws.[26,27]

Substituting Eq. (3.3) into $\theta_i\left(t\right), \dot{\theta}_i\left(t\right), \ddot{\theta}_i\left(t\right)$ yields $a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}$, and they are as follows:

$$\begin{aligned} \dot{\theta}_i\left(t\right) = {} & \Delta_{i1} \cos\left(a_{i7}t^7 + a_{i6}t^6 + a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t^1 + a_{i0}\right) \\ & \left(7a_{i7}t^6 + 6a_{i6}t^5 + 5a_{i5}t^4 + 4a_{i4}t^3 + 3a_{i3}t^2 + 2a_{i2}t + a_{i1}\right) \end{aligned} \tag{3.4}$$

$$\begin{aligned} \ddot{\theta}_i\left(t\right) = {} & -\Delta_{i1} \sin\left(a_{i7}t^7 + a_{i6}t^6 + a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t^1 + a_{i0}\right) \\ & \left(7a_{i7}t^6 + 6a_{i6}t^5 + 5a_{i5}t^4 + 4a_{i4}t^3 + 3a_{i3}t^2 + 2a_{i2}t + a_{i1}\right)^2 \\ & + \Delta_{i1} \cos\left(a_{i7}t^7 + a_{i6}t^6 + a_{i5}t^5 + a_{i4}t^4 + a_{i3}t^3 + a_{i2}t^2 + a_{i1}t^1 + a_{i0}\right) \\ & \left(42a_{i7}t^5 + 30a_{i6}t^4 + 20a_{i5}t^3 + 12a_{i4}t^2 + 6a_{i3}t + 2a_{i2}\right) \end{aligned} \tag{3.5}$$

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

$$a_{i0} = \arcsin\left(\frac{\theta_{i0} - \Delta_{i2}}{\Delta_{i1}}\right) \tag{3.6}$$

$$a_{i1} = a_{i2} = 0 \tag{3.7}$$

$$a_{i3} = -\left(3a_{i7}t_f^7 + a_{i6}t_f^6 - 10\left[\arcsin\left(\frac{\theta_{id} - \Delta_{i2}}{\Delta_{i1}}\right) - \arcsin\left(\frac{\theta_{i0} - \Delta_{i2}}{\Delta_{i1}}\right)\right]\right) \Big/ t_f^3 \tag{3.8}$$

$$a_{i4} = -\left(8a_{i7}t_f^7 + 3a_{i6}t_f^6 - 15\left[\arcsin\left(\frac{\theta_{id} - \Delta_{i2}}{\Delta_{i1}}\right) - \arcsin\left(\frac{\theta_{i0} - \Delta_{i2}}{\Delta_{i1}}\right)\right]\right) \Big/ t_f^4 \tag{3.9}$$

$$a_{i5} = -\left(6a_{i7}t_f^7 + 3a_{i6}t_f^6 - 6\left[\arcsin\left(\frac{\theta_{id} - \Delta_{i2}}{\Delta_{i1}}\right) - \arcsin\left(\frac{\theta_{i0} - \Delta_{i2}}{\Delta_{i1}}\right)\right]\right) \Big/ t_f^5. \tag{3.10}$$

Finally, only two parameters (a_{i6}, a_{i7}) remain in each joint motion function:

$$\boldsymbol{a} = [a_{16}, a_{17}, a_{26}, a_{27}, ..., a_{i6}, a_{i7}] \in \boldsymbol{R}^{2n}. \tag{3.11}$$

In the optimization algorithm, ***aa*** is also referred to as a *Lagrange multiplier*. When ***aa*** is solved for, the manipulator motion paths are determined.

Equation of system state

In this research, the aim of motion planning has been to bring the initial position and attitude of the base close to the desired values. Therefore, the equation of system state is defined as follows (with the attitude of the base represented by a *quaternion function*[28]):

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{Q}_b \\ \boldsymbol{P}b \end{pmatrix} \in \boldsymbol{R}^7. \tag{3.12}$$

The terms of the system state equation can be calculated by numerical integration:

$$\boldsymbol{Q}_b(t) = \int_0^t \frac{1}{2}\begin{bmatrix} -\boldsymbol{q}_b^T \\ \eta_b \boldsymbol{I} - \tilde{\boldsymbol{q}}_b \end{bmatrix} \boldsymbol{J}_{bm\_\omega}\dot{\boldsymbol{\theta}}dt, \tag{3.13}$$

$$\boldsymbol{p}_b = \int_0^t \boldsymbol{J}_{bm\_v}\dot{\boldsymbol{\theta}}dt. \tag{3.14}$$

The orientations of the two coordinate frames are given as $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2\boldsymbol{Q}_1$,$\boldsymbol{Q}_2$ and the orientation error is given as *dh* and *d**q**dh*,*d**q***[29]:

$$\begin{cases} \delta\eta = \eta_{b0}\eta_{bf} + \boldsymbol{q}_{b0}\boldsymbol{q}_{bf} \\ \delta\boldsymbol{q} = \eta_{b0}\boldsymbol{q}_{bf} - \eta_{bf}\boldsymbol{q}_{b0} - \tilde{q}_{b0}q_{bf} \end{cases}. \tag{3.15}$$

The goal of motion planning is to bring the distance between the final and the initial/desired state close to zero:

$$\boldsymbol{X}_{b0} - \boldsymbol{X}_{bf} \to 0. \tag{3.16}$$

Because this method does not calculate the inverse of $\boldsymbol{J}_{bm}\boldsymbol{J}_{bm}$ (the GJM), it is also referred to as a motion-planning approach based on direct kinematics.[30]

4. Solution of Motion-Planning Problem by the Optimization Algorithm

Considering the dynamic coupling relationship between the manipulators and the base, manipulator motion could disturb the position and attitude of the free-floating base that supports the manipulators. The problem of minimizing disturbance to the base can be divided two parts: (1) minimizing position disturbance; (2) minimizing attitude disturbance.[31,32] Moreover, to ensure that the joint motion cannot

