Multiobjective preimpact trajectory planning of space manipulator for self-assembling a heavy payload

Yong Liu, Zhe Du, Zhe Wu, Fei Liu and Xiaojun Li

Abstract
To assemble a heavy payload to the spacecraft (free-floating base), the present study proposes a scheme of multiobjective trajectory planning for preimpact motion of redundant space manipulator (mounted on the base). Force impulse for self-assembly is derived as the function of joint angles/velocities, base pose, and impact direction. The trajectory planning problem is formulated as multiobjective optimization to minimize force impulse, base attitude disturbance, and energy consumption in the load-carrying process. A two-stage trajectory planning algorithm is proposed. To be specific, at the first stage, multiple desired configurations at the contact point are generated by position-level inverse kinematics with Newton–Raphson iterative method. At the second stage, joint trajectories satisfying joint angle limits and desired motion of the payload are parameterized by coefficients of sinusoidal polynomial functions. Multiobjective particle swarm optimization algorithm is adopted to solve the problem of multiobjective trajectory planning, and screening process is conducted to reserve nondominated solutions in limits of joint torques. The algorithm is implemented to a seven degrees of freedom space manipulator, and the effectiveness of the proposed method is verified by simulation results.

Keywords
Space manipulator, self-assembly task, heavy payload, trajectory planning, multiobjective particle swarm optimization

Date received: 21 August 2019; accepted: 17 December 2020

Introduction
Space manipulators are increasingly critical to future on-orbit servicing, especially to capture, assemble, and maintain large-space structures. After capturing a heavy payload, the space manipulator mounted on the spacecraft base is employed to assemble payload to the base. In the preassembly operation, space manipulator should carry the payload from capture pose (including position and orientation) to preassembly pose. However, for the dynamic coupling effect between space manipulator and spacecraft, attitude disturbance of the base induced by heavy payload should be considered in the motion planning process. Existing studies considering payload constraints were conducted for ground manipulators. For on-orbit load-carrying operation, a multiobjective optimization-based approach was proposed to achieve the maximum-payload trajectory planning of free-floating space manipulator (FFSM). Dalla and Pathak proposed a method for multiple space manipulators to perform a docking task in the control of base disturbance. However, impact problem has not been considered in the mentioned studies.

Corresponding author:
Zhe Wu, School of Mechanical Engineering, Hefei University of Technology, No. 193 Tunxi Road, Baohe District, Hefei 230009, China. Email: wuzhe@hfut.edu.cn
For space manipulators, the effect exerted by impact force should be analyzed. Yoshida et al. divided the capture mission into three phases and proposed extended generalized inertia tensor to formulate collision dynamics.\(^9\) Chen et al. explored the impact dynamics and the motion after impact.\(^10\) Huang et al. derived the force and momenta transmission from the end-effector to the base under impact between the end-effector and the target.\(^11,12\) For the trajectory planning problem involving impact force influence, Wee et al. developed a joint space planning strategy that achieves both trajectory tracking and impact minimization.\(^13\) Zhang et al. investigated a desired configuration at the contact point; subsequently, they proposed a scheme of preimpact trajectory planning to achieve minimized base attitude disturbance caused by impact,\(^14\) whereas the results were achieved by depending on a specified base pose. In the present study, base disturbance in preimpact motion is not considered, which cannot be neglected when carrying a heavy payload. Cong and Zhang analyzed a dual-arm space manipulator and proposed the “straight arm capture” concept, which is not suitable for self-assembly.\(^15\) However, the mentioned studies were conducted for capture tasks, in which impact influences on space manipulator were exerted by an external force, and impact force displayed the only relationship to the configuration of the manipulator. For impact process of self-assembly, the collision between the base and the payload should be considered as internal force. Base suffers directly impact the assembly mechanism and the disturbance force transmitted from payload to the base. When carrying a heavy payload, the force impulse can be extremely large, which may damage payload, spacecraft, and manipulator. Large disturbance to the base may take place for the nonholonomic characteristics,\(^16\) which is not expected since the requirements of communication and observation. Moreover, for large mass and inertia tensor of payload, energy consumption involved in load-carrying operation can significantly increase. Given task constraints and limited capacity of space manipulator, multiple objectives and constraints should be considered. Misra and Bai and Luo et al. adopted quadratic programming method to optimize base disturbance and avoid collision.\(^17,18\) Similar method was employed for local optimization of reaction torque and joint torque simultaneously.\(^19\) The task-priority-based null-space trajectory planning problems were studied for redundant space manipulator.\(^20-22\) Moreover, evolutionary algorithms were adopted to perform optimization, such as multiswarm particle swarm optimization (PSO),\(^16\) differential evolution algorithm,\(^23\) and genetic algorithm.\(^24\) In the mentioned studies, trajectory planning was altered to a single-objective problem, and weighting coefficient exists in cost functions to determine the weight of each objective. Wang et al. explored the application of PSO to trajectory planning and discussed nondominated solutions.\(^25\) The Pareto Front was obtained by mapping the solutions of single-objective optimization to objective space, multiobjective evolutionary computation and dominance relations were uninvolved. Multiobjective optimization was introduced for mobile and industrial manipulators,\(^26,27\) which also applies to trajectory planning of space manipulator.

In the present study, a scheme of multiobjective preimpact trajectory planning is proposed for self-assembly task of space manipulator when carrying a heavy payload. The rest of this study is organized as follows: The second section analyzes the force impulse for self-assembly task. In the third section, how to formulate trajectory planning as a multiobjective constrained optimization problem is discussed. The fourth section illustrates the two-stage algorithm that combines of solving multiple configurations and multiobjective evolutionary computation. The fifth section presents the simulation results of a seven degrees of freedom (7-DOF) space manipulator. The conclusions of the work are presented in the sixth section.

