Acta Astronautica 112 (2015) 77–88

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Trajectory planning of free-floating space robot using Particle Swarm Optimization (PSO)

Mingming Wang [a,b,*], Jianjun Luo [b], Ulrich Walter [a]

[a] Institute of Astronautics, Technical University of Munich, 85748 Garching, Germany
[b] Science and Technology on Aerospace Flight Dynamics Laboratory, Northwestern Polytechnical University, 710072 Xi'an, China

ARTICLE INFO

Article history:
Received 4 November 2014
Accepted 5 March 2015
Available online 12 March 2015

Keywords:
Trajectory planning
Free-floating
Bézier curve
Particle Swarm Optimization

ABSTRACT

This paper investigates the application of Particle Swarm Optimization (PSO) strategy to trajectory planning of the kinematically redundant space robot in free-floating mode. Due to the path dependent dynamic singularities, the volume of available workspace of the space robot is limited and enormous joint velocities are required when such singularities are met. In order to overcome this effect, the direct kinematics equations in conjunction with PSO are employed for trajectory planning of free-floating space robot. The joint trajectories are parametrized with the Bézier curve to simplify the calculation. Constrained PSO scheme with adaptive inertia weight is implemented to find the optimal solution of joint trajectories while specific objectives and imposed constraints are satisfied. The proposed method is not sensitive to the singularity issue due to the application of forward kinematic equations. Simulation results are presented for trajectory planning of 7 degree-of-freedom (DOF) redundant manipulator mounted on a free-floating spacecraft and demonstrate the effectiveness of the proposed method.

1. Introduction

The increasing demands of satellite maintenance, on-orbit assembly and space debris removal, etc. call for application of space robot to perform tasks in the particular harsh space environment. Examples include "Robot Technology Experiment (ROTEX)" [1], "Engineering Test Satellite VII (ETS-VII)" [2] and "Orbital Express (OE)" [3]. In light of the space robots currently planned by world wild space agencies, an increase in the number and the capacity of robot applied in space missions will be a foregone conclusion in the coming future [4]. In these space robotic programs, space missions executed under free-floating mode are of great interests to the researchers. However, space robot exhibits some special characteristics due to the dynamic coupling between the space manipulators and the spacecraft (base). Accordingly, particular trajectory planning techniques have to be developed to cope with the dynamic coupling issue of free-floating space robot.

In order to address the continuous path tracking issue of the end-effector, the concept of Generalized Jacobian Matrix (GJM) [5], Path Independent Workspace (PIW) [6], bidirectional path planning method [7] and Enhanced Disturbance Map (EDM) [8] was introduced for free-floating space robot. It is shown that the system states depend not only on the joint variables but also on the history of their trajectories. This introduces the so-called non-holonomic redundancy to the space robot. Another widely investigated method was Reaction Null-Space (RNS) as proposed in [9–11]. But the volume of RNS is

* Corresponding author at: Institute of Astronautics, Technical University of Munich, 85748 Garching, Germany.
Tel.: +49 89289 15996; fax: +49 89289 16004.
E-mail addresses: m.wang@tum.de (M. Wang), jjluo@nwpu.edu.cn (J. Luo), walter@tum.de (U. Walter).

http://dx.doi.org/10.1016/j.actaastro.2015.03.008
0094-5765/

limited especially for 6 DOF manipulators. Besides, trajectory planning of free-floating space robot can also be treated as an optimization issue. Various searching methods like variational approach [12], Genetic Algorithm (GA) [13], Particle Swarm Optimization (PSO) [14–16] and Sequential Quadratic Programming (SQP) [17] were employed to search the optimal solution to steer the end-effector of a free-floating space robot to a target pose.

This paper introduces a new method for trajectory planning issue of kinematically redundant manipulator while cope with joint range, velocity, acceleration limits with different objectives. The reason for choosing kinematically redundant manipulator is the existence of infinite solutions which can be employed for additional objectives, such as minimize base disturbance, avoid collision, or maximize the manipulability. In order to perform optimization, the joint trajectory is generally parametrized by polynomial functions [14,15] or B-Spline [13,17], however, it is not easy for above curves to deal with the imposed constraints. In this paper, the Bézier curve for its simplicity and normalization is used to represent the shape of joint trajectories and limit the values of joint range, rate, acceleration. PSO with adaptive inertia weight and various fitness functions and constraints is implemented to find the optimal solution for trajectory planning of free-floating space robot.

The rest of this paper is organized as follows: Section 2 introduces the kinematics and dynamics of free-floating space robotic system. Section 3 discusses the path planning issue and how to delineate this issue as an optimization problem. Section 4 depicts the concepts of PSO, fitness selection and constraints handling. Section 5 shows the simulation results of trajectory planning using PSO applied to a 7 DOF space robot. The conclusive remarks are listed in Section 6.

2. Modelling of free-floating space robot

2.1. Kinematics and dynamics of space robot

Before discussion space robot in detail, some symbols and variables used in the following sections are listed in Table 1. A space robotic system is composed of a spacecraft and an n DOF manipulator, in total $n+1$ bodies as shown in Fig. 1. Many investigations have been conducted in the field of space robot dynamics. Refer to [9,11], the dynamics equations of space robot using Lagrangian mechanism can be expressed as follows:

$$\begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{bm} \\ \mathbf{H}_{bm}^{\mathrm{T}} & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_b^{\mathrm{T}} \\ \mathbf{J}_e^{\mathrm{T}} \end{bmatrix} \boldsymbol{f}_e \tag{1}$$

When no external force vector applied on the end-effector, i.e. $\boldsymbol{f}_e = \mathbf{0}$, and no active actuators are applied on the base, i.e. $\boldsymbol{f}_b = \mathbf{0}$, then the system is called a free-floating space robot. According to the angular momentum conservation law, the total momentum $\boldsymbol{L}_0$ around the mass center of the whole system is conserved in the free-floating mode, which can be expressed by

$$\boldsymbol{L}_0 = \mathbf{H}_b \dot{\boldsymbol{x}}_b + \mathbf{H}_{bm} \dot{\boldsymbol{\theta}} \tag{2}$$

Suppose the initial momentum $\boldsymbol{L}_0 = \mathbf{0}$, since $\mathbf{H}_b$ is invertible, the motion of the base can be described by

$$\dot{\boldsymbol{x}}_b = \begin{bmatrix} \dot{\boldsymbol{r}}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = \mathbf{J}_a \dot{\boldsymbol{\theta}} = -\mathbf{H}_b^{-1} \mathbf{H}_{bm} \dot{\boldsymbol{\theta}} \tag{3}$$

By substituting Eq. (3) into the kinematic mapping of the end-effector, $\dot{\boldsymbol{x}}_e = \mathbf{J}_b \dot{\boldsymbol{x}}_b + \mathbf{J}_e \dot{\boldsymbol{\theta}}$, the motion of the end-effector is given as follows:

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} \dot{\boldsymbol{r}}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_g \dot{\boldsymbol{\theta}} = \left(\mathbf{J}_e - \mathbf{J}_b \mathbf{H}_b^{-1} \mathbf{H}_{bm} \right) \dot{\boldsymbol{\theta}} \tag{4}$$

where $\mathbf{J}_g$ is termed Generalized Jacobian Matrix (GJM) which is derived in [5]. From Eqs. (3) and (4), one can see that with appropriate design of joint trajectories, the motion of the base and the end-effector is interactional because of the dynamics coupling effect between the manipulator and the base.

2.2. Orientation representation

In practice, one can designate Euler angles, angle/axis, or unit quaternion as orientation representation [18] according to different application background. However, for a redundant manipulator with dexterous workspace, orientation representation like Euler angles or angle/axis is not sufficient since the existence of orientation singularity. For instance, if the orientation is described by Z–Y–X Euler

Table 1
Kinematic and dynamic symbols used in the paper.