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

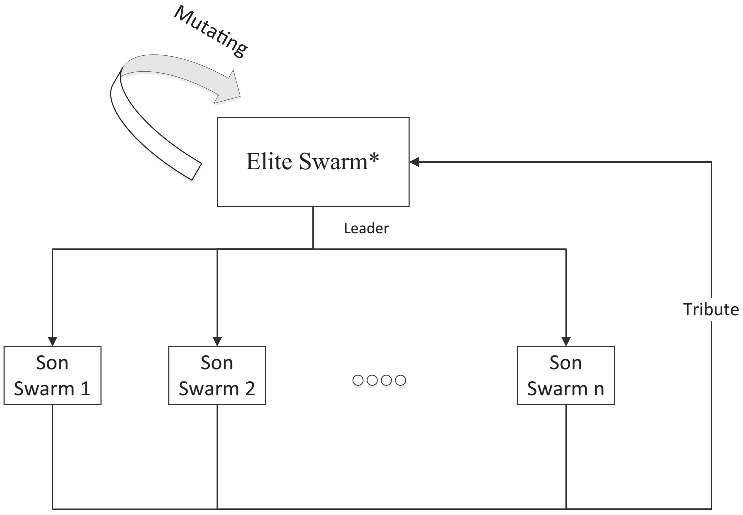

Fig. 1. Topological structure of the multi-swarm PSO algorithm.

exceed the maximal torque that the motor can provide, the joint rates and accelerations must be constrained within boundaries as previously mentioned. The objective function is defined as follows:

$$\boldsymbol{F}\left(\boldsymbol{a}\right)=\frac{\left\|d\boldsymbol{q}_b\right\|}{K_q}+\frac{\left\|d\boldsymbol{p}_b\right\|}{K_p}+\frac{L_{\dot{\theta}}}{K_{\dot{\boldsymbol{\Theta}}}}+\frac{L_{\ddot{\theta}}}{K_{\ddot{\boldsymbol{\Theta}}}}. \tag{4.1}$$

Here, $d\boldsymbol{q}_b$, $d\boldsymbol{p}_b$, $L_{\dot{\theta}}$, $L_{\ddot{\theta}}$ are the quaternion error of the base attitude Eq. (3.15), the position error of the base $\left(p_i - p_f\right)$, and the constraints on joint rates and accelerations (described in a later section), respectively. As discussed earlier, these four items are all functions of ***aa***, the Lagrange multiplier of the seventh-order polynomial function. K_q, K_p, $K_{\dot{\Theta}}$, $K_{\ddot{\Theta}}$ are determined by the accuracy requirement, where $\|x\| = \sqrt{x^T \cdot x}$.

According to the properties of the PSO algorithm, the position of each particle with regard to a higher-dimensional space can be expressed by a group of Lagrange multipliers ***aa***. Because the objective function is determined by ***aa***, it is possible to use the position of a particle, with which the minimal objective function value can be calculated, to express a solution to the motion-planning problem.

5. Motion Planning Based on Multi-Swarm PSO

PSO was originally developed by Kennedy and Eberhart.[33] It is a stochastic optimization method based on simulating the social behavior of bird flocks and fish schools. The goal of PSO is to find optimal or near-optimal solutions efficiently in large search spaces. In PSO, each individual, called a particle, and the population, called a swarm, adjusts its path toward its own and its neighborhood's previous best position. In the global variant of PSO, the whole swarm is considered the neighborhood. This enables global sharing of information from the discoveries and previous experience of all other companions during the search for promising regions of the space.

Because of the particularities of space-manipulator modeling, i.e., there are many sub-optimal solutions close to the optimal solution, general algorithms (including PSO[34–36] and Simu-APSO) cannot search effectively for the global minimum of the objective function, even with repeated parameter adjustments. For this reason, the authors have designed a multi-swarm PSO algorithm, inspired by the idea of swarm intelligence,[37] to plan the motion of a set of manipulators. Its topological structure is shown in Fig. 1.

Multi-swarm PSO consists of one elite swarm and several child swarms, where the members of the elite swarm are initialized as the best-performing particles within child swarms. Each child swarm evolves independently, but it will follow the motion of elite particles which are distributed by the elite swarm, and at the end of evolution, the best-performing particle from every child swarm will be

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

submitted to the elite swarm to replace poorly performing particles in the elite swarm. The procedure of multi-swarm PSO can be described as follows:
Step 1: Initialization of child swarms by repeatedly calculating the objective function;
Step 2: Sorting all particles based on their values of the objective function and then selecting the best N particles to initialize the elite swarm;
Step 3: Initial evolution of the elite swarm and selection of elite particles;
Step 4: Distributing the elite particles, which are the best particles in the elite swarm, to every child swarm. The velocity equation of child swarm *i* is as follows:

$$\boldsymbol{v}_i^{k+1} = \omega \boldsymbol{v}_i^k + c_1 r_1 \left(\boldsymbol{p}_{\text{best\_}i}^k - \boldsymbol{a}_i^k\right) + c_2 r_2 \left(\boldsymbol{g}_{\text{best\_}i}^k{}' - \boldsymbol{a}_i^k\right) + c_3 r_3 \left(\boldsymbol{e}_{\text{best\_}i}^k - \boldsymbol{a}_i^k\right). \tag{5.1}$$

Step 5: Independent evolution of every child swarm, selection of the best-performing particles for submission to the elite swarm and removal of poorly performing particles within the elite swarms;
Step 6: After absorbing the particles provided by the child swarms, the elite swarm evolves to the next generation according to the new position equation:

$$\boldsymbol{a}_i^{k+1} = \boldsymbol{a}_i^k + \boldsymbol{v}_i^{k+1} + \boldsymbol{\delta}. \tag{5.2}$$

Here, $\delta\boldsymbol{\delta}$ is called the *mutagenic factor* and is determined as follows:

$$\begin{cases} \delta \in U\left(a, b\right), \Delta f_i \leq \varepsilon \\ \delta \in N\left(0, \sigma\right), \Delta f_i \leq \varepsilon \end{cases}. \tag{5.3}$$

Here, $\Delta f_i = \left| f(\boldsymbol{a}_i^k) - f(\boldsymbol{a}_i^{k-1}) \right|$, and ε is the value of error bounds.
Step 7: Selection of elite particles and evaluation of whether they satisfy the convergence criterion; if they do not, return to Step 4; if they do, go to Step 8.
Step 8: End of algorithm and output of results.