### Modeling of free-floating space manipulator for self-assembling a heavy payload

#### Mathematical model of space manipulator with a payload

Figure 1 shows the general model of a free-floating space manipulator that consists of a base, a \(n-\)DOF revolute-jointed manipulator, and a payload. The symbols and variables applied for modeling of the system in this section are listed in Table 1.

The payload is attached to the end-effector with rigid connection; thus, link 1 and payload can be considered a single composite rigid body. Mass and inertia tensor of the composite body can be expressed as

\[
\begin{align*}
    m_c &= m_n + m_L \\
    \mathbf{I}_c &= \mathbf{I}_n + m_L \mathbf{P}_L \\
    \mathbf{I}_L &= \mathbf{I}_L + m_L \left( \mathbf{P}_L \mathbf{I}_n \mathbf{P}_L^T - \mathbf{P}_L \mathbf{P}_L^T \right) \mathbf{I}_L
\end{align*}
\]

where \(m_c\) denotes the mass of the combined body; \(\mathbf{I}_L\) and \(\mathbf{I}_L\) represent inertial tensors of payload in \(\Sigma_L\) and \(\Sigma_n\), respectively.
Table 1. Symbols applied in modeling of FFSM.

| Symbols | Representation |
|---------|----------------|
| $m_i, m_l$ | Mass of link $i (i = 1, 2, \ldots, n)$ and the payload |
| $\Sigma_i$ | Inertia frame |
| $\Sigma_B$ | Base frame |
| $\Sigma_L$ | Payload frame |
| $\Sigma_i$ | Link $i$ frame |
| $C$ | Center of mass |
| $x_b$ | Pose of the base in inertial frame |
| $x_L$ | Pose of payload in inertial frame |
| $q_m = [q_1, q_2, \ldots, q_m]^T$ | Joint vector |
| $\tau_m = [\tau_1, \tau_2, \ldots, \tau_m]^T$ | Joint torque vector |
| $J_b$ | Jacobian matrix of the base |
| $J_m$ | Jacobian matrix of manipulator |
| $F_b$ | External force on the base |
| $F_L$ | External force on payload |
| $H_b$ | Inertia matrix of the base |
| $H_m$ | Inertia matrix of manipulator |
| $H_{ibm}$ | The coupling inertia matrix |
| $c_0 \in \mathbb{R}^{6 \times 1}$ | Nonlinear terms of the base |
| $c_m \in \mathbb{R}^{n \times 1}$ | Nonlinear terms of manipulator |

FFSM: free-floating space manipulator.

respectively; $R_L$ is the rotation matrix of $\Sigma_L$ with respect to $\Sigma_n$; and $E_3$ is the third-order unit matrix.

Next, the dynamics equation of space manipulator can be formulated as

$$
\begin{bmatrix}
H_b & H_{bm} \\
H_{bm}^T & H_m
\end{bmatrix}
\begin{bmatrix}
\ddot{x}_b \\
\ddot{q}_m
\end{bmatrix}
+ \begin{bmatrix}
c_b \\
c_m
\end{bmatrix}
= \begin{bmatrix}
F_b \\
\tau_m
\end{bmatrix}
+ \begin{bmatrix}
J_b^T \\
J_m^T
\end{bmatrix} F_L
$$

(2)

where the inertia and Jacobian matrices are calculated considering payload parameters by equation (1).

Under the free-floating condition, the total momentum of the system is conserved, as expressed by

$$
L_0 = H_b \ddot{x}_b + H_{bm} \ddot{q}_m
$$

(3)

The initial momentum is assumed as $L_0 = 0$, and then, the motion relationship of the base and the manipulator is expressed as

$$
\ddot{x}_b = J_{bm} \ddot{q}_m
$$

(4)

where $J_{bm} = -H_b^{-1} H_{bm}$ is coupling Jacobian matrix. For the coupling motion of the manipulator and the base, trajectory planning of joints displays an association with the stability of the base.

Substitute equation (4) to the following general kinematics equation of FFSM

$$
\ddot{x}_L = J_m \ddot{q}_m + J_b \ddot{x}_b = J_{\text{float}} \ddot{q}_m
$$

(5)

where $J_{\text{float}} = J_m - J_b H_b^{-1} H_{bm}$ denotes the generalized Jacobian matrix. With respect to equations (4) and (5), motion of the payload and the base is decided by joint rotation. With appropriate planning of joint trajectories, the assembly of a heavy payload and stability of the base can be simultaneously achieved.

**Force impulse analysis for self-assembly**

At the contact point, the following collision assumptions are made to develop the force impulse function:

1. The interaction duration is significantly short, so the contact force acts instantaneously.
2. The pose changes of payload in the impact process are negligible, and the effects of other forces except for the contact force can be ignored.
3. Based on the mentioned assumptions, the inertia term of dynamic equation is dominant and other terms are disregarded.
4. Impulsive forces and moments are induced on an act–react principle at the single-point of contact.

Next, equation (2) can be reformulated as

$$
H_b \ddot{x}_b + H_{bm} \ddot{q}_m + c_b = F_b + J_b^T F_L
$$

$$
H_{bm} \ddot{x}_b + H_{bm} \ddot{q}_m + c_m = \tau_m + J_m^T F_L
$$

(6)

$\ddot{x}_b$ is eliminated in equation (6), and the definition is made; it can be therefore expressed as

$$
H^* \ddot{q} + c^* = \tau_m + J_{\text{float}}^T F_L + J_{\text{base}}^T F_b
$$

(7)

The matrices in equation (7) are listed as follows

$$
H^* = H_m - H_{bm} H_b^{-1} H_{bm}
$$

$$
c^* = c_m - H_{bm} H_b^{-1} c_b
$$

(8)

