

MULTI-WAYPOINT-BASED PATH PLANNING FOR FREE-FLOATING SPACE ROBOTS

Suping Zhao,^{*,**} Bruno Siciliano,^{**} Zhanxia Zhu,^{*} Alejandro Gutiérrez-Giles,^{**} and Jianjun Luo^{*}

Abstract

This paper studies the multi-waypoint-based path planning problem (MWPP) for redundant space robots. The end-effector of a space robot should visit a set of predefined waypoints with optimal distance, and the free-floating base should suffer minimum attitude disturbances from the manipulator during manoeuver. The MWPP is decomposed into two sub-problems: the problem of optimal waypoint-sequence and the problem of optimal joint-movements. First, the Hybrid Self-adaptive Particle Swarm Optimization algorithm is proposed for optimal waypoint-sequence. Second, an Improved Particle Swarm Optimization algorithm, combined with direct kinematics of the space robot, is proposed for optimal joint-movements. Finally, simulations are presented to validate the approach, including comparisons with other approaches.

Key Words

Multi-waypoint-based path planning problem, free-floating base, redundant space robots, waypoint-sequence, particle swarm optimization algorithm

1. Introduction

Free-floating space robots (FFSRs) attract both scientists and practitioners' extensive attention due to energy saving capability. Path planning is essential for both industrial [1]–[4] and space robots [5], [6]. Transforming a path planning problem into a parametric optimization problem is a well-received approach. The optimization algorithms, including a hybrid algorithm [1] (combing genetic algorithm and tabu search), a Dubins-RRT algorithm [2], an evolving ant colony algorithm [3] (combing genetic algorithm and ant colony algorithm) and an improved memetic algorithm [4], were employed to optimize unknown parameters so that the developed industrial robot reaches desired

position. In [5] and [6], a genetic algorithm [5] and a particle swarm optimization algorithm combined with a differential evolution algorithm [6] were employed to optimize parameters for space robots, respectively. Although the path planning problem for FFSRs was well studied, traditional approaches contribute to solving point-to-point path planning problems, meaning that a FFSR executes a task each time. With complexity increment in outer space, space missions increase. If a FFSR executes several tasks successively, energy will be employed more efficiently.

The MWPP means that a series of tasks are executed successively, where the location of each task is viewed as a waypoint and specific methods for executing tasks are not considered. In MWPP, an optimal path, along which each waypoint is visited once, is derived considering optimization in distance, time or energy. In fact, MWPP for industrial robots [7], [8] was widely studied for high productivity. However, the developed industrial robots are non-redundant. Redundant robots provide high dexterity, which can be employed to optimize performance indicators such as singularity avoidance, joint limits and obstacle avoidance. The gravity factors were considered in [9]–[11], referring the mathematical models of robotics [9]–[11]. However, unlike ground environment [1]–[4], [7]–[11], there is microgravity in outer space. The base of FFSR is in the free-floating mode, and there is strong coupling between base and manipulator during manoeuver. Therefore, the approaches employed to study MWPP for industrial robots cannot be directly employed for FFSR.

The MWPP for FFSR is a novel problem, and some problems should be solved. The optimal waypoint-sequence should be managed first. Second, the attitude disturbance acting on free-floating base is accumulated during manoeuver, and the attitude deviation grows rapidly as waypoints increase. Third, the pose error at each waypoint influences path planning to next waypoint. As waypoints increase, the pose error is dramatically accumulated. In this work, the MWPP for FFSR is decomposed into two sub-problems: the problem of optimal waypoint-sequence and the problem of optimal joint-movements. First, the Hybrid Self-adaptive Particle Swarm Optimization (HSPSO) algorithm is proposed for optimal waypoint-sequence. Then, an Improved Particle Swarm Optimization (IPSO) algorithm is proposed for optimal joint-movements. The paper is organized as follows. Preliminaries and notation are presented

* National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, 127 Youyi West Road, Xi'an, 710072, China; e-mail: suping36@mail.nwpu.edu.cn, {zhuzhanxia, jjluo}@nwpu.edu.cn

** Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy; e-mail: siciliano@unina.it, alejandrogiles@yahoo.com.mx

Corresponding author: Zhanxia Zhu

Recommended by Dr. S.G. Ponnambalam
(DOI: 10.2316/J.2019.206-0032)

in Section 2. The solution of MWPP for FFSR is developed in Section 3. In Section 4, simulations are developed, including comparisons with other approaches. Section 5 concludes the paper.