Although the numbers of swarms and particles are greater than in standard PSO, because the structure of child swarms is simpler than in standard PSO (child swarms need only to update the position and velocity of every particle) and the information produced by child swarms is governed by the elite swarm, the new algorithm has greater computational efficiency than standard PSO. Because, in this algorithm, all the child swarms evolve independently, it can be optimized for parallel computation exclusively.

6. Simulation Results

6.1. Space-manipulator parameters

The object of this paper is a 6-DOF space manipulator which is mounted on a free-floating base, as shown in Fig. 2. The parameters of the whole system are shown in Table I and the D–H parameters are shown in Table II. The simulation environment is MATLAB 2010a with a "*Spacedyn*" toolbox programmed by Shimizu *et al.*[38]

The joint velocity and acceleration boundaries (which are empirical values) are defined as follows:

$$\begin{cases} \left|\dot{\theta}_i\right| \leq \dot{\theta}_{i\_\max} = 60^o \big/ s = 1.0472\text{rad} \big/ s \\ \left|\ddot{\theta}_i\right| \leq \ddot{\theta}_{i\_\max} = 60^o \big/ s = 1.0472\text{rad} \big/ s \end{cases}. \tag{6.1}$$

The coefficients of accuracy are as follows:

$$\begin{cases} \boldsymbol{K}_q = \sin\left(\pi \big/ 360\right) \\ \boldsymbol{K}_p = 2 \cdot 10^{-3}, \boldsymbol{K}_{\dot{\boldsymbol{\Theta}}} = \boldsymbol{K}_{\ddot{\boldsymbol{\Theta}}} = 0.01 \end{cases}. \tag{6.2}$$

Table I. Mass properties of each body.

	l_0	l_1	l_2	l_3	l_4	l_5	l_6
m (kg)	500	4.239*	6.3585	21.195	21.195	6.3585	4.239
l (m)	4	0.2	0.3	1	1	0.3	0.2
r (m)	2	0.05**	0.05	0.05	0.05	0.05	0.05
$I_{xx}\ (\mathrm{kg/m^2})$	1166.7	0.168	0.0517	1.7804	1.7804	0.0517	0.0168
$I_{yy}\ (\mathrm{kg/m^2})$	1166.7	0.168	0.0517	1.7804	1.7804	0.0517	0.0168
$I_{zz}\ (\mathrm{kg/m^2})$	166.7	0.0009	0.0013	0.0044	0.0044	0.0013	0.0009

*In addition to the base, the density of links is $\rho = 2.7 \times 10^3 \mathrm{kg/m^3}$ and the radius of links is $r = 0.05m$.

**All the links are regarded as cylinders with their centers of mass coinciding with their geometric centers and neglecting gravity.

Table II. D–H parameters of the whole system.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	$l_0/2 + l_1$	$90^\circ + \theta_1$
2	-90°	0	0	θ_2
3	90°	0	$l_2 + l_3$	θ_3
4	-90°	0	0	θ_4
5	0	l_4	0	θ_5
6	-90°	0	0	θ_6
e	0	0	$l_5 + l_6$	0

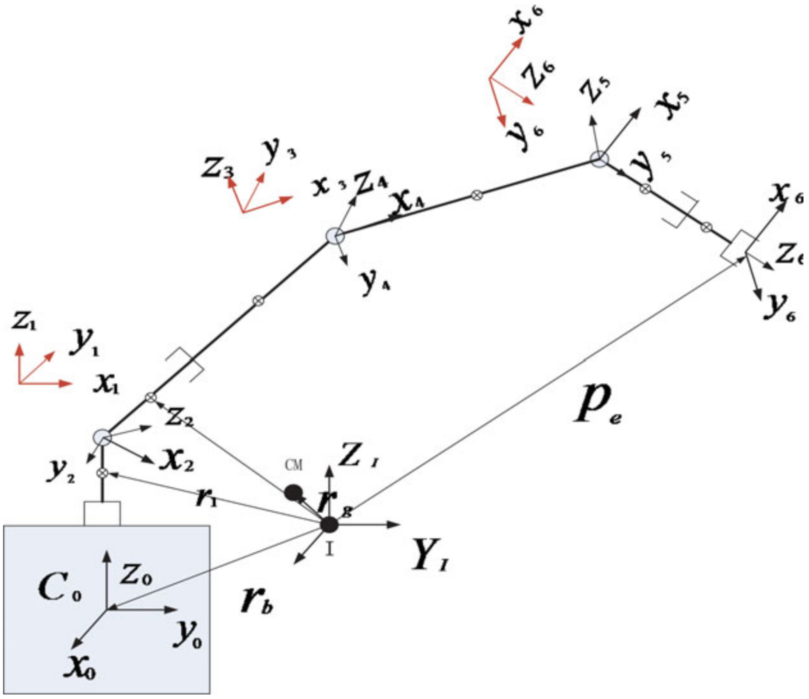

Fig. 2. Modeling of 6-DOF space manipulators.