When ignoring centrifugal force and Coriolis force, the inertia term of dynamics equation is dominant. Equation (7) can be simplified as

$$
H^* \ddot{q}_m = J_{\text{float}}^T F_L + J_{\text{base}}^T F_b
$$

(9)

Define $F_T$ as the contact force, force impulse is formulated as

$$
\Phi = \int_{t_0}^{t_0 + \delta t} F_T dt
$$

(10)

At the contact point, the contact force acts payload and base are assumed as $F_T$ and $-F_T$, respectively. At the impact time $t_c$, it can be equivalently considered that $F_L(t_c) = F_T$ and $F_b(t_c) = -F_T$. Then, we have

$$
\begin{align*}
\Phi_L &= \int_{t_0}^{t_0 + \delta t} F_L dt = \Phi \\
\Phi_b &= \int_{t_0}^{t_0 + \delta t} F_b dt = -\Phi
\end{align*}
$$

(11)

The integral formula above is taken in a very short time $\delta t$, and $\delta q_m = q_m(t_0 + \delta t) - q_m(t_0)$ is substituted into the integral. Then, it yields

$$
H^* \delta q_m = J_{\text{float}}^T \Phi_L + J_{\text{base}}^T \Phi_b = J_{\text{float}}^T \Phi - J_{\text{base}}^T \Phi
$$

(12)
So that
\[ \delta \dot{q}_m = (H^*)^{-1} J_e \Phi \]  
where \( J_e = (J_{\text{float}}^T - J_{\text{float}}^T J_e) \).

The integral of equation (5) in \( \delta t \) is taken as
\[ \delta \dot{x}_L = J_{\text{float}}(H^*)^{-1} J_e \Phi \]  
Define \( D_1 = J_{\text{float}}(H^*)^{-1} J_e \), and then, \( \delta \dot{x}_b = D_1 n \eta \).

According to equation (4), the motion of the base is calculated by
\[ \dot{x}_b = J_{\beta m} J_{\text{float}}^T \dot{x}_L \]  
Substituting equation (14) into the integral of equation (16), it easily yields
\[ \delta \dot{x}_b = J_{\beta m}(H^*)^{-1} J_e \Phi \]  
Define \( D_2 = -J_{\beta m}(H^*)^{-1} J_e \), and then, \( \delta \dot{x}_b = D_2 n \eta \).

The collision consists of stages of compression and rebound. At the end of the compression stage, the relative velocity of contact surface between the base and the payload is zero, which is expressed as
\[ n^T [\dot{x}_b + \delta \dot{x}_b] - (\dot{x}_L + \delta \dot{x}_L) = 0 \]  
Substituting equations (15) to (17) into equation (18), the force impulse in the self-assembly process is defined as
\[ \eta = -\frac{n^T (J_{\beta m} J_{\text{float}}^T - E) \dot{x}_L}{n^T (D_1 + D_2) J_e n} \]  
Since the relative motion of the base and the payload is attributed to the joint rotation, the given relative velocity of self-assemble task can be described as
\[ \Delta \dot{x} = J_m (\dot{q}_m(t_c)) \dot{q}_m(t_c) \]  
where \( \dot{q}_m(t_c) \), \( \dot{q}_m(t_c) \) denote joint angles and velocities at the impact time, respectively.

Take Poisson model in collision and set the restitution coefficient as \( e \), the total impulse is considered as
\[ \eta_c = -(1 + e) \frac{n^T (J_{\beta m} J_{\text{float}}^T - E) \Delta \dot{x}}{n^T (D_1 + D_2) J_e n} \]  
We can establish the force impulse function as
\[ \eta_c = f(q_m(t_c), \dot{q}_m(t_c), x_b(t_c), n) \]  
where \( q_m(t_c) \), \( n \), and \( \Delta \dot{x} \) are determined according to the task. However, due to the dynamics coupling effect, \( x_b(t_c) \) relies on the history motion of the joints.

Multiobjective optimization problem for trajectory planning

The purpose of this study is to obtain appropriate joint trajectories, which accomplish self-assembly operation of a heavy payload and maintain stability of the whole system. Hence, trajectory planning problem exhibits the following characteristics:

1. While the payload pose reaches the contact point, the relative motion of payload and base should satisfy the assembly requirements.
2. Impulse force should be minimized to avoid damaging any part of the multibody system.
3. Due to load-carrying capacity constraint of space manipulator, disturbance of the base, driving torque constraints of the joints, and total energy of the system should be considered.

Multiobjective optimization problem statement of preimpact trajectory planning

Given the mentioned analysis, preimpact trajectory planning of FFSM can be solved as a constrained multiobjective optimization problem (MOP) as follows
\[
\min y = f(\lambda) = \{f_1(\lambda), f_2(\lambda), \ldots, f_M(\lambda)\} \\
\text{s.t.} \ g(\lambda) \leq 0, \ h(\lambda) = 0
\]  
where \( \lambda \) denotes the decision vector that expresses joint trajectories; \( y = [f_1, f_2, \ldots, f_M] \) indicates the \( M \)-dimensional objective vector, each element of which represents an objective function; \( g \) and \( h \) are equality and inequality constrained conditions, respectively.

The initial constraints of space manipulator are defined as
\[
\begin{align*}
& h_1(\lambda) = q(t_0) - q_{\text{ini}} \\
& h_2(\lambda) = \dot{q}(t_0) - \dot{q}_{\text{ini}} \\
& h_3(\lambda) = \ddot{q}(t_0) - \ddot{q}_{\text{ini}}
\end{align*}
\]  
where \( t_0 \) represents the initial time and \( q_{\text{ini}} \) is initial joint angle vector.

The final constraints at the contact point are written as
\[
\begin{align*}
& h_4(\lambda) = x_L(t_f) - q_{\text{ini}} \\
& h_5(\lambda) = \Delta \dot{x}(t_f) - \ddot{x}_{\text{des}} \\
& h_6(\lambda) = \dot{q}(t_f) - \dot{q}_{\text{des}}
\end{align*}
\]  
where \( x_{\text{des}} \) and \( \dot{x}_{\text{des}} \) are desired pose and velocity of the payload in base frame.