Symbols	Representation
J_i, C_i	Joint i and mass center of link i
$\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{R}^3$	Position vector from J_i to C_i and from C_i to J_{i+1}
$\boldsymbol{r}_{C_i} \in \mathbb{R}^3$	Position vector of mass center of link i
$\boldsymbol{r}_b, \boldsymbol{r}_e \in \mathbb{R}^3$	Position vector of base and end-effector
$\boldsymbol{\omega}_b, \boldsymbol{\omega}_e \in \mathbb{R}^3$	Angular velocity of base and end-effector
$m_i \in \mathbb{R}, \mathbf{I}_i \in \mathbb{R}^{3\times3}$	Mass and inertia matrix of link i
$\mathbf{H}_b \in \mathbb{R}^{6\times6}$	Inertia matrix of the base
$\mathbf{H}_{bm} \in \mathbb{R}^{6\times n}$	Coupling inertia matrix
$\mathbf{H}_m \in \mathbb{R}^{n\times n}$	Inertia matrix of the manipulator
$\boldsymbol{c}_b \in \mathbb{R}^6, \boldsymbol{c}_m \in \mathbb{R}^n$	Velocity dependent non-linear terms
$\boldsymbol{f}_b, \boldsymbol{f}_e \in \mathbb{R}^6$	Force and moment exert on base and end-effector
$\boldsymbol{\tau} \in \mathbb{R}^n$	Torque exert on manipulator joints

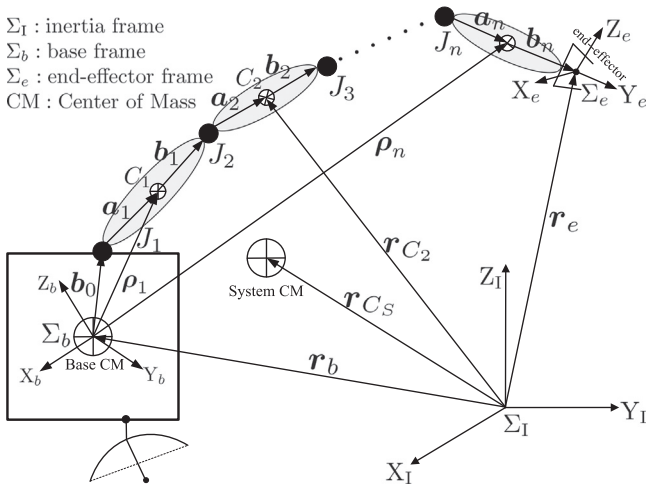

Fig. 1. Schematic diagram of space robot.

angles $\boldsymbol{\phi}=[\alpha,\beta,\gamma]$, the relation between rotational velocity $\dot{\boldsymbol{\phi}}$ and angular velocity $\boldsymbol{\omega}$ is

$$\boldsymbol{\omega}=\begin{bmatrix}0 & -\sin\alpha & \cos\alpha\cos\beta\\ 0 & \cos\alpha & \sin\alpha\cos\beta\\ 1 & 0 & -\sin\beta\end{bmatrix}\dot{\boldsymbol{\phi}}=\mathbf{J}_{\phi}\dot{\boldsymbol{\phi}} \tag{5}$$

The determinant of matrix $\mathbf{J}_{\phi}$ is $-\cos(\beta)$, which indicates that $\mathbf{J}_{\phi}$ is not invertible for $\beta=\pm\pi/2$. This means at such pose the angular velocity $\boldsymbol{\omega}$ cannot be expressed by means of $\dot{\boldsymbol{\phi}}$ and infinite rotational velocities $(\dot{\alpha},\dot{\beta},\dot{\gamma})$ are required. This phenomenon is termed as representation singularity of $\boldsymbol{\phi}$.

The drawback of representation singularity can be overcome by different four parameters, namely, the unit quaternion. Define a quaternion $\boldsymbol{q}=\{\eta,\boldsymbol{\varepsilon}\}\in\mathbb{R}^4=\cos\varphi/2+\boldsymbol{r}\cdot\sin\varphi/2$, η is the scalar part and $\boldsymbol{\varepsilon}$ is the vector part of the quaternion. $\boldsymbol{r}$ and φ are the unit vector and the rotation angle along the vector $\boldsymbol{r}$, respectively. The quaternion $\boldsymbol{q}$ is constructed by $\eta^2+\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}=1$. Note that $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same orientation. If $\boldsymbol{q}_1=\{\eta_1,\boldsymbol{\varepsilon}_1\}$ and $\boldsymbol{q}_2=\{\eta_2,\boldsymbol{\varepsilon}_2\}$ denote the quaternions corresponding to two rotation frames, the relative quaternion can be calculated as follows where $*$ is the quaternion production operator:

$$\{\delta\eta,\delta\boldsymbol{\varepsilon}\}=\boldsymbol{q}_1*\boldsymbol{q}_2^{-1}=\{\eta_1\eta_2+\boldsymbol{\varepsilon}_1^{\mathrm{T}}\boldsymbol{\varepsilon}_2,\eta_2\boldsymbol{\varepsilon}_1-\eta_1\boldsymbol{\varepsilon}_2-\tilde{\boldsymbol{\varepsilon}}_1\boldsymbol{\varepsilon}_2\} \tag{6}$$

where $\tilde{\boldsymbol{\varepsilon}}$ is defined as

$$\tilde{\boldsymbol{\varepsilon}}=\begin{bmatrix}0 & -\varepsilon_z & \varepsilon_y\\ \varepsilon_z & 0 & -\varepsilon_x\\ -\varepsilon_y & \varepsilon_x & 0\end{bmatrix} \tag{7}$$

Note that when $\delta\boldsymbol{\varepsilon}=\mathbf{0}$ it implies that the two rotational frames coincide. The relationship between the angular velocity $\boldsymbol{\omega}$ and the time derivative of the quaternion $\boldsymbol{q}$ is

$$\begin{bmatrix}\dot{\eta}\\ \dot{\boldsymbol{\varepsilon}}\end{bmatrix}=\frac{1}{2}\begin{bmatrix}-\boldsymbol{\varepsilon}^{\mathrm{T}}\\ \eta\mathbf{E}_3-\tilde{\boldsymbol{\varepsilon}}\end{bmatrix}\boldsymbol{\omega}=\mathbf{J}_q\boldsymbol{\omega} \tag{8}$$

3. Trajectory planning of space robot

The objective of trajectory planning is to generate applicable joint motion laws without violating the imposed constraints to complete the desired manipulators tasks [19]. Generally, a trajectory planning algorithm should have the following features:

- The specified objectives should be optimized under feasible regions.
- The continuity of the joint position, velocity should be guaranteed.
- Undesirable effects during motion should be minimized.

3.1. Problem description

The trajectory planning issue for free-floating space robot is investigated here. Not like manipulator with fixed base, non-holonomic characteristic of free-floating space robot should be taken into account in trajectory planning, which implies that the final pose of the space robot not only relies on its inverse kinematics, but also depends on the dynamics coupling effect. Suppose $\boldsymbol{x}_e=(\boldsymbol{r}_e,\boldsymbol{q}_e)$ and $\boldsymbol{x}_b=(\boldsymbol{r}_b,\boldsymbol{q}_b)$ represent the pose of the end-effector and the base, respectively. The pose $\boldsymbol{x}$ with upper scripts s, f, d stands for the start, final and desired pose, respectively, according to Eqs. (3) and (4), the final pose of base and end-effector can be calculated as follows:

$$\boldsymbol{x}_b^f=\begin{bmatrix}\boldsymbol{r}_b^f\\ \boldsymbol{q}_b^f\end{bmatrix}=\int_{t_s}^{t_f}\begin{bmatrix}\mathbf{E}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_q(\boldsymbol{q}_b)\end{bmatrix}\mathbf{J}_a\dot{\boldsymbol{\theta}}\,dt \tag{9}$$

$$\boldsymbol{x}_e^f=\begin{bmatrix}\boldsymbol{r}_e^f\\ \boldsymbol{q}_e^f\end{bmatrix}=\int_{t_s}^{t_f}\begin{bmatrix}\mathbf{E}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_q(\boldsymbol{q}_e)\end{bmatrix}\mathbf{J}_g\dot{\boldsymbol{\theta}}\,dt \tag{10}$$