The initial system state and end-effector position are as follows:

$$\begin{cases} \begin{bmatrix} \boldsymbol{P}_{b0} & \boldsymbol{Q}_{b0} \end{bmatrix} = \begin{bmatrix} 0 & -0.3333 & -3.0966 & 1 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{P}_{e0} = \begin{bmatrix} 0 & 0.6666 & -0.0966 \end{bmatrix} \end{cases}. \tag{6.3}$$

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

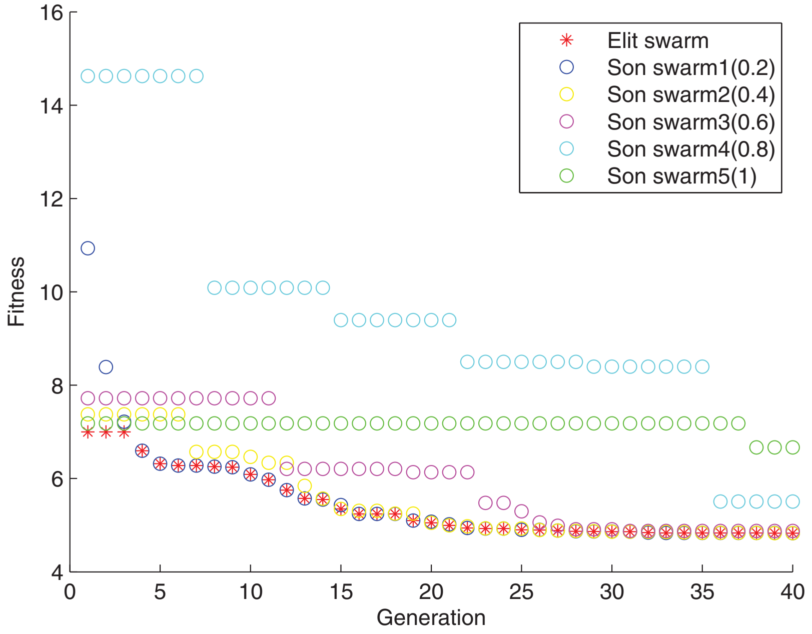

Fig. 3. Variation in the best fitness of every swarm.

The desired system state and end-effector position are as follows:

$$\begin{cases} \begin{bmatrix} \boldsymbol{P}_{bd} & \boldsymbol{Q}_{bd} \end{bmatrix} = \begin{bmatrix} 0 & -0.3333 & -3.0966 & 1 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{P}_{ed} = \begin{bmatrix} 0.6981 & 1.1462 & 0.1714 \end{bmatrix} \end{cases}. \tag{6.4}$$

The goal of planning is that the position and attitude of the base remain almost unchanged when the end-effector moves from $\boldsymbol{P}_{e0}$ to $\boldsymbol{P}_{ed}$. For this reason, the final base attitude was set equal to the initial value. In the discussion above, the units of distance are m, and the quaternion has no units.

6.2. *Simulation results*

The algorithm parameters are given below.

$N_e = 40$, $N_s = 40$ number of individuals in the elite swarm and each child swarm respectively;

$nS = 5$ number of child swarms;

$c_1 = 2.05, c_2, 2.05, c_3 = 2$ learning factors, of which the first two learning factors of the elite swarms are shared with the child swarms;

$\boldsymbol{\omega}_n = \begin{bmatrix} 0.2 & 0.4 & 0.6 & 0.8 & 1 \end{bmatrix}$ set of inertial weights of the child swarms;

$M = 40$ number of generations;

$D = 12$ dimension of the algorithm;

$\delta = 0.05$ mutagenic factor.

Figure 3 shows the relationship between the value of the objective function and the number of generations.

The optimal ***aa*** and fitness values are as follows:

$$\begin{cases} \boldsymbol{a} = [-0.1754 \quad 0.2040 \quad 0.0322 \quad 0.1530 \\ \qquad 0.2502 \quad 0.1389 \quad -0.0579 \quad 0.0757 \\ \qquad 0.0480 \quad 0.2341 \quad -0.1018 \quad -0.0546] \times 10^{-5} \\ F(\boldsymbol{a}) = 4.8309 \end{cases}. \tag{6.5}$$

Figures 4 and 5 show the curves of base position and attitude, respectively.

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

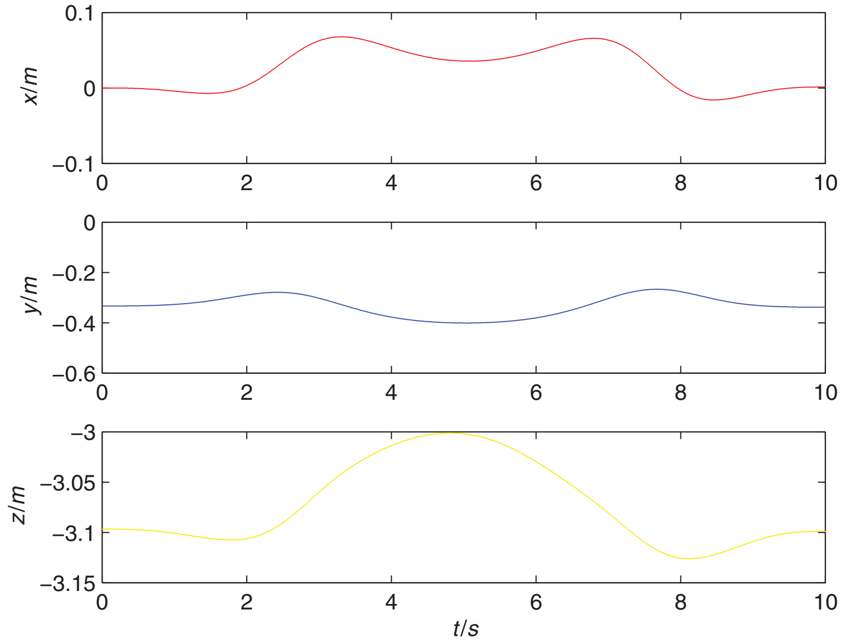

Fig. 4. Base position curves.