Joint angle constraints for \( t_r \in [t_0, t_f] \)
\[
\begin{align*}
& g_1(\lambda) = q_{\text{min}} - q_i(t_r) \\
& g_2(\lambda) = q_i(t_r) - q_{\text{max}}
\end{align*}
\]  
where \( [q_{\text{min}}, q_{\text{max}}] \) denotes the limit of joint \( i \).
Joint torque constraints are
\[ g_1(\lambda) = \tau_{i_{\text{min}}} - \tau_i(t) \]
\[ g_4(\lambda) = \tau_i(t) - \tau_{i_{\text{max}}} \]
where \( \tau_{i_{\text{min}}} \) and \( \tau_{i_{\text{max}}} \) respectively, denote the lower and upper torque limitations of joint \( i \).

**Definition of cost functions**

To maintain stability of the space manipulator system, force impact, attitude disturbance of the base, and total energy involved in preimpact motion should be simultaneously minimized. Thus, the following three cost functions are established.

1. Force impulse

According to equation (22), force impulse is affected by the motion of space manipulator. To ensure the safety of the system under self-assembly collision, the cost function below is considered
\[ f_1 = |\eta_C| \]  

2. Attitude disturbance of the base

Attitude disturbance of the base should be limited as impacted by the requirements of communication and observation.\(^{23}\) When carrying a heavy payload, the range of base movement could be extended; accordingly, more fuel consumption can be achieved for the maintenance of its attitude. The objective function to minimize base disturbance is given by
\[ f_2 = \int_{t_0}^{t_f} ||J_{\text{imu}} \dot{\mathbf{q}}_m|| dt \]  
where \( || \cdot || \) is the norm of a vector.

3. Energy of the whole system

For the considerable mass and inertia tensor of the payload, the total energy involved in the motion may significantly increase. It is noteworthy that given the confliction between multiple objectives in solving MOP, energy may increase as well when \( f_1 \) or \( f_2 \) is optimized. Thus, the energy minimizing cost function is defined as
\[ f_3 = \sum_{i=1}^{n} \int_{t_0}^{t_f} [\dot{\mathbf{q}}_i(t) \tau_i(t)]^2 dt \]  

**Parameterization of joint trajectories**

To obtain smooth and continuous trajectories of joint angles, velocities, and accelerations, sinusoidal polynomial function is adopted to parameterize joint trajectories in the present study.

The trajectories are defined as
\[ \mathbf{q}_i(t) = \Delta_{1i} \sin \left( \sum_{j=0}^{7} \lambda_{ij} t^j \right) + \Delta_{2i} \]
\[ \dot{\mathbf{q}}_i(t) = \Delta_{1i} \cos \left( \sum_{j=0}^{7} \lambda_{ij} t^j \right) \left( \sum_{j=1}^{7} j \lambda_{ij} t^{j-1} \right) \]
\[ \ddot{\mathbf{q}}_i(t) = -\Delta_{1i} \sin \left( \sum_{j=0}^{7} \lambda_{ij} t^j \right) \left( \sum_{j=1}^{7} j^2 \lambda_{ij} t^{j-1} \right)^2 + \Delta_{1i} \cos \left( \sum_{j=0}^{7} \lambda_{ij} t^j \right) \left( \sum_{j=2}^{7} j(j-1) \lambda_{ij} t^{j-2} \right) \]  
\((i = n, 1, \ldots, j; j = 0, 1, \ldots, 8)\)

where \( \lambda_{ij} \) denotes the coefficient of the polynomial, \( \Delta_{1i} = (q_{i_{\text{max}}} - q_{i_{\text{min}}})/2 \) and \( \Delta_{2i} = (q_{i_{\text{max}}} + q_{i_{\text{min}}})/2 \). With sinusoidal polynomial functions, smooth and continuous trajectories of joint motions and joint angle constraints expressed in equation (26) can be achieved.

For a given self-assembly task, the values of \( \mathbf{q}_{\text{ini}}, \mathbf{x}_{\text{des}}, \) and \( \dot{x}_{\text{des}} \) are already determined, and the following conditions have to be met
\[ \dot{\mathbf{q}}_{\text{ini}} = \ddot{\mathbf{q}}_{\text{ini}} = \dot{\mathbf{q}}_{\text{des}} = 0 \]  
where \( \dot{\mathbf{q}}_{\text{des}} \) can be easily obtained by velocity-level inverse kinematics in base frame
\[ \dot{\mathbf{q}}_{\text{des}} = J_m^T (\mathbf{q}_{\text{des}}) \dot{x}_{\text{des}} \]  

In the base frame, any position-level inverse kinematics method is adopted to obtain \( \mathbf{q}_{\text{des}} \) suitable for self-assembly pose \( \mathbf{x}_{\text{des}} \); subsequently, \( \mathbf{q}_{\text{ini}} \) and \( \mathbf{q}_{\text{des}} \) in equations (32) and (33) are substituted into equation (31)
\[ \lambda_{i0} = \lambda_{i1} = \lambda_{i2} = 0 \]
\[ \lambda_{i3} = \frac{A_i}{t_f^3} - 3 \lambda_{i7} t_f^2 - \lambda_{i6} t_f^3 + \omega_i t_f^2 - \phi_i t_f \]
\[ \lambda_{i4} = 8 \lambda_{i7} t_f^6 - 3 \lambda_{i6} t_f^7 - 2 \omega_i t_f + \phi_i \]
\[ \lambda_{i5} = -6 \lambda_{i7} t_f^6 - 3 \lambda_{i6} t_f^7 + \omega_i \]  
where the symbols are calculated as follows
\[ \Delta_{i3} = \arcsin \left( \frac{q_{i_{\text{des}}} - \Delta_{2i}}{\Delta_{1i}} \right), \Delta_{i4} = \arcsin \left( \frac{q_{i_{\text{ini}}} - \Delta_{2i}}{\Delta_{1i}} \right) \]
\[ A_i = (\Delta_{i3} - \Delta_{i4}), B_i = \frac{q_{i_{\text{des}}} - \Delta_{i2}}{\Delta_{i1} \cos(\Delta_{i3})}, C_i = \frac{(q_{i_{\text{des}}} - \Delta_{i2}) B_i^2}{\Delta_{i1} \cos(\Delta_{i3})} \]
\[ \omega_i = \frac{C_i}{2 t_f^2} - \frac{3 B_i}{t_f^2} + \frac{6 A_i}{t_f^3}, \phi_i = \frac{B_i}{t_f} - \frac{3 A_i}{t_f^2} \]  