2. Preliminaries and Notation

2.1 Mathematical Model of FFSR

The FFSR satisfies linear and angular momentum conservation laws, shown in (1). For simplicity, the initial momentum is usually set to 0.

$$\mathbf{H}_b \dot{\mathbf{x}}_b + \mathbf{H}_{bm} \dot{\mathbf{q}} = \mathbf{0} \quad (1)$$

As \mathbf{H}_b is symmetric and positive definite, the motion of the free-floating base can be expressed as

$$\dot{\mathbf{x}}_b = \begin{bmatrix} \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = -\mathbf{H}_b^{-1} \mathbf{H}_{bm} \dot{\mathbf{q}} = \mathbf{J}_{bm} \dot{\mathbf{q}} \quad (2)$$

The kinematic equation of FFSR is formulated as

$$\begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_b \begin{bmatrix} \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} + \mathbf{J}_m \dot{\mathbf{q}} = [\mathbf{J}_m + \mathbf{J}_b \mathbf{J}_{bm}] \dot{\mathbf{q}} = \mathbf{J}_g \dot{\mathbf{q}} \quad (3)$$

In the above equations, $\dot{\mathbf{x}}_b \in \mathbf{R}^6$ denotes the base velocity; $\dot{\mathbf{q}} \in \mathbf{R}^n$ denotes the vector of joint angular velocity; $\mathbf{H}_b \in \mathbf{R}^{6 \times 6}$ denotes the inertia matrix of base; $\mathbf{H}_{bm} \in \mathbf{R}^{6 \times n}$ denotes the coupling inertia matrix; $\mathbf{v}_b \in \mathbf{R}^3$ denotes the linear velocity of base; $\boldsymbol{\omega}_b \in \mathbf{R}^3$ denotes the angular velocity of base; $\mathbf{J}_{bm} \in \mathbf{R}^{6 \times 6}$ denotes the base-manipulator Jacobian; $\mathbf{J}_b \in \mathbf{R}^{6 \times 6}$ denotes the base Jacobian; $\mathbf{J}_m \in \mathbf{R}^{6 \times n}$ denotes the manipulator Jacobian; $\mathbf{v}_e \in \mathbf{R}^3$ denotes the linear velocity of end-effector; $\boldsymbol{\omega}_e \in \mathbf{R}^3$ denotes the angular velocity of end-effector and \mathbf{J}_g denotes the Generalized Jacobian matrix.

2.2 MWPP for FFSR

2.2.1 Waypoint-Sequence Problem

The pose of each waypoint is predefined in Euclidean space. The end-effector is required to visit each waypoint once and remain in last waypoint. The optimal path length and the angular distance of end-effector are considered and expressed as

$$\min F := \sum_{i=1}^{N-1} (d_{t_i, t_{i+1}} + \xi r_{t_i, t_{i+1}}) \quad (4)$$

In (4), N denotes the number of waypoints; $\{t_i\}$ ($i=1, \dots, N$) denotes a waypoint-sequence; $d_{t_i, t_{i+1}}$ and $r_{t_i, t_{i+1}}$ denote the sub-path length and the angular distance from waypoint t_i to waypoint t_{i+1} , respectively; ξ makes a trade-off between the path length and the angular distance.

Assume that (X_i, Θ_i) and (X_{i+1}, Θ_{i+1}) are pose of waypoints t_i and t_{i+1} , respectively, then $d_{t_i, t_{i+1}} = \|X_i - X_{i+1}\|$ and $r_{t_i, t_{i+1}} = \|\Theta_i - \Theta_{i+1}\|$, where $\|\cdot\|$ denotes the Euclidean norm.

2.2.2 Optimal Joint-Movements Problem

As the end-effector should visit each waypoint accurately and the base attitude should be minimized during manoeuvre, the objective function is given by

$$\min G := G_1 + \zeta G_2 \quad (5)$$

In (5), ζ makes a trade-off between G_1 and G_2 . G_1 , shown in (6), is generated by solutions of two sub-problems and includes two parts. The first part denotes the path length error and the angular distance error, while the second part denotes the sum of pose error at the waypoints. G_2 , shown in (7), denotes the attitude disturbance acting on the base during manoeuvre.

$$G_1 := e'_p + \lambda_1 e''_p + \sum_{k=1}^N (e'_k + \lambda_2 e''_k) \quad (6)$$

$$G_2 := \sqrt{\alpha_0^2 + \beta_0^2 + \gamma_0^2} \quad (7)$$

In (6), e'_p and e''_p denote the path length error and the angular distance error, respectively; e'_k and e''_k denote the module values of position error and attitude error at waypoint k , respectively. λ_1 makes a trade-off between the path length error and the angular distance error, while λ_2 makes a trade-off between the position error and the attitude error at waypoints. In (7), α_0 , β_0 and γ_0 denote the Euler angles around Z_0 -, Y_0 - and X_0 -axis, respectively. Besides, joint angles and joint angular velocities should be within $[-180 \text{ deg}, 180 \text{ deg}]$ and $[-1 \text{ deg/s}, 1 \text{ deg/s}]$, respectively.