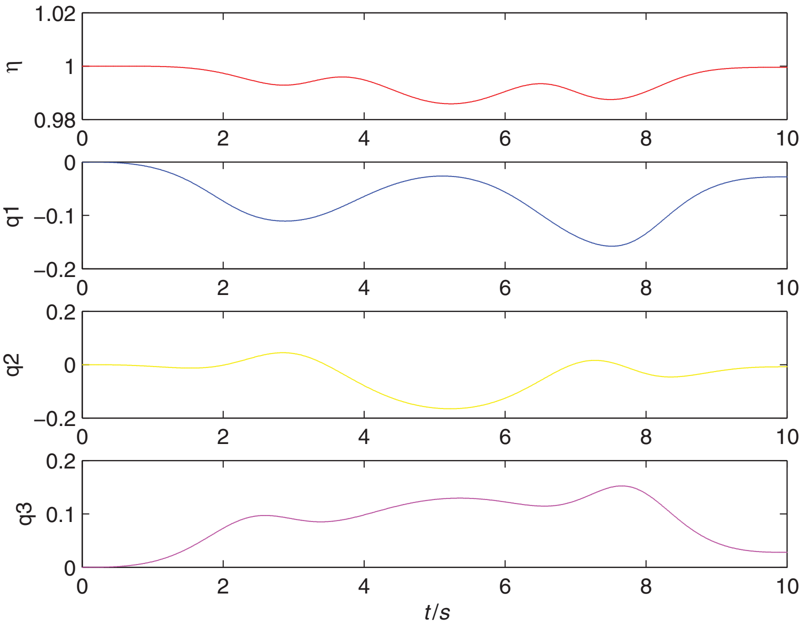

Fig. 5. Base attitude curves.

Figure 6 shows the planned joint trajectories, illustrating that the base disturbance has been minimized. Figures 7 and 8 show the rate curves and the joint acceleration curves, respectively.

After planning, the system state is as follows:

$$\begin{cases} \begin{bmatrix} \boldsymbol{P}_{bf} & \boldsymbol{Q}_{bf} \end{bmatrix} = \begin{bmatrix} 0.0013 & -0.3380 & -3.0989 & 0.9998 & -0.0123 & -0.0039 & 0.0140 \end{bmatrix} \\ \boldsymbol{P}_{ef} = \begin{bmatrix} 0.1423 & 0.5711 & 0.1051 \end{bmatrix} \end{cases}. \tag{6.6}$$

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

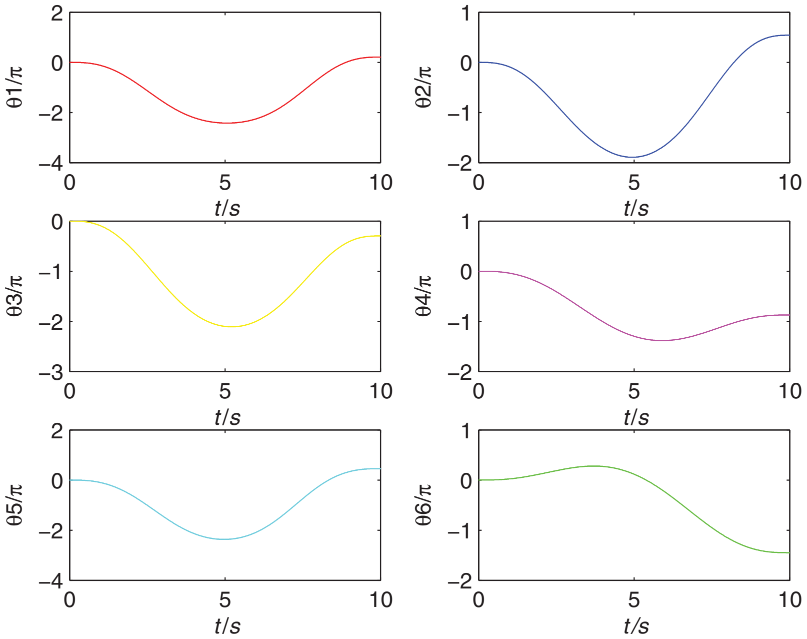

Fig. 6. Planned joint angles.

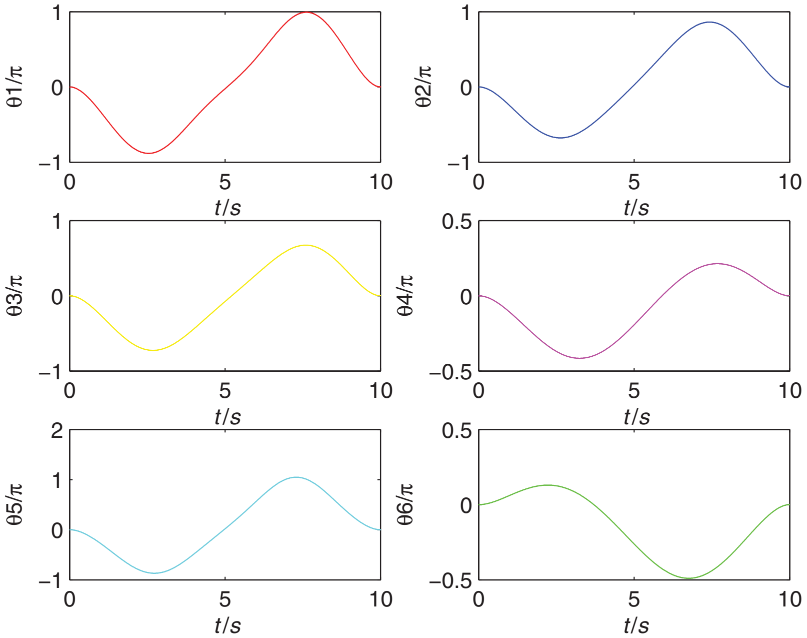

Fig. 7. Joint velocity curves.