Given \( \lambda = [\lambda_{16}, \lambda_{17}, \lambda_{26}, \lambda_{27}, \ldots, \lambda_{16}, \lambda_{17}] \in \mathbb{R}^{2n \times 1} \) as the decision vector, joint trajectories are determined, and equation (26) is satisfied.
Preimpact trajectory planning of free-floating space manipulator using multiobjective particle swarm optimization

Brief review of multiobjective particle swarm optimization

Multiobjective particle swarm optimization (MOPSO) algorithm refers to an evolutionary technology based on swarm intelligence that simulates social behavior. Given the advantages of fast convergence, less computing resource occupancy, and excellent diversity, MOPSO algorithm is adopted to solve the trajectory optimization problem.

Pareto dominance is incorporated into a conventional single-objective PSO algorithm to process the problems with several cost functions in MOPSO. This algorithm employs an external repository of particles as guidance to lead the evolution of the population toward the Pareto Front.

In accordance with the multiobjective optimization theory, when the following Pareto dominance principle is satisfied, \( \lambda^0 \) dominates \( \lambda^1 \) (\( \lambda^0 \succ \lambda^1 \))
\[
\begin{align*}
    f_m(\lambda^0) &\geq f_m(\lambda^1), \quad m = 1, 2, \ldots, M \\
    f_m(\lambda^0) &< f_m(\lambda^1), \quad \exists m \in \{1, 2, \ldots, M\} 
\end{align*}
\]

The solution satisfying equation (36) is called a nondominated solution. Therefore, different from PSO, in the velocity formula of particle’s flight, the best position that all the particles ever had \( g_{best} \) is replaced by a value taken from repository \( R[h] \). Subsequently, the velocity formula is written as
\[
V[k] = \chi \cdot [V[k] + c_1 r_1 (p_{best}[k] - P[k]) + c_2 r_2 (R[h] - P[k])]
\]

where \( P[k] \) and \( V[k] \), respectively, denote position and velocity of particle \( k \); \( \chi = 2/(2 - \phi - \sqrt{\phi(\phi - 4)}) \) represents the constriction factor; \( \phi = c_1 + c_2 \), while \( c_1 \) and \( c_2 \) refer to the learning factor that regulates the step size of the particles flying to \( p_{best} \) (the best position a particle has ever explored) and \( g_{best} \); \( r_1 \) and \( r_2 \) indicate independent random numbers in the range \([0, 1]\).

Figure 2 simply depicts the steps of MOPSO algorithm.

Trajectory planning algorithm

Equation (22) indicates that the final configuration of space manipulator is capable of affecting the force impulse. Moreover, joint trajectories can vary significantly with different final states of the manipulator, which affects base disturbance and energy consumption. For a redundant space manipulator, multiple configurations should be considered in the trajectory planning process when payload reaches the contact point.29

Figure 2. Flowchart of MOPSO. MOPSO: multiobjective particle swarm optimization.

Figure 3. Flowchart of Newton–Raphson iteration.
In the present study, Newton–Raphson iterative method is employed to compute \( q_{\text{des}} \), and MOPSO is adopted for optimizing calculation. To obtain the applicable motion of space manipulator without violating the constraints, the preimpact trajectory planning algorithm is proposed as the following two stages.

Stage I: Obtain a set of desired configurations with Newton–Raphson iteration.

\( T_r \) is defined as the corresponding pose matrix of the payload in \( \Sigma_B \) for any given joint vector \( q_r \). Differential motion from \( T_r \) to desired pose \( T_{\text{des}} \) is written as

\[
T_{\text{des}} = T_r + \Delta T
\]

where \( \Delta T \) is differential motion matrix, as expressed below

\[
\Delta T = T_r^{-1} T_{\text{des}} - E_4
\]

\[
= \begin{pmatrix}
0 & -\delta z & \delta y & dx \\
\delta z & 0 & -\delta z & dy \\
-\delta y & \delta z & 0 & dz \\
0 & 0 & 0 & 0
\end{pmatrix}
\]

where \( dx, dy, \) and \( dz \) denote the differential translation operators; \( \delta x, \delta y, \) and \( \delta z \) represent the differential rotation operators. Subsequently, the differential motion vector \( D = [dx \ dy \ dz \ \delta x \ \delta y \ \delta z] \) is obtained.

According to equation (33), joint differential motion vector is calculated as

\[
dq = J_q^+ D
\]

Figure 3 shows the Newton–Raphson iteration. Execute iterative computation until the maximum iteration \( N \) is reached, corresponding desire preimpact configuration for the given joint vector can be obtained if \( ||dq|| \leq \xi \), where \( \xi \) is a given sufficiently small number.