3. Solving Method of MWPP for FFSR

3.1 HPSO

Inspired by birds flock looking for food, Kennedy and Eberhart proposed the Particle Swarm Optimization (PSO) algorithm. The mathematical equation is given by

$$\begin{cases} v_i^{m+1} = \omega v_i^m + c_1 r_1 (p_i^m - x_i^m) + c_2 r_2 (g_i - x_i^m) \\ x_i^{m+1} = x_i^m + v_i^{m+1} \end{cases} \quad (8)$$

In (8), v_i^m and x_i^m denote the velocity and the position of particle m at iteration i , respectively. ω , c_1 and c_2 denote the inertia, the cognitive and the social weights, respectively. r_1 and r_2 denote two random numbers that are distributed uniformly in $[0, 1]$. p_i^m denotes the best position of particle m at iteration i . g_i denotes the global best position of the whole swarm at iteration i so far.

Due to the search mechanism, PSO is easy to get trapped in local optimum. To solve the problem, the three

control parameters in HSPSO are variables, formulated in (9), (10) and (11). ω_i^m , $c_{1_i}^m$ and $c_{2_i}^m$ denote the control parameters of particle m at iteration i . ω_s , c_{1_s} and c_{2_s} denote the corresponding initial values, while ω_f , c_{1_f} and c_{2_f} denote the corresponding final values. T denotes the maximum iteration. $gp_m := \|g_i - p_i^m\|$ denotes the distance between personal best position of particle m and global best position at iteration i .

$$\omega_i^m = (\omega_s - \omega_f) \cdot \exp\left(-\frac{\omega_s - \omega_f}{gp_m} \cdot \frac{i}{T}\right) + \omega_f \quad (9)$$

$$c_{1_i}^m = (c_{1_s} - c_{1_f}) \cdot \exp\left(-\frac{c_{1_s} - c_{1_f}}{gp_m} \cdot \frac{i}{T}\right) + c_{1_f} \quad (10)$$

$$c_{2_i}^m = (c_{2_s} - c_{2_f}) \cdot \exp\left(-\frac{c_{2_s} - c_{2_f}}{gp_m} \cdot \frac{i}{T}\right) + c_{2_f} \quad (11)$$

According to the dynamic system theory, if the magnitude of solutions, *i.e.* eigenvalues of coefficient matrix in system (8), is less than 1, the system is stable. Then, we can get the convergence region of the system, $\{\omega < 1, c_1 r_1 + c_2 r_2 > 0, 4\omega - 4(c_1 r_1 + c_2 r_2) + 4 > 0\}$. According to (9)–(11), gp_m influences control parameters, thus influencing convergence and convergent behaviour of the system. Moreover, initial parameters of control parameters should follow convergence region of the system. Based on the above analysis, the parameters are defined as follows: $\omega_s = 0.96$; $\omega_f = 0.1$; $c_{1_s} = c_{2_f} = 1.8$; $c_{2_s} = c_{1_f} = 0.1$; $T = 400$, and ξ in (4) is set to 2.

HSPSO, combining advantages of PSO and GA, avoids particles getting stagnated [12]–[14]. Main steps of HSPSO are followed.

Step 1. Initialize parameters, including ω_s , ω_f , c_{1_s} , c_{1_f} , c_{2_s} , c_{2_f} , T , p_1 , p_2 and ε_1 , where p_1 , p_2 and ε_1 denote the crossover probability, mutation probability and predefined optimization precision, respectively.

Step 2. Update the fitness value according to (4).

Step 3. Update p_i^m and g_i .

Step 4. Update ω_i^m , $c_{1_i}^m$, and $c_{2_i}^m$ according to (9), (10) and (11), respectively.

Step 5. Update v_i^{m+1} and x_i^{m+1} according to (8).

Step 6. Perform crossover and mutation according to the probabilities p_1 and p_2 , respectively.

Step 7. Examine the terminal condition, $i > T$ or $F < \varepsilon_1$. If the terminal condition is satisfied, the iteration ends, else return to *Step 2*.

3.2 Solving Method of Optimal Joint-Movements Problem

3.2.1 Piecewise-Sine Functions for Joint-Movements

The path, along which end-effector visits waypoints, consists of several sub-paths. For each sub-path, joint-movements are depicted with sine functions:

$$q_k^K(t) = \Delta \cdot \sin(a_{k3}^K t^3 + a_{k2}^K t^2 + a_{k1}^K t + a_{k0}^K) \quad (12)$$