The relative difference between the final pose and the desired pose both for base and end-effector is given by

$$\delta\boldsymbol{x}_b=\boldsymbol{x}_b^f-\boldsymbol{x}_b^d=\begin{bmatrix}\boldsymbol{r}_b^f-\boldsymbol{r}_b^d\\ \eta_b^d\boldsymbol{\varepsilon}_b^f-\eta_b^f\boldsymbol{\varepsilon}_b^d-\tilde{\boldsymbol{\varepsilon}}_b^f\boldsymbol{\varepsilon}_b^d\end{bmatrix} \tag{11}$$

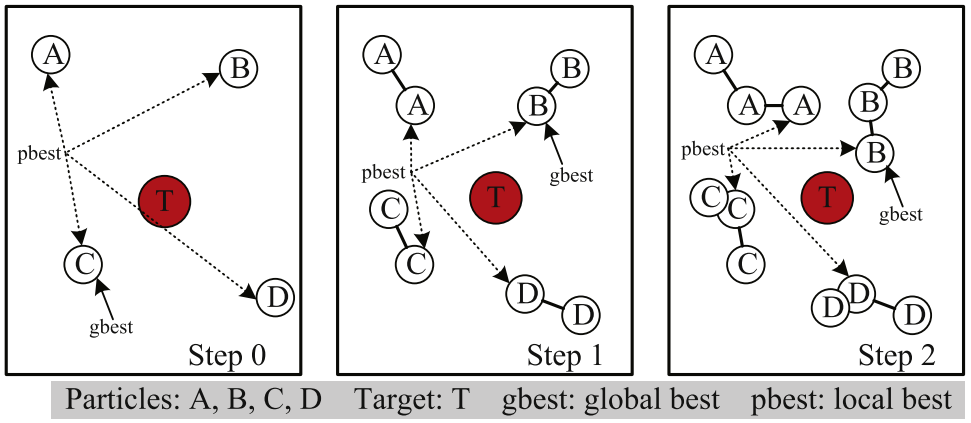

Fig. 2. Schematic diagram of PSO with 4 particles.

$$\delta \boldsymbol{x}_e = \boldsymbol{x}_e^f - \boldsymbol{x}_e^d = \begin{bmatrix} \boldsymbol{r}_e^f - \boldsymbol{r}_e^d \\ \eta_e^d \boldsymbol{\varepsilon}_e^f - \eta_e^f \boldsymbol{\varepsilon}_e^d - \tilde{\boldsymbol{\varepsilon}}_e^f \boldsymbol{\varepsilon}_e^d \end{bmatrix} \tag{12}$$

The solution of trajectory planning problem is to generate appropriate motion laws for each joint yielding the following equality and inequality constraints:

$$\begin{cases} \boldsymbol{x}_b(t_s) = \boldsymbol{x}_b^s, \boldsymbol{x}_e(t_s) = \boldsymbol{x}_e^s, \boldsymbol{\theta}(t_s) = \boldsymbol{\theta}^s \\ \dot{\boldsymbol{\theta}}(t_s) = \ddot{\boldsymbol{\theta}}(t_s) = \mathbf{0}, \dot{\boldsymbol{\theta}}(t_f) = \ddot{\boldsymbol{\theta}}(t_f) = \mathbf{0} \\ \boldsymbol{\theta}_{min} \le \boldsymbol{\theta} \le \boldsymbol{\theta}_{max}, \dot{\boldsymbol{\theta}}_{min} \le \dot{\boldsymbol{\theta}} \le \dot{\boldsymbol{\theta}}_{max}, \ddot{\boldsymbol{\theta}}_{min} \le \ddot{\boldsymbol{\theta}} \le \ddot{\boldsymbol{\theta}}_{max} \end{cases} \tag{13}$$

where $\boldsymbol{\theta}^s$ is the start configuration of space robot, and $\boldsymbol{x}_b^s$ and $\boldsymbol{x}_e^s$ are the initial pose of the base and end-effector, respectively. The determination of the joint trajectories requires the final pose of the end-effector approach to its desired pose as much as possible, i.e. $\delta \boldsymbol{x}_e \Rightarrow \mathbf{0}$.

3.2. Cost functions for trajectory planning

Except the end-effector's manipulation mission, for a redundant space manipulator, additional cost functions should be defined for other specific reasons [17]. For instance, large orientation disturbance to the base is not expected during the motion of the space manipulator since the requirements of communication and observation. A cost function that defines the minimum disturbance to the base can be given by

$$\Gamma = \int_{t_s}^{t_f} \|\boldsymbol{x}_b\|_{\mathbf{Q}} \, dt \tag{14}$$

where $\|\cdot\|$ represents the norm of the vector and $\mathbf{Q}$ is a positive definite weight matrix. One can also define the total joint angle change as a goal function when drastic joint variations are not expected:

$$\Gamma = \int_{t_s}^{t_f} \|\boldsymbol{\theta} - \boldsymbol{\theta}^s\|_{\mathbf{Q}} \, dt \tag{15}$$

Here $\boldsymbol{\theta}^s$ is the start configuration of the space manipulator. If one hope that the robot operates at its central configuration as close as possible, another goal can be defined as

$$\Gamma = \int_{t_s}^{t_f} \|\boldsymbol{\theta} - \boldsymbol{\theta}^c\|_{\mathbf{Q}} \, dt \tag{16}$$

where $\boldsymbol{\theta}^c = (\boldsymbol{\theta}_{min} + \boldsymbol{\theta}_{max})/2$ represents the middle configuration of the robot within its lower boundary $\boldsymbol{\theta}_{min}$ and upper boundary $\boldsymbol{\theta}_{max}$. When the maximum manipulability is required at the final stage, one can define

$$\Gamma = \sqrt{\det\left(\mathbf{J}^{\mathrm{T}}(\boldsymbol{\theta}^f)\mathbf{J}(\boldsymbol{\theta}^f)\right)} \tag{17}$$

Likewise, additional cost functions can also be incorporated when new objectives are required. The trajectory planning issue under a list of inequality constraints $g_i(\boldsymbol{\theta}(t))$ and equality constraints $h_i(\boldsymbol{\theta}(t))$ can be described as follows:

$$\begin{aligned} &\min_{\boldsymbol{\theta}(t)} \quad \Gamma(\boldsymbol{\theta}(t)) \\ &\text{subject to} \quad g_i(\boldsymbol{\theta}(t)) < 0 \\ &\qquad h_i(\boldsymbol{\theta}(t)) = 0 \end{aligned} \tag{18}$$

3.3. Parametrization of the trajectories

The aforementioned optimization issue in Eq. (18) is solved as a non-linear programming problem with imposed equality and inequality constraints. Each joint trajectory $\theta_i(t)$ is parametrized by the Bézier curve which is widely used in compute graphics to model sooth curves [20,21]. In this paper, fifth-order Bézier curves are employed to describe the ith joint trajectory:

$$\theta_i(\tau) = \sum_{j=0}^{m} b_{j,m}(\tau) P_{ij} = \sum_{j=0}^{m} \binom{m}{j} (1-\tau)^{m-j} \tau^j P_{ij}, \quad \tau \in [0,1] \tag{19}$$

where the polynomials $b_{j,m}(\tau)$ are known as Bernstein basis polynomial of order m and $\binom{m}{j}$ is the binomial coefficient. P_{ij} is the given point to construct the Bézier curve. Since τ is the normalized time, for the trajectory execution time $T = t_f - t_s$, if we define $t = \tau \cdot T$, the joint trajectory becomes as follows:

$$\begin{cases} \theta_i = \sum_{j=0}^{m} \binom{m}{j} \left(1 - \frac{t}{T}\right)^{m-j} \left(\frac{t}{T}\right)^j P_{ij} \\ \dot{\theta}_i = \frac{d\theta_i}{d\tau} \frac{d\tau}{dt} = \frac{1}{T} \dot{\theta}_i = \frac{m}{T} \sum_{j=0}^{m-1} b_{j,m-1}\left(\frac{t}{T}\right) (P_{i,j+1} - P_{ij}) \end{cases} . \tag{20}$$

Table 2
Mass and inertia parameters of space robot.