A set of random joint vectors \( \{q_r\} \) is generated, the corresponding set of configurations can be yielded by complying with Newton–Raphson iteration. The elements that do not satisfy equation (26) are eliminated, and then, \( n_p \) types of configurations with the smallest value of \( ||q_{\text{des}} - q_{\text{ini}}|| \) are selected. Lastly, \( \{q_{\text{des}}\} \in \mathbb{R}^{1 \times n_p} \) consisting of the selected configurations is determined.

Stage II: Solve nondominated solutions for trajectory planning problem using MOPSO.

Step 1 Initialize the population:

(a) Generate a population containing \( n_p \) particles and restrict the search range as \( [\lambda_{\min}, \lambda_{\max}] \) in all directions of the \( 2n \)-dimensional decision space. Then, assign each particle a joint vector in \( \{q_{\text{des}}\} \).
A random decision vector \( \lambda \) is generated in the search range and then substituted to equation (31) to calculate \( \mathbf{q}, \dot{\mathbf{q}}, \) and \( \ddot{\mathbf{q}} \). Execute kinematics and dynamics calculation (where spatial operator algebra method is applied to reduce computational complexity) determine the values of joint torques.

If equation (27) is satisfied, \( \lambda \) is stored as the initial position of the particle; otherwise, loop to (b) until a desired decision vector is determined.

If the number of stored decision vectors reaches \( n_p \), define initial velocities of all the particles as zero.

**Step 2** Initialize external repository:

(a) The maximum allowable number of the repository is limited to \( n_R \).

---

**Table 3. Dynamics parameters.**

| Part      | \( m_i \) (kg) | \( I_i \) (kg m\(^2\))          | \( \mathbf{p}_i \) (m)  |
|-----------|-----------------|----------------------------------|-------------------------|
| Base      | \( 7.5 \times 10^4 \) | \( \begin{bmatrix} 7.5 \times 10^3 & 7.5 \times 10^3 & 7.5 \times 10^3 \end{bmatrix} \) | \( [0, 0, 0] \) |
| Link7     | 30              | \( \begin{bmatrix} 0.98 & 0.57 & 0.98 \end{bmatrix} \) | \( [0, -0.265, 0] \) |
| Link6     | 30              | \( \begin{bmatrix} 0.57 & 0.98 & 0.98 \end{bmatrix} \) | \( [-0.265, 0, 0] \) |
| Link5     | 70              | \( \begin{bmatrix} 3 & 197.2 & 197.2 \end{bmatrix} \) | \( [2.9, 0, 0] \) |
| Link4     | 75              | \( \begin{bmatrix} 1.91 & 0 & -4 \end{bmatrix} \) | \( [2.7, 0, 0.5] \) |
| Link3     | 30              | \( \begin{bmatrix} 0.98 & 0.98 & 0.57 \end{bmatrix} \) | \( [0, 0, 0.265] \) |
| Link2     | 30              | \( \begin{bmatrix} 0.98 & 0.98 & 0.57 \end{bmatrix} \) | \( [0, 0, 0.265] \) |
| Link1     | 40              | \( \begin{bmatrix} 1.7 & 1.9 & 1 \end{bmatrix} \) | \( [0, 0, 2.322] \) |
| Payload   | \( 2.5 \times 10^4 \) | \( \begin{bmatrix} 2.5 \times 10^3 & 2.5 \times 10^3 & 2.5 \times 10^3 \end{bmatrix} \) | \( [0, 0, 0.265] \) |
Table 4. Extreme conditions of final repository in objective space.

| Cost function | Minimum value | Corresponding objective vector | Maximum value | Corresponding objective vector |
|---------------|---------------|----------------------------------|---------------|--------------------------------|
| $f_1$         | 0.20          | [0.20, 1.27, 3.5611 $\times 10^4$] | $8.90 \times 10^3$ | [8.90 $\times 10^3$, 0.14, 4.25 $\times 10^4$] |
| $f_2$         | 0.02          | [0.78e+03, 0.02, 5.8677 $\times 10^4$] | 1.51          | [7.72, 1.51, 1.36 $\times 10^4$] |
| $f_3$         | 301.92        | [4.00 $\times 10^4$, 0.69, 301.93] | 1.25 $\times 10^6$ | [1.74 $\times 10^4$, 0.12, 1.25 $\times 10^4$] |

The bold values are used to highlight the positions of the maximum and minimum values.

Figure 10. Pareto Front of $f_1$ and $f_2$.

(b) Substitute kinematics and dynamics calculation results of the respective particle into equations (28) to (30) to determine the objective vector of particle $k$: $y^0_k = [f^0_1(k), f^0_2(k), f^0_3(k)]$.

(c) Store nondominated decision vector in $R_0$ according to equation (36).

(d) Initialize the best position particle $k$ has experienced as $\hat{P}_k = P_0[k]$. 

Step 3: Execute evolutionary computations:

(a) Calculate particles’ velocity using equation (37) and update the position of the respective particle. According to dynamics calculation results, only new solutions that satisfy joint torque constraints are reserved in each iteration.

(b) Add these reserved solutions into the repository via a similar solution procedure in (b) and (c) of step 2, nondominated solutions can be acquired to guide the evolutionary process of the population.

(c) The set of nondominated solutions is finally yielded, while the maximum number of iteration is reached. An appropriate solution can be taken based on the meanings of the particle’s position in objective space, according to the task priority or practical engineering requirements.

Simulation results

Seven degrees of freedom space manipulator

As shown in Figure 4, a 7DOF redundant space manipulator is studied in the present study. D-H coordinate system is established, and the relevant parameters are listed in Table 2.

Figure 11. Joint trajectories for solutions A and B: (a) Solution A and (b) solution B.