In (12), $K = 1, \dots, N-1$ and $k = 1, \dots, n$, where N and n denote the number of waypoints and the number of joints of FFSR, respectively. $q_k^K(t)$ denotes the movements of joint k , corresponding to sub-path K along which end-effector moves from waypoint K to waypoint $K+1$. Δ denotes the physical limits on joint angles, which is defined as 180° . a_{ki}^K ($i = 0, 1, 2, 3$) denotes the coefficients of polynomials. Moreover, t is from initial time t_s^K to final time t_f^K of sub-path K . To ensure continuity of joint-movements, FFSR has the same joint configuration at the end of sub-path K and at the start of sub-path $K+1$:

$$q_k^K(t_f^K) = q_k^{K+1}(t_s^{K+1}) \quad (13)$$

Based on (12), joint angular velocities are given by

$$\begin{aligned} \dot{q}_k^K(t) = \Delta \cdot \cos(a_{k3}^K t^3 + a_{k2}^K t^2 + a_{k1}^K t + a_{k0}^K) \cdot \\ (3a_{k3}^K t^2 + 2a_{k2}^K t + a_{k1}^K) \end{aligned} \quad (14)$$

Joint angular velocities are zero at each waypoint, given by (15). Besides, the initial joint configuration corresponding to the starting waypoint is predefined, shown in (16).

$$\dot{q}_k^K(t_s^K) = \dot{q}_k^K(t_f^K) = 0 \quad (15)$$

$$q^1(t_s^1) = [q_1^1, q_2^1, \dots, q_n^1]' \quad (16)$$

According to equations from (12) to (16), the optimal joint-movements problem is transformed into a parametric optimization problem. Substituting (13) and (15) (if $K=1$, substituting (15) and (16)) into (12) and (14), a_{k0}^K , a_{k1}^K and a_{k2}^K are derived:

$$a_{k2}^K = -\frac{3}{2}a_{k3}^K (t_s^K + t_f^K) \quad (17)$$

$$a_{k1}^K = 3a_{k3}^K t_s^K t_f^K \quad (18)$$

$$a_{k0}^K = \arcsin\left(\frac{q_k^K(t_s^K)}{\Delta}\right) - \frac{3}{2}a_{k3}^K (t_s^K)^2 (t_s^K + t_f^K) \quad (19)$$

Once a_{k3}^K ($k = 1, \dots, n$) is obtained, the joint-movements corresponding to sub-path K are obtained. The base attitude of FFSR in MWPP grows rapidly as waypoints increase. Besides, the pose error of end-effector at each waypoint has a great influence on path planning toward next waypoint. As waypoints increase, this pose error is dramatically accumulated. To solve such problems, IPSO is developed.

3.2.2 IPSO

In IPSO, the velocity and the position of each particle are expressed in (8), and the control parameters are expressed as

$$\omega_i^m = \frac{\omega_s - \omega_f}{1 + \exp\left(\beta \cdot \left(i - \frac{T \cdot (1 + \ln(\beta))}{\mu_1}\right) \cdot \mu_2\right)} + \omega_f \quad (20)$$

$$c_{1_i}^m = \frac{c_{1_s} - c_{1_f}}{1 + \exp\left(\beta \cdot \left(i - \frac{T \cdot (1 + \ln(\beta))}{\mu_1}\right) \cdot \mu_2\right)} + c_{1_f} \quad (21)$$

$$c_{2_i}^m = \frac{c_{2_s} - c_{2_f}}{1 + \exp\left(\beta \cdot \left(i - \frac{T \cdot (1 + \ln(\beta))}{\mu_1}\right) \cdot \mu_2\right)} + c_{2_f} \quad (22)$$

In (20), (21) and (22), each argument of β , μ_1 and μ_2 has different constant values during optimization processes of sub-paths. In this part, parameters are defined as follows: $\omega_s = 0.94$, $\omega_f = 0.4$, $c_{1_s} = c_{2_f} = 2.5$, $c_{1_f} = c_{2_s} = 0.5$, $T = 100$. β , μ_1 and μ_2 vary from 0.35 to 0.45, from 0.1 to 0.2, and from 0.15 to 0.35, respectively. Main steps of IPSO for optimal joint-movements are followed.

Step 1. Initialize parameters I, the initial joint configuration corresponding to the waypoint K ($K = 1, \dots, N - 1$) and the desired pose for the waypoint $K + 1$.

Step 2. Examine the terminal condition I, if $K \geq N$ (the number of waypoints), the iteration ends.

Step 3. Initialize parameters II, including ω_s , ω_f , c_{1_s} , c_{1_f} , c_{2_s} , c_{2_f} , T , and ε_2 .

Step 4. Update fitness values according to (5), (6) and (7).

Step 5. Examine the terminal condition II, if $G < \varepsilon_2$, return to *Step 1*, else go to *Step 6*.

Step 6. Update p_i^m and g_i .

Step 7. Update ω_i^m , $c_{1_i}^m$ and $c_{2_i}^m$ according to (20), (21) and (22), respectively.

Step 8. Update v_i^{m+1} and x_i^{m+1} according to (8).

Step 9. $K = K + 1$ and return to *Step 4*.

4. Simulation Results

Simulations of HSPSO for optimal waypoint-sequence and IPSO for optimal joint-movements are developed.