	Base	Link1	Link2	Link3	Link4	Link5	Link6	Link7
m (kg)	200	4.0	8.0	2.0	6.0	2.0	2.0	5.0
I_{xx} (kg·m²)	50	0.3	0.8	0.1	0.7	0.1	0.1	0.4
I_{xy} (kg·m²)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
I_{xz} (kg·m²)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
I_{yy} (kg·m²)	100	0.3	0.5	0.1	0.4	0.1	0.1	0.4
I_{yz} (kg·m²)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
I_{zz} (kg·m²)	100	0.2	0.8	0.1	0.7	0.1	0.1	0.2
a_x (m)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
a_y (m)	0.0	0.0	−0.7	0.0	−0.5	0.0	0.0	0.0
a_z (m)	0.0	0.128	0.084	0.09	0.084	0.09	0.084	0.12
b_x (m)	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b_y (m)	0.084	0.084	−0.66	−0.084	−0.46	0.084	−0.09	0.0
b_z (m)	−0.6	0.128	0.0	0.0	0.0	0.0	0.0	0.21

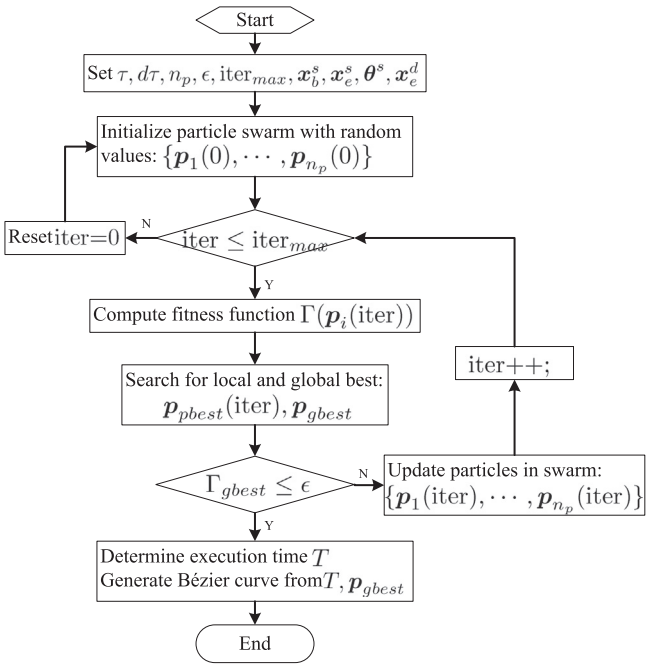

Fig. 3. The flow chart of PSO algorithm for trajectory planning.

The joint velocity and acceleration boundaries in Eq. (13) can be satisfied through determination of the execution time T by the following expression:

$$T \geq \max\left(\frac{|\dot{\boldsymbol{\theta}}_{max}|}{\dot{\boldsymbol{\theta}}_{max}}, \sqrt{\frac{|\ddot{\boldsymbol{\theta}}_{max}|}{\ddot{\boldsymbol{\theta}}_{max}}}\right) \tag{21}$$

where $\dot{\boldsymbol{\theta}} = d\boldsymbol{\theta}/d\tau$ and $\ddot{\boldsymbol{\theta}} = d^2\boldsymbol{\theta}/d\tau^2$. Substituting the equality constraints in Eqs. (13)–(20)

$$\begin{cases} P_{i0} = P_{i1} = P_{i2} = \theta_i^s \\ P_{i3} = P_{i4} = P_{i5} \end{cases} \tag{22}$$

Since θ_i^s is already known, the Bézier curve only relies on one parameter P_{i5}. Define $\boldsymbol{p} = [P_{15}, \ldots, P_{n5}]^{\mathrm{T}}$ as design variables, once $\boldsymbol{p}$ is determined, each joint trajectory is solved accordingly. Consider Eqs. (3) and (4), the pose of base and end-effector can be derived as

$$\boldsymbol{x}_b = \int_0^1 \begin{bmatrix} \mathbf{E}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_q(\boldsymbol{q}_b) \end{bmatrix} \mathbf{J}_a \dot{\overline{\boldsymbol{\theta}}}\, d\tau \tag{23}$$

$$\boldsymbol{x}_e = \int_0^1 \begin{bmatrix} \mathbf{E}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_q(\boldsymbol{q}_e) \end{bmatrix} \mathbf{J}_g \dot{\overline{\boldsymbol{\theta}}}\, d\tau \tag{24}$$

Consequently, the trajectory planning issue expressed in Eq. (18) can be transformed to

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \Gamma(\boldsymbol{p}) \\ \text{subject to} \quad & g_i(\boldsymbol{p}) < 0 \\ & h_i(\boldsymbol{p}) = 0 \\ & \boldsymbol{p}_{min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{max} \end{aligned} \tag{25}$$

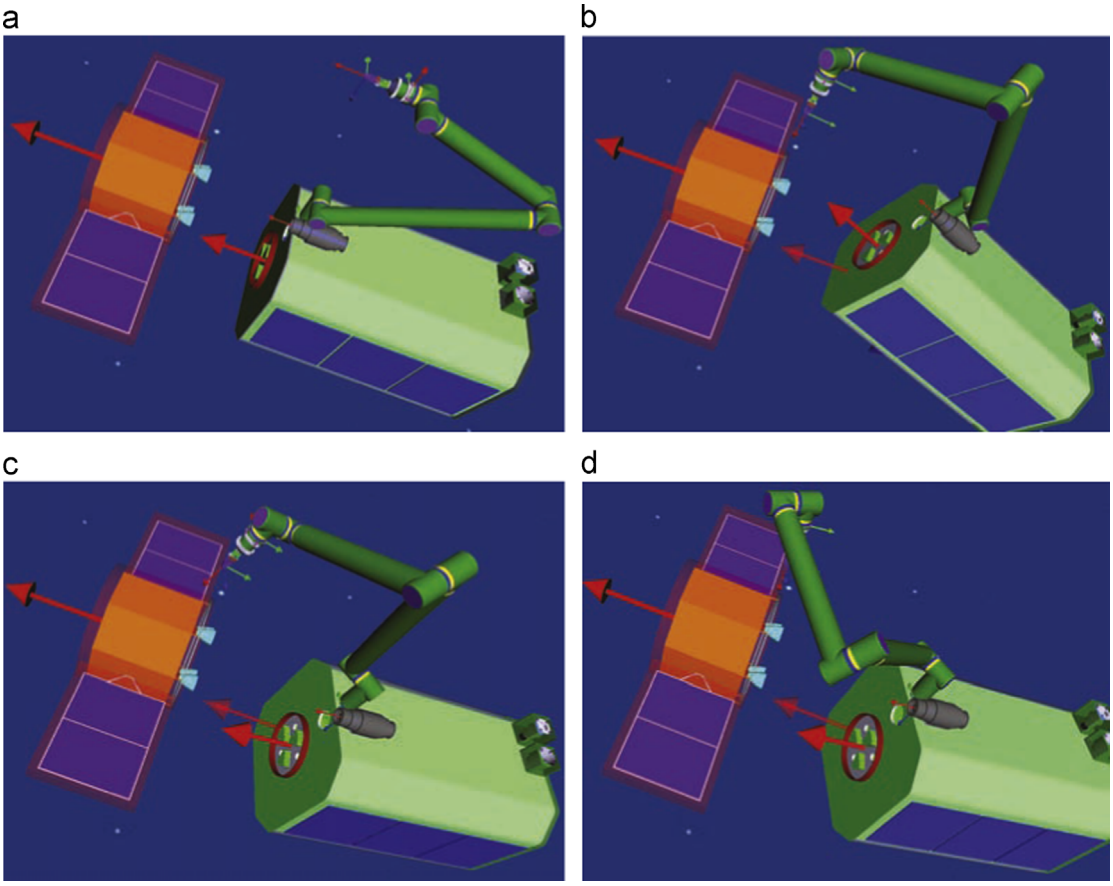

Fig. 4. Initial and final pose of the space robot.