Table 3 lists the dynamics parameters, in which only the elements of diagonal matrixes are given: $d_4 = 0.52$ m, $d_7 = d_1 = 1.2$ m, $d_6 = d_5 = d_3 = d_2 = 0.53$ m, and $a_5 = a_4 = 5.8$ m. Pose of $\Sigma_8$ is $[-0.2$ m, 0 m, 2 m, 0°, 0°, 0°] in $\Sigma_B$. Pose of the assembly mechanism (mounted on the payload) is [0 m, 1.5 m, −1 m, 0°, 0°, 0°] in $\Sigma_1$.

Results and discussions

The pose of the base at initial time is zero in inertia frame; the initial joint angles are set to $q_{ini} = [70°, −100°, −100°, −40°, 20°, −90°, 20°]$; the desired pose of assembly mechanism when payload reaches the contact point is $x_{des} = [0$ m, $−6.5$ m, $−2$ m, 0°, 0°, $−180°]$ in base frame; the desired self-assembly velocity in base frame is $\dot{x}_{des} = [0, 0.1$ m/s, 0, 0, 0, 0]; and the total time for
preimpact motion of space manipulator is $t_f = 200s$. The upper and lower limits of joint $i$ ($i = 7, 6, \ldots, 1$) are $q_{i_{\min}} = -270^\circ$ and $q_{i_{\max}} = 270^\circ$, and the limits of joint torques reach $\tau_{i_{\min}} = -500N \cdot m$ and $\tau_{i_{\max}} = 500N \cdot m$.

At stage I of the simulation, the relevant parameters in Newton–Raphson iteration are set to $\xi = 0.0001$ and $N = 10,000$. We generate 5500 uniformly distributed random configurations and corresponding desired joint vectors are solved using iteration equation. $q_{\text{des}} \in R^{7 \times 200}$ is selected according to the value of $|q_{\text{des}} - q_{\text{ini}}|$. For stage II, the specific parameters of MOPSO are set: the range of values for 14 decision variables refers to $[l_{\min}, l_{\max}] = [-10 \times 10^{-16}, +10 \times 10^{-16}]$; $n_p = 200$; $n_R = 1000$; and $c_1 = c_2 = 2.05$. The yielded 200 joint vectors in stage I are assigned for the particles. After 200 iterations of evolutionary computation, Figure 5 shows the distribution of all nondominated solutions in objective space. It is seen that all the solutions distribute on a surface, which can be approximately regarded as the Pareto Front. Execute dynamics calculation for each nondominated solution, and the maximum absolute value of joint torques does not exceed 499.6272 N m, revealing that all the solutions satisfy equation (27).

The optimal values of three cost functions in each generation are shown in Figures 6 to 8, demonstrating high convergence effects. The number of particles in external repository is elevated, as shown in Figure 9. $n_R$ is reached at 115th iteration and the subsequent fluctuation of the curve indicates the maintenance of repository. The extreme values of cost functions are listed in Table 4.

### Table 5. Relevant vectors of solutions A and B.

| Solution | Decision vector/$\times 10^{-16}$ | Desired joint angle vector ($^\circ$) | Objective vector |
|----------|-----------------------------------|-------------------------------------|------------------|
| A        | $[2.02, 1.09, -2.66, -6.82, -0.40, 5.93, 0.65, \ldots]$ | $[107.92, -70.64, -151.10, -94.07, 65.18, \ldots]$ | $[14.30, 0.18, 7.60 \times 10^7]$ |
| B        | $[0.85, 2.26, 4.50, -6.58, 0.72, 5.08, 0.98, -7.62, \ldots]$ | $[100.89, -71.24, -143.99, -94.07, 58.06, \ldots]$ | $[163.19, 0.18, 9.24 \times 10^7]$ |

![Figure 12. Payload velocity changes in base frame: (a) Linear velocity (solution A), (b) linear velocity (solution B), (c) angular velocity (solution A), and (d) angular velocity (solution B).](image-url)
angles listed in Table 5 are reached for each solution, demonstrating that the self-assembly pose of payload is satisfied at the impact time. Figure 12 shows the velocity of the payload in base frame for both solutions, and the desired assembly velocity \( \dot{\mathbf{x}}_{\text{des}} = \{0, 0.1 \text{ m/s, 0, 0, 0, 0}\} \) is satisfied. All the requirements are met.

Figure 13 shows the variations of base attitude (represented by Z-Y-X Euler angles). Given the kinematics calculation, the final pose of the base in inertia frame is \( \mathbf{x}^b_b(t_f) = [2.21 \text{ m, } -0.27 \text{ m, 0.95 m, } -9.78^\circ, -1.26^\circ, 3.25^\circ] \) and \( \mathbf{x}^b_B(t_f) = [2.19 \text{ m, } -0.24 \text{ m, 0.85 m, } -9.29^\circ, -4.21^\circ, 0.01^\circ] \), and the disturbance of the base is similar. Although \( f_2^A = 101.68\% f_2^B \), values of the other two cost functions for solution A are much smaller than that of solution B (\( f_1^A = 8.76\% f_1^B \) and \( f_3^A = 82.25\% f_3^B \)). By contrast, \( \lambda_A \) is considered a proper solution for the given trajectory planning problem. As shown in Figures 11 and 13, for the appropriate planning of joint trajectories, base attitude changes are smooth and continuous. Since optimization of base attitude disturbance is considered in the trajectory planning, final attitude values of the base are acceptable. Conclusively, the obtained nondominated solution is applicable.

**Conclusion**

Given the significance of on-orbit assembling a heavy payload to free-floating base with a space manipulator, the present study proposes a scheme of multiobjective preimpact trajectory planning.

- The force impulse in the self-assembly process is formulated as a function of joint angles, joint velocities, base pose, and collision direction. As revealed from such function, force impulse is determined by the current state of the system at the contact point as well as the history motion of the space manipulator.
- The joint trajectories are built by coefficients of sinusoidal polynomial functions. Inverse kinematics is adopted to transmit constraints, so pose and velocity requirements for self-assembly are satisfied. Based on redundancy of space manipulator, multiple configurations are obtained as the input conditions for trajectory planning.
- The trajectory planning algorithm based on MOPSO minimizes force impulse, base disturbance, and energy cost simultaneously. Only solutions that satisfy joint torque constraints are reserved for both initialization and evolution. Each of the yielded nondominated solutions is of explicit physical significance in objective space, which provides convenience to pick out an appropriate solution.