4.1 Simulations of HSPSO for Optimal Waypoint-Sequence

4.1.1 Comparisons

With different numbers of waypoints, GA, SIACO [15] and HSPSO are employed to search for optimal solution of (4). Figure 1 presents the curve trends of the fitness value and the CPU time of GA, SIACO and HSPSO. The following conclusions can be derived: (i) HSPSO provides better fitness values than GA when the number of waypoints is more than 3. The difference between fitness values of HSPSO and of GA is 3.2987 when the number is 5. As waypoints increase, the difference becomes larger, and the difference is 51.4586 when the number is 30. (ii) SIACO provides worse fitness values than HSPSO. The difference grows as waypoints increase, and the difference is larger than 10 when the number is more than 19. (iii) As waypoints increase, CPU time of SIACO grows fast. Compared with SIACO, CPU time of HSPSO grows more slowly.

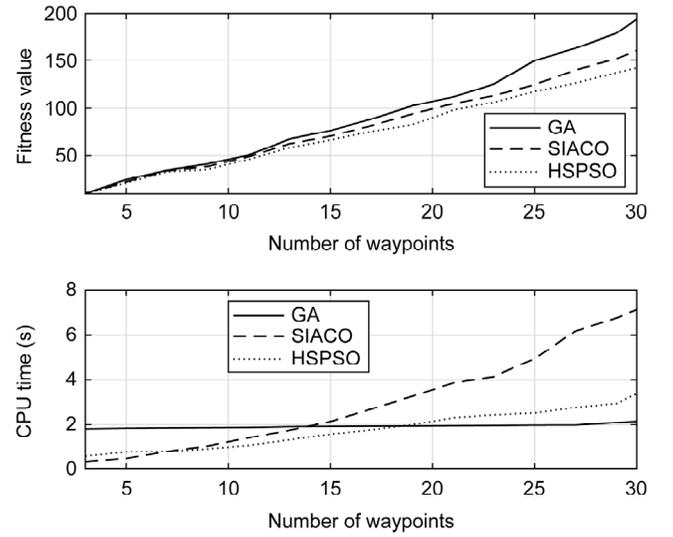


Figure 1. Fitness value and CPU time of GA, SIACO and HSPSO.

4.1.2 Specific Case

The end-effector should visit six waypoints, which are defined as follows: $A: (-4.77, 4.68, 8.07, 11.9, -28.2, 24.7)$; $B: (-4.30, 5.00, 8.50, 15.0, -25.0, 20.0)$; $C: (-4.00, 4.15, 8.20, 5.0, -20.0, 25.0)$; $D: (-3.68, 4.68, 8.62, 10.0, -20.0, 24.0)$; $E: (-4.10, 4.25, 7.92, 15.0, -15.0, 20.0)$; and $F: (-3.90, 4.85, 8.80, 25.0, -15.0, 30.0)$. The units of position and attitude are meters and degrees, respectively. In the aforementioned six waypoints, A is the starting point. Using HSPSO, the objective function (4) is optimized and the optimal waypoint-sequence is derived: $A \rightarrow B \rightarrow F \rightarrow D \rightarrow C \rightarrow E$.

4.2 Simulations of IPSO for Optimal Joint-Movements

Comparisons of six algorithms are studied first. Then, based on the optimal waypoint-sequence in Section 4.1.2, movements of FFSR are developed. The FFSR employed for simulation has seven rotational joints and is redundant [6], [16]. Physical parameters of FFSR refer to [6], and the initial joint configuration is predefined as $[120, 30, -45, 90, 150, -30, 60]^T$ (deg), corresponding with the starting point A .

4.2.1 Comparisons

Comparisons of IPSO with five other algorithms are developed. The five algorithms include PSO, another Improved PSO (called IPSO-2), Multi-verse Optimization algorithm (MVO) [17], Grasshopper Optimization Algorithm (GOA) [18], and Moth-flame Optimization algorithm (MFO) [19]. MVO, GOA and MFO are, three novel meta-heuristic algorithms, employed for solving global optimization problems in fields like path planning and controlling.

The differences of PSO, IPSO and IPSO-2 are updating mechanisms of control parameters. In PSO, all control