The errors in base position and attitude are as follows:

$$\begin{bmatrix} \boldsymbol{P}_{b\_\text{error}} & \boldsymbol{Q}_{b\_\text{error}} \end{bmatrix} = \begin{bmatrix} 0.0013 & -0.0047 & -0.0023 & 0.0002 & -0.0123 & -0.0039 & 0.0140 \end{bmatrix}. \tag{6.7}$$

Figure 9 shows the end-effector path, where the green asterisk on the end-point of the path is the starting point of the end-effector before planning. The other end-point of the path represents the termination of planning. The red asterisk, the lower of the two nearly coincident asterisks, is the desired destination point of the end-effector, and the remaining, pink, asterisks represent the planning

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

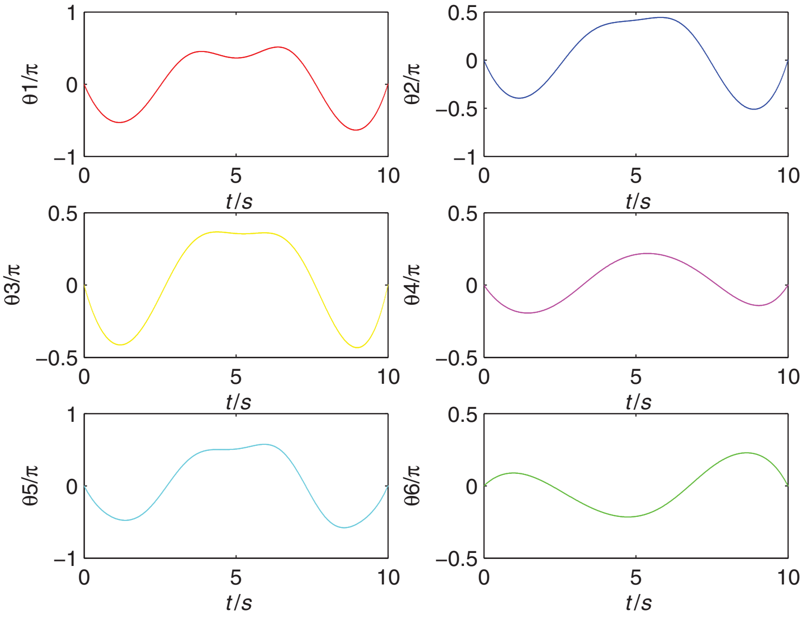

Fig. 8. Joint acceleration curves.

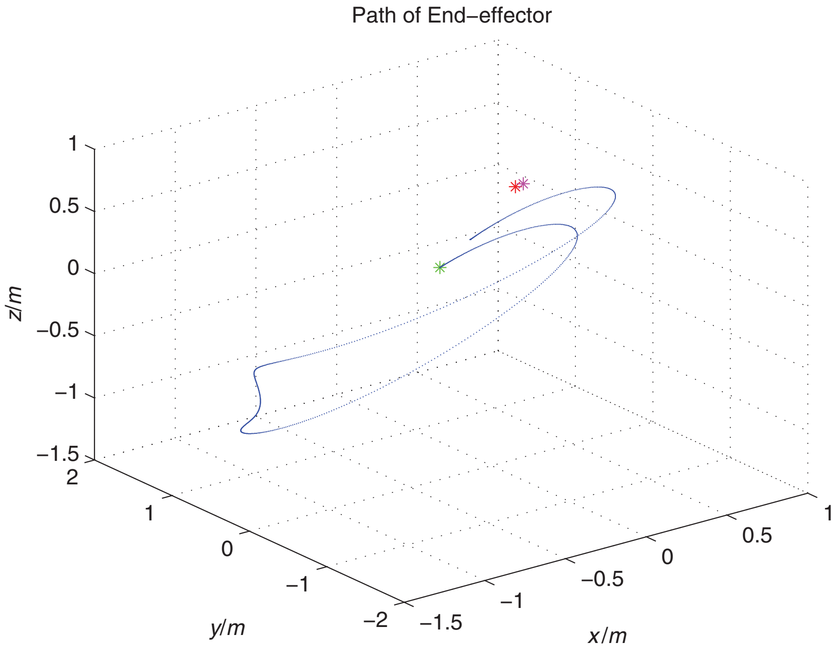

Fig. 9. End-effector path.

termination point after error correction. Because tiny errors in the base can become magnified by the length of the manipulators and affect the end-effector position, the error in Eq. (6.7) must be corrected. The end-effector position after correction is as follows:

$$\boldsymbol{P}_{em} = \begin{bmatrix} 0.7325 & 1.1171 & 0.1915 \end{bmatrix}. \tag{6.8}$$

The total error of the system is as follows:

$$\boldsymbol{P}_{e\_\text{error}} = \begin{bmatrix} 0.0344 & -0.0291 & 0.0201 \end{bmatrix}. \tag{6.9}$$

Here, $\boldsymbol{P}_{e\_\text{error}}$ is produced by the process of calculation.

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

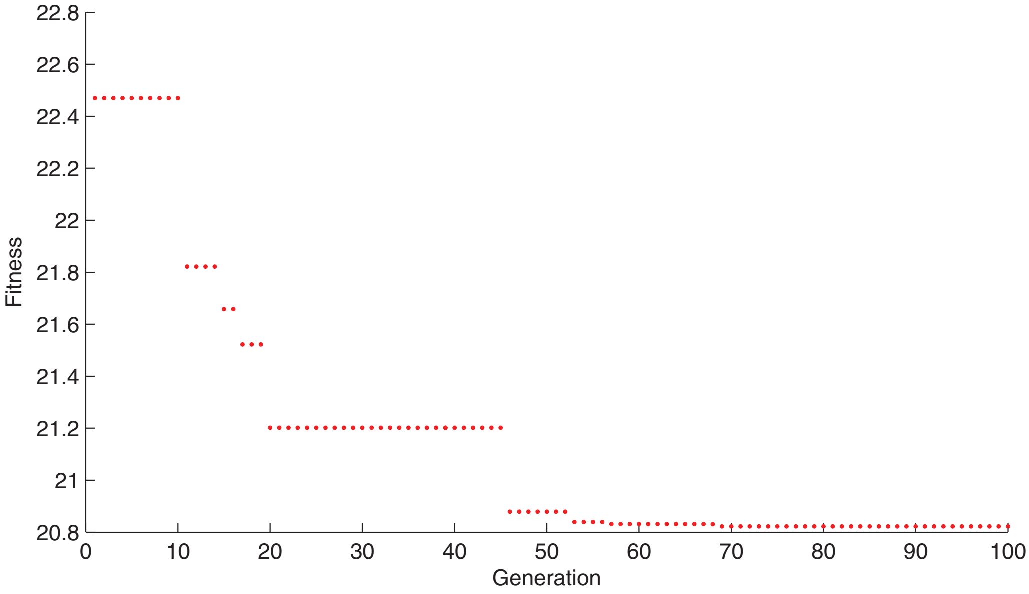

Fig. 10. Variation of the fitness.