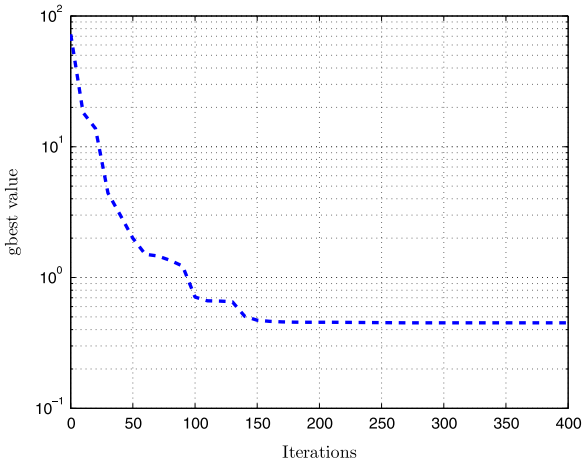

Fig. 5. The convergence of the best fitness evaluation.

4. Particle Swarm Optimization

PSO is a stochastic search method yet with simpler philosophy. It was inspired by the coordinated motion of swarmed animals like flying birds and swimming fishes [22]. The states adjust of each particle in swarm takes into account the effect of stochastic, cognitive and social influence. A schematic diagram of the PSO with 4 particles is shown in Fig. 2. The PSO algorithm first initializes a population of particles with random initial values within the feasible searching space. The dimension of the particle represents the number of design variables. Each particle is evaluated by the fitness function to search its own best known position (local) so far and the swarm's best known position (global) so far in the searching space. The movement of each particle is guided by the local and global best position and updated in each generation. When better positions are being discovered, these will be then chosen to lead the movement of the swarm. This process is repeated generation by generation until a specified condition is met or a promising solution is found.

According to the illustration in Section 3.3, the design variables are the joint position $\boldsymbol{p}$. Suppose the position and the velocity of the ith particle are represented by $(p_{i1}, p_{i2}, \ldots, p_{in})$ and $(v_{i1}, v_{i2}, \ldots, v_{in})$, respectively, their updates can be expressed as follows as determined by PSO algorithm:

$$\begin{cases} v_{ik} = wv_{ik} + c_1 r_1 (p_{b_{ik}} - p_{ik}) + c_2 r_2 (p_{g_i} - p_{ik}) \\ p_{ik} = p_{ik} + v_{ik} \end{cases} \tag{26}$$

where c_1 and c_2 are acceleration constants and r_1 and r_2 are the uniformly distributed value between $[0, 1]$. The vector $(p_{b_{i1}}, \ldots, p_{b_{in}})$ stores the local best position of ith particle so far and $(p_{g_1}, \ldots, p_{g_n})$ represents the global best position of the swarm so far. w is an adaptive inertia weight factor which

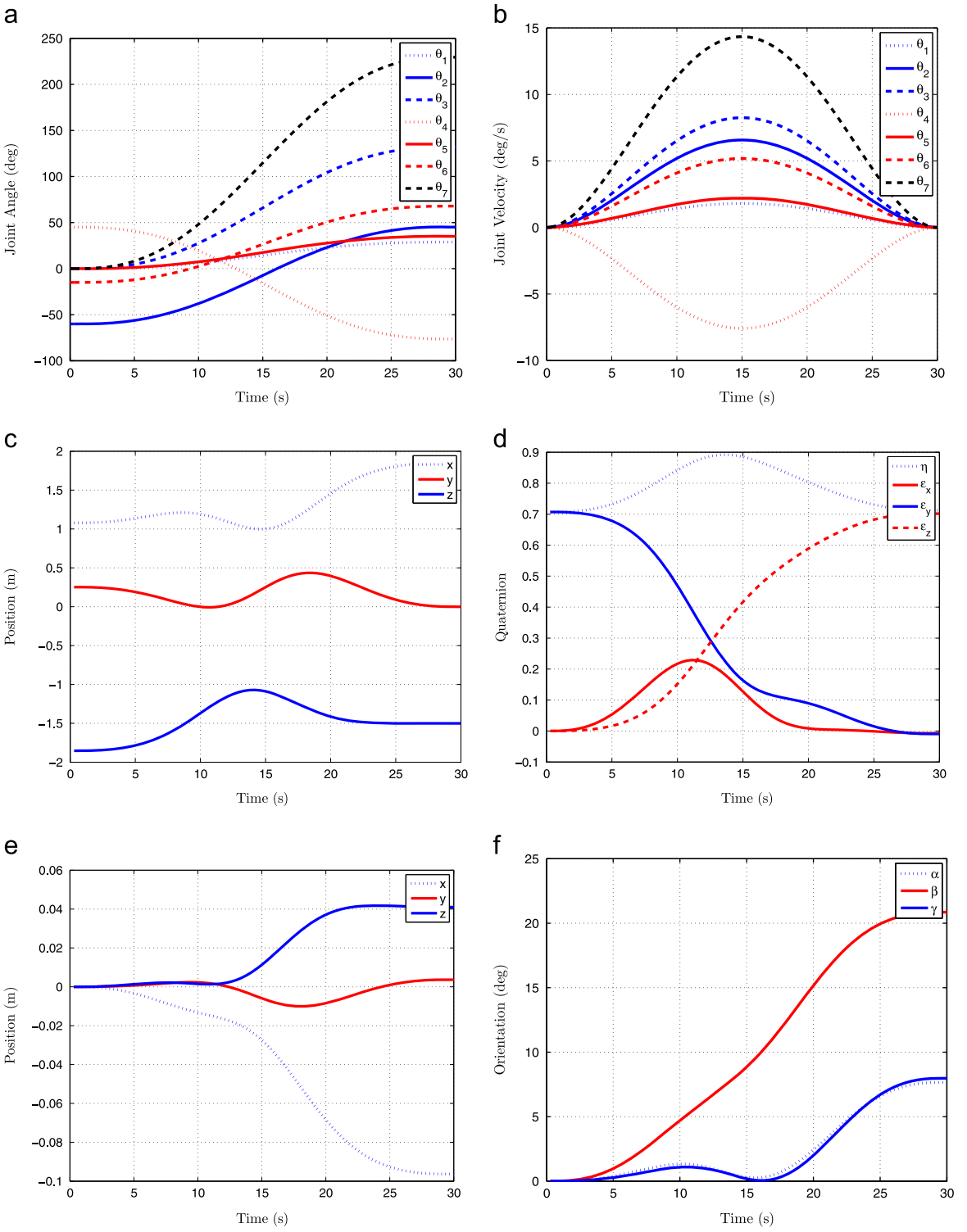

Fig. 6. The joint motion laws and pose change of the end-effector and base for case 1.

effectively controls the scope of the search [23]. In this paper, a linear decreasing strategy on w is used:

$$w = w_{min} + \frac{\text{iter}_{max} - \text{iter}}{\text{iter}_{max}}(w_{max} - w_{min}) \tag{27}$$

where iter_{max} is the maximal number of iterations, w_{min} and w_{max} are the lower and the upper bound of inertia weight, respectively. From Eq. (26), the update velocity of ith particle is consisting of three components: a momentum of its previous velocity, velocity increments according to its local