Table 1
Parameters of IPSO, IPSO-2 and PSO

| | ω | ω_s | ω_f | c_1 | c_{1_s} | c_{1_f} |
|--------|----------|------------|------------|----------|------------|------------|
| IPSO | (15) | 0.94 | 0.4 | (16) | 2.5 | 0.5 |
| IPSO-2 | (15) | 0.94 | 0.4 | (16) | 2.5 | 0.5 |
| PSO | 0.7 | 0.7 | 0.7 | 2.0 | 2.0 | 2.0 |
| | c_2 | c_{2_s} | c_{2_f} | μ_1 | μ_2 | |
| IPSO | (17) | 0.5 | 2.5 | $M_1(K)$ | $M_2(K)$ | |
| IPSO-2 | (17) | 0.5 | 2.5 | 0.15 | 0.25 | |
| PSO | 2.0 | 2.0 | 2.0 | / | / | |

parameters are constants during the whole optimization process. In IPSO and IPSO-2, control parameters are updated based on (20), (21) and (22). The differences of IPSO and IPSO-2 are the selection of arguments μ_1 and μ_2 . In IPSO, μ_1 and μ_2 have different constant values when the end-effector moves between different waypoints. μ_1 and μ_2 are fine tuned, based on the value of K , position and attitude of initial and final waypoints of sub-path K , and path length of sub-path K . While in IPSO-2, μ_1 and μ_2 are constants. Parameters of IPSO, IPSO-2 and PSO are presented in Table 1, where $M_1=[0.15, 0.15, 0.147, 0.14, 0.18, 0.15]^T$ and $M_2=[0.25, 0.35, 0.50, 0.57, 0.25, 0.25]^T$. For sub-path K , μ_1 and μ_2 are selected as $M_1(K)$ and $M_2(K)$, respectively. Parameters of MVO, GOA and MFO refer to [17]–[19]. Moreover, parameters in (5) and (6) are defined as $\zeta = \lambda_1 = \lambda_2 = 2$.

Main steps of IPSO are presented in Section 3.2.2. Figure 2 presents fitness curves of six algorithms.

The optimal fitness values of IPSO, IPSO-2, PSO, MVO, GOA and MFO are 0.0975, 3.3172, 9.3374, 4.7408, 4.0153 and 10.7507, respectively. It can be seen that IPSO has a better optimization result than the other five algorithms. Therefore, the proposed updating mechanism of control parameters, during the optimization process for each sub-path, is valid. Besides, fine-tuning updating mechanisms of control parameters, among optimization processes for different sub-paths, contributes to optimization results with high accuracy.

4.2.2 Specific Case

Movements of FFSR for the specific case studied in Section 4.1.2 are developed in this part by IPSO, which can be seen in Fig. 3. (a) and (b) present the position and attitude variation of end-effector, respectively. α_{EE} , β_{EE} and γ_{EE} denote the Euler angles around Z_{E-} , Y_{E-} and X_{E-} axis, respectively. The end-effector visits each waypoint with high accuracy. It can also be seen that the sub-paths are optimum but not the minimum, as the objective of the second sub-problem considers both the optimal length of end-effector and the minimal attitude disturbance on base. Coupling between manipulator and base is also a factor. However, the error between the optimum and the minimum is minimized, without conflicting with the result of the first sub-problem. (c) presents the attitude trajectory of the free-floating base, where α_0 , β_0 and γ_0 denote the Euler angles. The magnitude of attitude disturbance on the free-floating base is 0.01 (deg). (d) and (e) present the angular and angular-velocity trajectories of seven joints, showing that all joints move smoothly. The joint angles and the joint angular velocities are within the predefined ranges of $[-180 \text{ deg}, 180 \text{ deg}]$ and $[-1 \text{ deg/s}, 1 \text{ deg/s}]$, respectively. Besides, the angular velocity of each joint is zero at each waypoint.

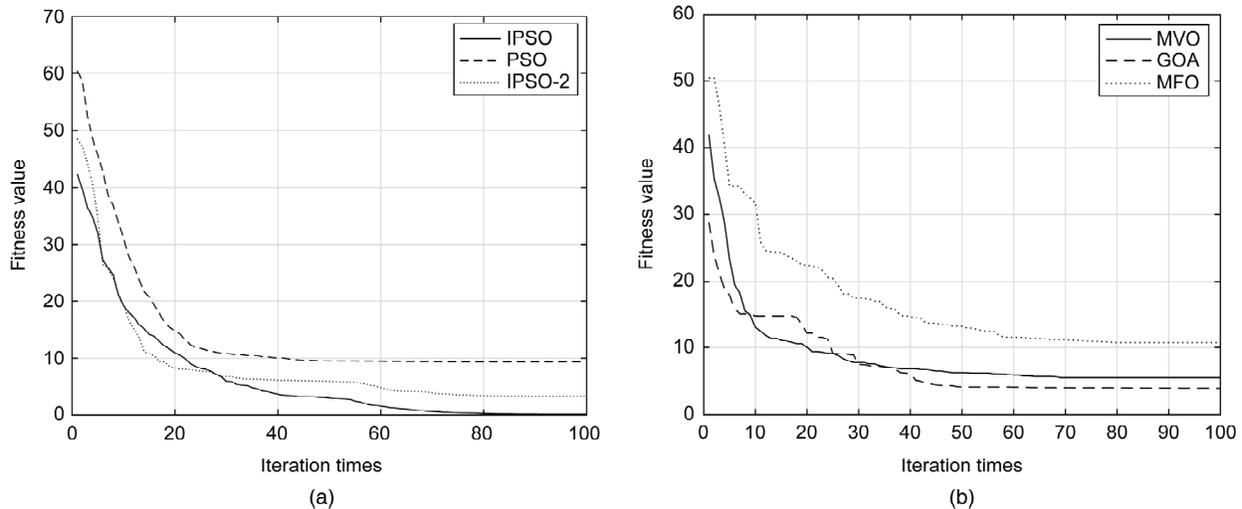


Figure 2. Iterative evolution of the six algorithms.