As shown in Figs. 4 and 5, it is clear that base disturbances during manipulator operation cannot be ignored. The aim of motion planning to minimize base disturbance is to limit disturbances of the initial and final states to a minimum. To compensate for the tiny remaining error, it is possible to install orthogonal reaction wheels, which consume only a small amount of fuel, on the base instead of controlling the attitude by thrusters in real time.

6.3. Result of SimuAPS for comparison

The parameters of algorithm:

$$\begin{bmatrix} N & c_1 & c_2 & \lambda & M & D \end{bmatrix} = \begin{bmatrix} 40 & 2.05 & 2.05 & 0.5 & 100 & 12 \end{bmatrix} \tag{6.10}$$

N	number of individuals in group;
c_1, c_2	learning factors;
λ	annealing constant;
M	number of generation;
D	dimension of algorithm.

Figure 10 shows the relationship between the value of objective function and the number of generation.

The optimal ***aa*** and the value of fitness are as follows:

$$\begin{cases} \boldsymbol{a} = [-0.0707 \quad -0.0324 \quad -0.0270 \quad 0.0961 \\ \qquad -0.1664 \quad 0.1770 \quad -0.0072 \quad 0.0147 \\ \quad 0.0536 \quad -0.1915 \quad 0.0754 \quad 0.0253] \times 10^{-5} \\ \qquad\qquad F(\boldsymbol{a}) = 20.8224 \end{cases} . \tag{6.11}$$

It can be noticed that the fitness is an invariable value after generation 70, while the value of the presented method is around 30 shown in Fig. 3, and the output value *F(a)* is 20.8224 while the value of the presented method is 4.8309, as shown in Formula (6.5). The results show that the presented method is more efficient and accurate.

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

7. Conclusions

In this paper, a 6-DOF space manipulator has been modeled, and a plan for minimizing base disturbances due to manipulator motion has been presented. The objective function has a number of extreme points due to the complexity of the motion equations of 6-DOF space manipulators. In particular, several sub-optimal solutions are located close to the global optimum. General algorithms have difficulty searching for the best result, even the Simu-APSO algorithm, which can search globally and locally at the same time. To deal with this problem more effectively, a new multi-swarm PSO algorithm has been designed. From the results described above, it is evident that the proposed algorithm can converge rapidly on a better solution than those found using Simu-APS algorithms and that the new algorithm only needs a few generations to find the optimum.

The main shortcoming of this algorithm is that its convergence time cannot satisfy the requirements of real-time planning. However, thanks to the parallelism of the child swarms, it is possible to use a multi-core CPU (or even a GPU) to solve the objective function more efficiently.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 61425002), the Program for Liaoning Innovative Research Team in University (No. LT2015002), and by the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University.

References

1. P. F. Huang, Y. S. Xu and B. Liang, "Dynamic balance controlof multi-arm free-floating space robots," *Int. J. Adv. Robot. Syst.* **2**, 117–125 (2005).
2. Y. Nakamura and R. Mukherjee, "Nonholonomic path planning of space robots via a bidirectional approach," *IEEE Trans. Robot. Autom.* **7**, 500–514 (1991).
3. E. Papadopoulos, I. Papadimitriou and I. Poulakakis, "Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems," *Robot. Auton. Syst.* **51**, 229–247 (2005).
4. W. Xu, B. Liang, C. Li, W. Qiang, Y. Xu and K. K. Lee, "Non-Holonomic Path Planning of Space Robot based on Genetic Algorithm," *IEEE International Conference on*, *Robotics and Biomimetics ROBIO'06*, Kunming, China, (2006) pp. 1471–1476.
5. S. Dubowsky, E. E. Vance and M. A. Torres, "The Control of Space Manipulators Subject to Spacecraft Attitude Control Saturation Limits," *Proceedings of NASA Conference Space Telerobotics*, Cambridge, MA, (1989) pp. 409–418.
6. Z.Vafa and S. Dubowsky, "Minimization of Spacecraft Disturbances in Space Robotic Systems," *Advances in the Astronautical Science*, Cambridge, MA, (1988) pp. 91–108.
7. S. Dubowsky and M. Torres, "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," *IEEE International Conference on Robotics and Automation*, Cambridge, MA, USA, (1991) pp. 2522–2528.
8. E. Papadoupolous and S. Dubowsky, "Coordinated Manipulator/Spacecraft Motion Control for Space Robotic Systems," *IEEE International Conference on Robotics and Automation*, Cambridge, MA, (1991) pp. 1696–1701.
9. D. Nenchev, K. Yoshida and Y. Umetani, "Analysis, Design and Control of Free-Flying Space Robots Using Fixed-Attitude-Restricted Jacobian Matrix[C]," The fifth international symposium on Robotics research. MIT Press, (1991) pp. 251–258.
10. K. Yoshida, K. Hashizume and S. Abiko, "Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm," *Proceedings of the IEEE International Conference on Robotics and Automation*, Piscataway, USA, (2001) pp. 441–446.
11. K. Yoshida, R. Kurazume and Y. Umetani, "Dual Arm Coordination in Space Free-Flying Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, (1991) pp. 2516–2521.
12. K. Yamada, "Arm Path Planning for a Space Robot," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, (1993) pp. 2049–2055.
13. T. Suzuki and Y. Nakamura, "Planning Spiral Motion of Nonholonomic Space Robots," *Proceedings of the IEEE International Conferences on Robotics and Automation*, Minneapolis, Minnesota, (1996) pp. 718–725.
14. C. H. Wang, B. M. Feng and G. C. Ma, "Robust Tracking Control of Space Robots using Fuzzy Neural Network," *Proceedings IEEE International Symposiumon Computational Intelligence in Robotics and Automation*, Espoo, (2005) pp. 181–185.
15. K. Senda, Y. Murotsu and M. Ozaki, "A method of attitude control for space robots: An approach using a neural network," *Trans. Japan Soc. Mech. Eng. Ser. C* **57**, 2356–2362 (1991).