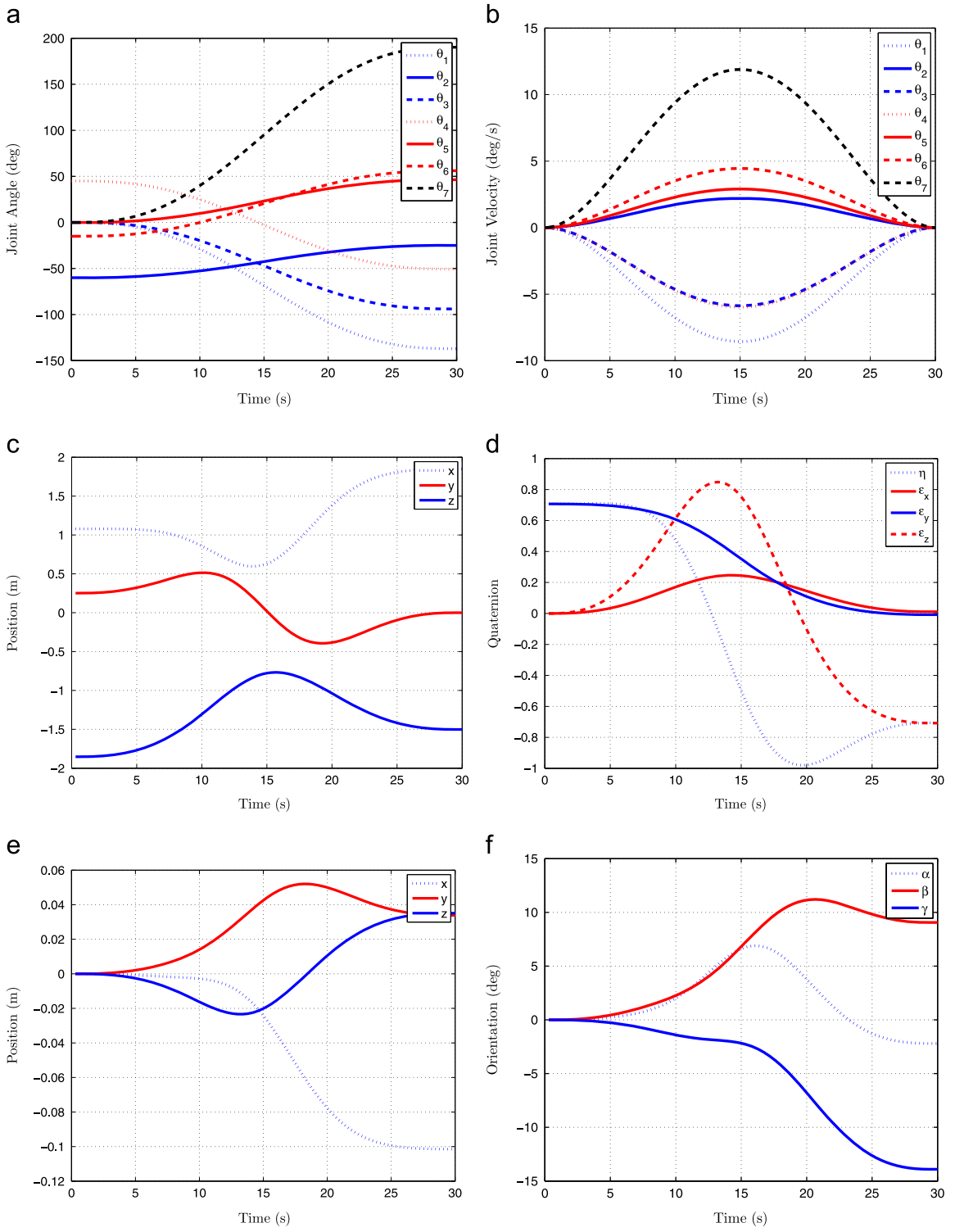

Fig. 7. The joint motion laws and pose change of the end-effector and base for case 2.

best and global best position. Eventually, the position of the particle is renewed with its previous position and new displacement induced by the new velocity.

In order to choose the local and global best particles in each iteration, fitness function is employed in PSO to evaluate the quality of particle in swarm and drive them

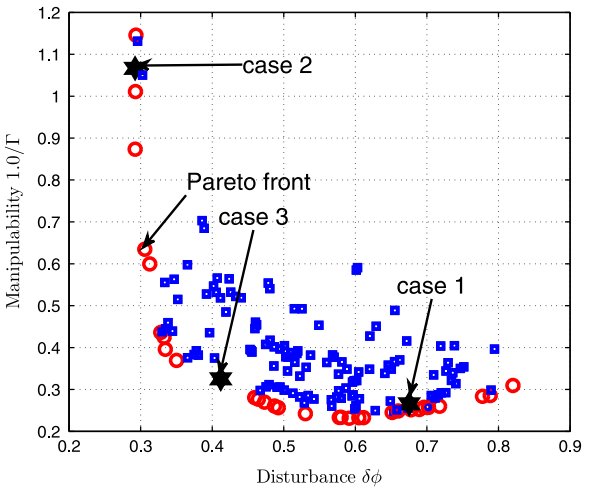

Fig. 8. The description of the Pareto front.

to the target. How to choose the fitness function depends on the robot type, required space missions and the optimized objectives. In fact, refer to Section 3.2, the fitness function can be designed to minimize or maximize multiple/disjunctive cost functions. Due to the redundancy resolution, for the point-to-point trajectory planning, a good choice for the fitness function is

$$\Gamma(\boldsymbol{p}) = \|\delta \boldsymbol{x}_e\|_{\mathbf{Q}_e} + \sum_i \Gamma_i(\boldsymbol{p}) \tag{28}$$

Another issue that encounters in PSO is the practical constraints imposed to the design variables as illustrated in Eq. (13). There exist various strategies, such as penalty function and repair algorithms, to cope with constraints in PSO [24]. In our method constraints are tackled by repairing infeasible solutions. The following strategy to particles' position is adopted:

$$p_{ik} = \begin{cases} p_{min} & \text{if } p_{ik} < p_{min} \\ p_{max} & \text{if } p_{ik} > p_{max} \\ p_{ik} & \text{otherwise} \end{cases} \tag{29}$$

Note that both position and velocity handling need to be considered in repairing strategy. If $p_{ik} < p_{min}$ without velocity alteration, $v_{ik} < 0$ holds will drive the particle out of the feasible searching space again (similarly for $p_{ik} > p_{max}$ and $v_{ik} > 0$). Another reason for velocity handling is the significant influence of velocity to the performance of PSO. When it is too large, possible solution may be skipped, otherwise, if the velocity is too small, the convergence speed would be too slow. Therefore, it would make sense to set the velocity to zero or invert the kth velocity component so that the particle returns to its feasible searching region.

5. Simulation results

The space robotic system used in this section is composed of a 6 DOF spacecraft and a 7 DOF kinematically redundant manipulator as shown in Fig. 1. The mass and inertia properties of the space robot are listed in Table 2, where $\boldsymbol{a}_i$, $\boldsymbol{b}_i$, and $\mathbf{I}_i$ are expressed in its own body frame.