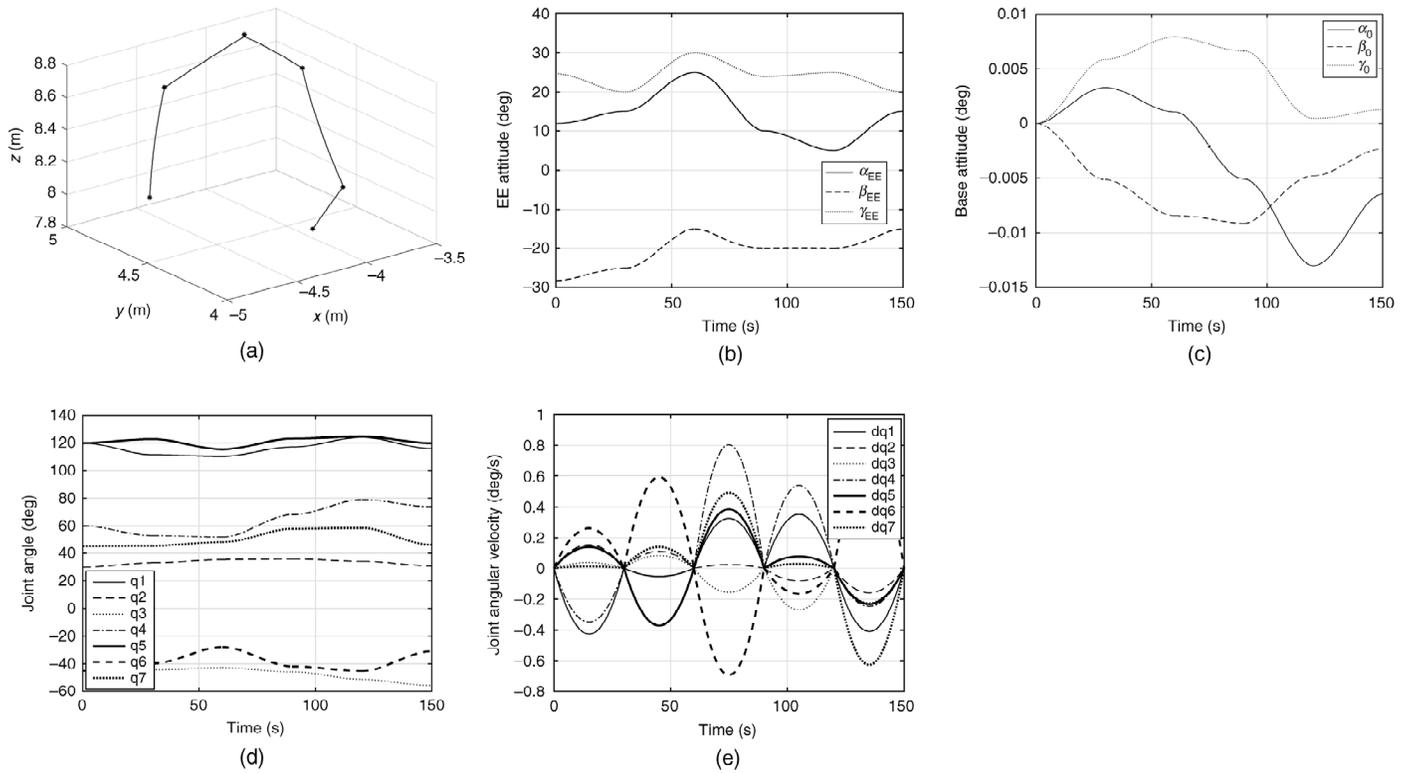


Figure 3. Time histories of FFSR movements.

5. Conclusion

The main contribution of this work is the study of MWPP for FFSR, which is decomposed into two sub-problems: the optimal waypoint-sequence problem and the optimal joint-movements problem. Firstly, HSPSO is employed to find optimal waypoint-sequence. Then, IPSO is employed for optimal joint-movements with minimum attitude disturbance acting on the free-floating base. Simulation results validate the developed approach. Future research will focus on improving HSPSO and IPSO. Although the two algorithms can improve optimization accuracy, CPU time gets longer as waypoints increase.

Acknowledgement

This work was supported by the National Natural Science Foundation of People's Republic of China (No. 11472213).