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press

16. T. Akiyama and Y. Sakawa, "Path planning of space robots by using nonlinear optimization technique," *Trans. Soc. Instrum. Control Eng.* **31**, 193–197 (1995).
17. B. Zappa, G. Legnani and R. Adamini, "Path planning of free-flying space manipulators: An exact solution for polar robots," *Mech. Mach. Theory* **40**, 806–820 (2005).
18. I. Belousov, C. Estevs, J. Laumond and E. Ferré, "Motion Planning for the Large Space Manipulators with Complicated Dynamics," *Proceeding IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, (2005) pp. 3713–3719.
19. K. Yamada, S. Yoshikawa and Y. Fujita, "Arm path planning of a space robot with angular momentum," *Adv. Robot.* **9**, 693–709 (1995).
20. D. N. Nenchev, "Controller for A Redundant Free-flying Space Robot with Spacecraft Atitude/Manipulator Motion Coordination," *Proceedings International Conference on Intelligent Robots and Systems*, Yokohama, (1993) pp. 2108–2114.
21. A. Akbarimajd, "Optimal motion planning of juggling by 3-DOF manipulators using adaptive PSO algorithm," *Robotica* **32**, 967–984 (2014).
22. A. Rana and A. Zalzala, "Near time-optimal collision-free motion planning of robotic manipulators using an evolutionary algorithm," *Robotica* **14**, 621–632 (1996).
23. G. G. Rigatos, "Distributed gradient and particle swarm optimization for multi-robot motion planning," *Robotica* **26**, 357–370 (2008).
24. S. Wang, J. Bao and Y. Fu, "Real-time motion planning for robot manipulators in unknown environments using infrared sensors," *Robotica* **25**, 201–211 (2007).
25. Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized jacobian matrix," *IEEE Trans. Robot. Autom.* **5**, 303–314 (Jun. 1989).
26. Z. Shuzhen, H. Jianlong, K. Minxiu and S. Lining, "Inverse Kinematics and Application of a Type of Motion Chain Based on Screw Theory and Analytic Geometry," *IEEE International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, China, (2010), pp. 680–684.
27. J. Xie, W. Qiang, B. Liang and C. Li, "Inverse Kinematics Problem for 6-DOF Space Manipulator Based On the Theory of Screws," *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*, Sanya, China, (Dec. 15–18, 2007) pp. 1659–1663.
28. J. C. K. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Trans. Robot. Autom.* **8**, 53–64 (Feb. 1992).
29. J. S. C. Yuan, "Closed-loop manipulator control using quaternion," *IEEE J. Robot. Autom.* **4**, 434–440 (Aug. 28, 1988).
30. W. Xu, B. Liang, C. Li, Y. Xu and W. Qiang, "Path planning of free floating robot in cartesian space using direct kinematics," *Int. J. Adv. Robot. Syst.* **4**, 17–26 (2007).
31. F. Zhang, Y. Fu, L. Hua, H. Chen, S. Wang and B. Guo, "Point-to-point Planning for Free-floating Space Manipulator with Zero-disturbance Spacecraft Attitude," *Proceeding of the IEEE International Conference on Information and Automation*, Shenyang, China, (Jun, 2012) pp. 142–147.
32. J. Wu, S. Shi, B. Wang, Z. Jiang and H. Liu, "Path Planning for Minimizing BaseReaction of Space Robot and its GroundExperimental Study," *Proceedings of the 2009 IEEE International Conferenceon Mechatronics and Automation*, Changchun, China, (Aug. 9–12, 2009) pp. 4627–4632.
33. J. Kennedy, "Particle Swarm Optimization," **In:** *Encyclopedia of Machine Learning*, Kazuya Yoshida (eds.) (Springer, 2010) pp. 760–766.
34. X. Huo, Y. Cheng, Y. Wang and Q. Hu, "PSO based Trajectory Planning of Free Floating Space Robot for Minimizing Spacecraft Attitude Disturbance," *IEEE Chinese Control and Decision Conference (CCDC)*, Mianyang, (2011) pp. 819–822.
35. P. Huang and Y. Xu, "PSO-Based Time-Optimal Trajectory Planning for Space Robot with Dynamic Constraints," *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics*, Kunming, (Dec. 17–20, 2006) pp. 1402–1407.
36. W. Xu, C. Li, B. Liang, Y. Liu and Y. Xu, "The cartesian path planning of free-floating space robot using particle swarm optimization," *Int. J. Adv. Robot. Syst.* **5**, 301–310 (2008).
37. Z. Qiang and G. Fang, "A Two-Phase Multi-population Particle Swarm Algorithm for Global Optimization," *Proceedings of 2006 International Conference on artificial Intelligence - 50 Years' Achievements, Future Directions and Social Impacts*, Beijing, (Aug. 2006) pp. 271–274.
38. M. Shimizu, T. Hiraoka, K. Fujishima, A. Kurosu, K. Hashizume and K. Yoshida, "The Spacedyn: A MATLAB Toolbox for Space and Mobile Robots," http://www.astro.mech.tohoku.ac.jp/spacedyn/, (1999).

https://doi.org/10.1017/S0263574715000855 Published online by Cambridge University Press