The flow chart of the proposed algorithm is shown in Fig. 3. During the processing, the PSO algorithm first finds the optimal solution to construct the Bézier curve, after that, the execution time T is determined according to Eq. (21) to fulfil the imposed constraints. The joint trajectory is accordingly generated. In order to verify the effectiveness of the proposed method, three cases will be investigated by numerical simulations. The involved parameters of PSO are listed as follows:

$$\begin{aligned} &n_p = 15; \quad \text{iter}_{max} = 400; \quad d\tau = 0.01; \\ &c_1 = c_2 = 1.496; \quad w_{max} = 0.7298; \quad w_{min} = 0.3; \end{aligned} \tag{30}$$

In the first simulation, the end-effector of the manipulator is commanded to move to a point above the target satellite from its initial pose $\boldsymbol{x}_e^s = (1.0785, 0.2520, -1.8528, 0.7071, 0.0, 0.7071, 0.0)$ to the final pose of the end-effector $\boldsymbol{x}_e^f = (1.85, 0.0, -1.5, 0.7071, 0.0, 0.0, 0.7071)$. The corresponding configuration of the space robot is $\boldsymbol{\theta}^s = (0.0, -\pi/3, 0.0, \pi/4, 0.0, -\pi/12, 0.0)$ and the initial pose of the base is $\boldsymbol{x}_b^s = (0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0)$. From Eq. (22) and considering the limits of joint range in (13), it shows that $\boldsymbol{p} \in [\boldsymbol{\theta}_{min}, \boldsymbol{\theta}_{max}]$. The fitness function in Eq. (28) is used without considering other specific objectives. The determination of $\mathbf{Q}_e$ depends on the position and orientation accuracy requirements. In this paper, the admitted position error is set to 0.005 m and orientation error is 1°, the weight matrix $\mathbf{Q}_e$ can be determined as follows:

$$\mathbf{Q}_e = \begin{bmatrix} \frac{1}{0.005}\mathbf{E}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \frac{1}{\sin\left(\frac{\pi}{360}\right)}\mathbf{E}_3 \end{bmatrix} \tag{31}$$

The proposed trajectory planning algorithm stops when the convergence criteria are satisfied $\Gamma(\boldsymbol{p}^*) \le \epsilon = 1.0$. PSO algorithm successfully found a solution $\boldsymbol{p}^* = (0.503, 0.788, 2.303, -1.334, 0.612, 1.184, 4.005)$ that fulfil all the constraints. Fig. 4 shows the initial and final pose of the space robotic system for the above solution. The convergence of best fitness function versus the number of iteration is shown in Fig. 5. The fitness value decreases monotonously which exhibits the searching power of the PSO algorithm. The execution time $T = 30$ s is determined by Eq. (21). In terms of the searching solution $\boldsymbol{p}^*$ and the execution time T, the joint motion laws and corresponding pose change of the end-effector and the base are shown in Fig. 6. It is noted that the orientation representation of the base is Z–Y–X Euler angles for intuition reasons. From the simulation results, one can see that with the designed joint motion, the end-effector successfully converged to the desired pose. The final errors of the system states are $\delta \boldsymbol{x}_e = [1.476, -0.612, -0.661, 1.694, -11.712, -7.291] \times 10^{-3}$. In fact, there are infinite solutions since the existence of the kinematic and non-holonomic redundancy where the chosen is just one of them. This solution can be used as an initial feasible solution for the purpose of subsequent optimization.

In the second and the third simulation, the end-effector is required to approach the target as case 1 but with additional objectives. The complete solution for simulation 2 includes not only approaching to the target, but also minimizing the base orientation disturbance which makes it a multiple objectives optimization problem. Two objectives are included in this simulation while approaching to

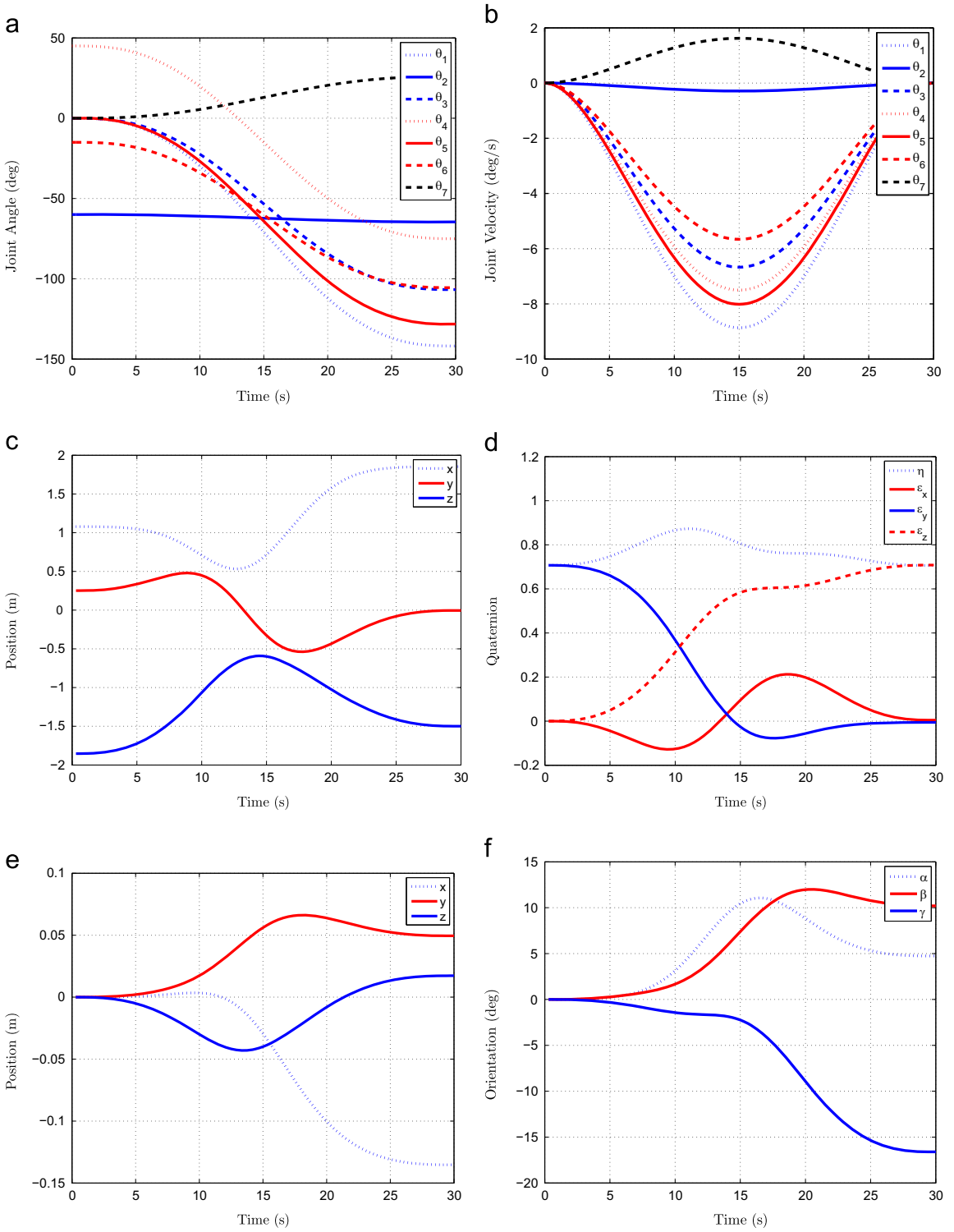

Fig. 9. The joint motion laws and pose change of the end-effector and base for case 3.

the target is the pre-emptive one. 100 solutions like in case 1 were first obtained and the solution for the second objective was selected from this pool. The optimal solution for both approaching mission and minimizing base orientation disturbance is $\boldsymbol{p}^* = (-2.392, -0.436, -1.639, -0.884, 0.810, 0.980, 3.319)$, while the total orientation disturbance is 16.74°. The final pose of the space robotic system can be found in Fig. 4. The final errors of the system

states are $\delta \boldsymbol{x}_e = [-0.412, -0.140, -1.503, 13.572, 2.239, -1.706] \times 10^{-3}$. Fig. 7 gives out the history of all joint motions and related change of the end-effector and the base. One can see that with the designed joint trajectories, the end-effector successfully approaches to the required pose while the objective of minimizing base orientation disturbance is obtained.