References

- [1] L. Wang and C. Luo, A hybrid genetic tabu search algorithm for mobile robot to solve AS/RS path planning, *International Journal of Robotics and Automation*, 33(2), 2018, 161–168.
- [2] B. Hao and Z. Yan, Recovery path planning for an agricultural mobile robot by Dubins-RRT* algorithm, *International Journal of Robotics and Automation*, 33(2), 2018, 202–207.
- [3] L. Wang, C. Luo, M. Li, and J. Cai, Trajectory planning of an autonomous mobile robot by evolving ant colony system, *International Journal of Robotics and Automation*, 32(4), 2017, 406–413.
- [4] J. Ni, K. Wang, Q. Cao, Z. Khan, and X. Fan, A memetic algorithm with variable length chromosome for robot path

- planning under dynamic environments, *International Journal of Robotics and Automation*, 32(4), 2017, 414–424.
- [5] P. Huang, Y. Xu, and B. Liang, Minimum-torque path planning of space robots using genetic algorithms, *International Journal of Robotics and Automation*, 21(3), 2006, 229–236.
- [6] X. Liu, H. Baoyin, and X. Ma, Optimal path planning of redundant free-floating revolute-jointed space manipulators with seven links, *Multibody System Dynamics*, 29(1), 2013, 41–56.
- [7] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control* (London: Springer-Verlag, 2009).
- [8] B. Siciliano and O. Khatib, *Springer handbook of robotics* (Berlin: Springer-Verlag, 2008).
- [9] J. Zhang, H. Lv, D. He, L. Huang, Y. Dai, and Z. Zhang, Discrete bioinspired neural network for complete coverage path planning, *International Journal of Robotics and Automation*, 32(2), 2017, 186–193.
- [10] L. Li, X. Wang, D. Xu, and M. Tan, An accurate path planning algorithm based on triangular meshes in robotic fibre placement, *International Journal of Robotics and Automation*, 32(1), 2017, 22–32.
- [11] J. Craig, *Introduction to robotics: mechanics and control* (NJ, USA: Pearson, 2005).
- [12] H. Liu, Z. Cai, and Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Applied Soft Computing*, 10(2), 2010, 629–640.
- [13] X. Zhao, A perturbed particle swarm algorithm for numerical optimization, *Applied Soft Computing*, 10(1), 2010, 119–124.
- [14] P. Chauhan, K. Deep, and M. Pant, Novel inertia weight strategies for particle swarm optimization, *Memetic Computing*, 5(3), 2013, 229–251.
- [15] G. Reinelt, *The traveling salesman: computational solutions for TSP applications* (Berlin: Springer-Verlag, 1994).
- [16] F. Caccavale and B. Siciliano, Kinematic control of redundant free-floating robotic systems, *Advanced Robotics*, 15(4), 2001, 429–448.

- [17] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*, 27(2), 2016, 495–513.
- [18] S. Saremi, S. Mirjalili, and A. Lewis, Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software*, 105, 2017, 30–47.
- [19] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, 89, 2015, 228–249.

Biographies



Suping Zhao is studying Ph.D. in flight vehicle design in Northwestern Polytechnical University, Xi'an, China. Her current research includes space robotics and unmanned aerial vehicles.



Alejandro Gutiérrez-Giles is a post-doctoral researcher in CINVESTAV-IPN, Department of Mechatronics. His current research includes robotics, control and observers.



Jianjun Luo is a professor of flight vehicle design in Northwestern Polytechnical University, Xi'an, China. He is a co-editor of National Key Laboratory of Aerospace Flight Dynamics. His current research includes flight dynamics and space robotics.



Bruno Siciliano is a director of the Interdepartmental Center for Advances in Robotic Surgery (ICAROS) as well as coordinator of the Laboratory of Robotics Projects for Industry, Services and Mechatronics (PRISMA Lab), at University of Naples Federico II. He is a fellow of the scientific societies IEEE, ASME, IFAC, he received numerous international prizes and awards, and he was

President of the IEEE Robotics and Automation Society from 2008 to 2009. Since 2012, he is on the Board of Directors of the European Robotics Association. He has delivered more than 150 keynotes and has published more than 300 papers and 7 books. His book “Robotics” is among the most adopted academic texts worldwide, while his edited volume “Springer Handbook of Robotics” received the highest recognition for scientific publishing: 2008 PROSE Award for Excellence in Physical Sciences & Mathematics. His research team got 20 projects funded by the European Union for a total grant of 12 M€ in the last ten years, including an Advanced Grant from the European Research Council.



Zhanxia Zhu is a professor of flight vehicle design in Northwestern Polytechnical University, Xi'an, China. She is an editor of the *Journal of Aerospace Science and Technology*. Her current research includes flight dynamic and control, orbit design of spacecraft, and ground microgravity experiment.