In the third simulation, both minimizing the base orientation disturbance and maximizing the finial manipulability are required as additional objectives. As shown in Fig. 8, the relationship between base orientation disturbance and manipulability is conflicting, consequently, it is not possible to reach an optimal solution with respect to both of the objectives evaluated individually. Attention has to be paid to the concept of Pareto optimality and Pareto front [25]. Here a linear combination of the two required objectives as expressed in Eq. (28) is used to evaluate the fitness function. The optimal solution for case 3 is $\boldsymbol{p}^* = (-2.476, -1.127, -1.863, -1.310, -2.237, -1.842, 0.452)$. The corresponding manipulability of the manipulator is 2.707 and the total base orientation disturbance is 20.06°. Fig. 4 exhibits the final pose of the space robotic system for case 3. The final errors of the system states are $\delta \boldsymbol{x}_e = [2.739, -4.383, 1.524, 7.275, -0.834, 1.537] \times 10^{-3}$. Fig. 9 shows the history of all joint motions and related change of the end-effector and the base. It is shown that the change of the base is bigger in case 3 when compared to Figs. 7 and 9 since the trade-off between the manipulability and base disturbance. The chosen cost function affects the solution of optimization and eventually influences the final pose of space robotic system. Conclusively, the generated joint trajectories for all three cases are smooth and applicable to the control of the space robot, which satisfy the requirements listed in Section 3.

6. Conclusions

Due to the dynamics coupling effect between the spacecraft and the manipulator, the determination of the end-effector pose relies not only on the current joint position, but also on the history motion of each joint. Therefore, the given end-effector pose cannot be handled only by inverse kinematics algorithm as fixed base manipulator. In this paper, a new trajectory planning method for free-floating space robot is proposed. Its main differences are as follows: (i) the non-holonomic property of the free-floating space robot is considered, (ii) kinematically redundant manipulator is employed for further optimization, (iii) joint trajectories are parametrized by the Bézier curves for its simplicity and normalization, (iv) the execution time of the joint motion is determined with respect to the joint velocity and acceleration boundaries, (v) PSO algorithm with adaptive inertia weight is adopted to search for optimal solutions to construct the Bézier curve, and (vi) multiple objectives optimization can be incorporated into the proposed method.

The use of the unit quaternion for attitude representation impedes the orientation of the base and the end-effector from the orientation representation singularity. Moreover, since the forward kinematics is applied without inversion of Jacobian matrix, the kinematics and dynamics singularities of space robot will not affect the joint motion during the whole trajectory time. The simulation results demonstrate that the proposed method can be used to minimize the orientation disturbance to the base or perform optimization with multiple objectives according to various cost functions. The change of the mass and inertia parameters due to the fuel consumption and structure flexibility is not considered which would be one of our future work. Collision detection and avoidance will also be incorporated into our future work.

Acknowledgments

The authors gratefully acknowledge the support of the TUM Graduate School's Thematic Graduate/Faculty Graduate Center Mechanical Engineering at the Technische Universität Muenchen. The first author gratefully acknowledges the financial support of Chinese Scholarship Council (CSC No. 2010629015).

References

[1] G. Hirzinger, B. Brunner, J. Dietrich, J. Heindl, Rotex—the first remotely controlled robot in space, in: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, vol. 3, 1994, pp. 2604–2611, http://dx.doi.org/10.1109/ROBOT.1994.351121.

[2] M. Oda, K. Kibe, F. Yamagata, ETS-VII space robot in-orbit experiment satellite, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, vol. 1, 1996, pp. 739–744, http://dx.doi.org/10.1109/ROBOT.1996.503862.

[3] A. Ogilvie, J. Allport, M. Hannah, J. Lymer, Autonomous satellite serving using the orbital express demonstration manipulator system, in: Proceedings of the Ninth International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2008, pp. 25–29.

[4] I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, E. Dupuis, Autonomous capture of a tumbling satellite, J. Field Robot. 24 (4) (2007) 275–296, http://dx.doi.org/10.10.1002/rob.20194.

[5] Y. Umetani, K. Yoshida, Resolved motion rate control of space manipulators with generalized Jacobian matrix, IEEE Trans. Robot. Autom. 5 (3) (1989) 303–314, http://dx.doi.org/10.1109/70.34766.

[6] E. Papadopoulos, S. Dubowsky, Dynamic singularities in the control of free-floating space manipulators, ASME J. Dyn. Syst. Meas. Control 115 (1) (1993) 44–52, http://dx.doi.org/10.1115/1.2897406.

[7] Y. Nakamura, R. Mukherjee, Nonholonomic path planning of space robot via a bidirectional approach, IEEE Trans. Robot. Autom. 7 (4) (1991) 500–514, http://dx.doi.org/10.1109/70.86080.

[8] S. Dubowsky, M. Torres, Path planning for space manipulators to minimize spacecraft attitude disturbances, in: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2522–2528, http://dx.doi.org/10.1109/ROBOT.1991.132005.

[9] K. Yoshida, K. Hashizume, S. Abiko, Zeros reaction maneuver: flight validation with ETS-VII space robot and extension to kinematically redundant arm, in: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, vol. 1, 2001, pp. 441–446, http://dx.doi.org/10.1109/ROBOT.2001.932590.

[10] D. Dimitrov, K. Yoshida, Utilization of holonomic distribution control for reactionless path planning, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, 2006, pp. 3387–3392, http://dx.doi.org/10.1109/IROS.2006.282574.

[11] S. Xu, H. Wang, D. Zhang, B. Yang, Adaptive reactionless motion control for free-floating space manipulators with uncertain kinematics and dynamics, in: Proceedings of the Third IFAC International Conference on Intelligent Control and Automation Science, vol. 3, 2013, pp. 646–653. http://dx.doi.org/10.3182/20130902-3-CN-3020.00145.

[12] O. Agrawal, Y. Xu, On the global optimum path planning for redundant space manipulators, IEEE Trans. Syst. Man Cybern. 24 (9) (1994) 1306–1316, http://dx.doi.org/10.1109/21.310507.

[13] P. Huang, Y. Xu, B. Liang, Minimum-torque path planning of space robots using genetic algorithms, Int. J. Robot. Autom. 21 (3) (2006) 229–236, http://dx.doi.org/10.2316/Journal.206.2006.3.206-2945.

[14] W. Xu, C. Li, B. Liang, Y. Liu, Y. Xu, The Cartesian path planning of free-floating space robot using particle swarm optimization, Int. J.

Adv. Robot. Syst. 5 (3) (2008) 301–310, http://dx.doi.org/10.5772/5605.
[15] W. Xu, C. Li, X. Wang, B. Liang, Y. Xu, Study on non-holonomic cartesian path planning of a free-floating space robotic system, Adv. Robot. 23 (1–2) (2009) 113–143, http://dx.doi.org/10.1163/156855308X392708.
[16] E. Kaigom, T. Jung, J. Rossmann, Optimal motion planning of a space robot with base disturbance minimization, in: Proceedings of the 11th Symposium on Advanced Space Technologies in Robotics and Automation, vol. 1, 2011, pp. 1–6.
[17] R. Lampariello, D. Tuong, C. Castellini, G. Hirzinger, J. Peters, Trajectory planning for optimal robot catching in real-time, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation, vol. 1, 2011, pp. 3719–3726, http://dx.doi.org/10.1109/ICRA.2011.5980114.
[18] J. Diebel, Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors, 2006.
[19] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics Modelling, Planning and Control, 1st edition, Springer, London, 2009.
[20] G. Farin, Curves and Surfaces for Computer-aided Geometric Design: A Practical Guide, 4th edition, Academic Press, San Diego, 1996.
[21] J. Faraway, M. Reed, J. Wang, Modelling three-dimensional trajectories by using bezier curves with application to hand motion, Appl. Stat. 56 (5) (2007) 571–585, http://dx.doi.org/10.1111/j.1467-9876.2007.00592.x.
[22] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43, http://dx.doi.org/10.1109/MHS.1995.494215.
[23] K. Kentzoglanakis, M. Poole, Particle swarm optimization with an oscillating inertia weight, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, 2009, pp. 1749–1750, http://dx.doi.org/10.1145/1569901.1570140.
[24] S. Helwig, Particle swarms for constrained optimization (Ph.D. thesis), Erlangen University, Erlangen, Germany, 2010.
[25] O. Castillo, L. Trujillo, P. Melin, Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots, Soft Comput. 11 (3) (2007) 269–279, http://dx.doi.org/10.1007/s00500-006-0068-4.