The proposed trajectory planning method is capable of generating multiple, feasible, and effective preimpact motion schemes of space manipulator for selection.
Accordingly, the on-orbit self-assembly of heavy payload and the spacecraft base is endowed with promising engineering application prospect. However, only preimpact process is considered in this study. Disturbance of the base and impact of the joint may cause discontinuous and instable motion of space manipulator system after assembly collision, which is to be studied in our future works.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Nature Science Foundation of China [No. 51605129], National Nature Science Foundation of China [No. 51704092], and Hefei University of Technology Xuancheng Campus Doctoral Production-Study Research Fund [No. XC2016JZBZ05].

ORCID iD
Zhe Wu @ https://orcid.org/0000-0001-5190-0162

References
1. Flores-Abad A, Ma O, Pham K, et al. A review of space robotics technologies for on-orbit servicing. Prog Aerosp Sci 2014; 68: 1–26.
2. Coleshill E, Oshinowo L, Rembala R, et al. Dextre: improving maintenance operations on the international space station. Acta Astronaut 2009; 64: 869–874.
3. Xu WF, Peng JQ, Liang B, et al. Hybrid modeling and analysis method for dynamic coupling of space robots. IEEE Trans Aero Elec Sys 2016; 52: 85–98.
4. Korayem MH, Nazemizadeh M, and RahimiHN. Dynamic optimal payload path planning of mobile manipulators among moving obstacles. Adv Robot 2014; 28: 1389–1402.
5. Korayem MH and Nikoobin A. Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem. Int J Adv Manuf Tech 2009; 44: 725–736.
6. Saramago SFP and Ceccarelli M. An optimum robot path planning with payload constraints. Robotica 2002; 20: 395–404.
7. Liu Y, Jia QX, Chen G, et al. Multi-objective trajectory planning of FFSM carrying a heavy payload. Int J Adv Robot Syst 2015; 12: 1–13. DOI: 10.5772/61235.
8. Dalla VK and Pathak PM. Docking operation by multiple space robots for minimum attitude disturbance. Int J Model Simul 2017; 38: 38–49.
9. Yoshida K, Sashida N, and Kurazume R. Modeling of collision dynamics for space free-floating links with extended generalized inertia tensor. In: IEEE international conference on robotics & automation, Nice, France, France, 12–14 May 1992, pp. 899–904. IEEE.
10. Chen G, Jia QX, Sun HX, et al. Modeling and simulation on impact motion of space robot for object capturing. In: 5th IEEE conference on industrial electronics and applications, Taichung, Taiwan, 15–17 June 2010, pp. 130–135. Taichung: IEEE.
11. Huang PF, Chen K, and Xu YS. Optimal path planning for minimizing disturbance of space robot. In: 2006 9th international conference on control, automation, robotics and vision, Singapore, 5–8 December 2006, p. 1465. Singapore: IEEE.
12. Huang PF, Yuan JP, Xu YS, et al. Approach trajectory planning of space robot for impact minimization. In: 2006 IEEE international conference on information acquisition, vols 1 and 2, conference proceedings, Weihai, China, 20–23 August 2006, pp. 382–387. Weihai: IEEE.
13. Wee LB and Walker MW. On the dynamics of contact between space robots and configuration control for impact minimization. IEEE Trans Robot Autom 1993; 9: 581–591.
14. Zhang L, Jia QX, Chen G, et al. Pre-impact trajectory planning for minimizing base attitude disturbance in space manipulator systems for a capture task. Chinese J Aeronaut 2015; 28: 1199–1208.
15. Cong PC and Zhang X. Preimpact configuration analysis of a dual-arm space manipulator with a prismatic joint for capturing an object. Robotics 2013; 31: 853–860.
16. Zhang Q, Ji L, Zhou DS, et al. Nonholonomic motion planning for minimizing base disturbances of space manipulators based on multi-swarm PSO. Robotica 2017; 35: 861–875.
17. Luo JJ, Yu M, Wang MM, et al. A fast trajectory planning framework with task-priority for space robot. Acta Astronaut 2018; 152: 823–835.
18. Misra G and Bai XL. Task-constrained trajectory planning of free-floating space-robotic systems using convex optimization. J Guid Control Dyn 2017; 40: 2857–2870.
19. Zhou C, Jin MH, Liu YC, et al. Motion planning for redundant free-floating space robot with local optimization of reaction torque and joint torque simultaneously. In: 2016 IEEE international conference on robotics and biomimetics (Robio), Qingdao, China, 3–7 December 2016, pp. 344–349. Qingdao: IEEE.
20. Jia Q, Zhang L, and Chen G. Pre-impact trajectory optimization of redundant space manipulator with multi-target fusion. J Aeronaut 2014; 35: 639–647.
21. Wang MM, Luo JJ, and Walter U. Novel synthesis method for minimizing attitude disturbance of the free-floating space robots. J Guid Control Dyn 2016; 39: 693–702.
22. Zhou C, Jin MH, Liu YC, et al. Hybrid task priority-based motion control of a redundant free-floating space robot. Chinese J Aeronaut 2017; 30: 2024–2033.
23. Wang MM, Luo JJ, Fang J, et al. Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm. Adv Space Res 2018; 61: 1525–1536.
24. Zeng C, Wei XP, and Zhang Q. Subsection evolution in GA for trajectory planning of a space manipulator. *Int J Adv Robot Syst* 2013; 10. DOI: Art. 23810.5772/56401.

25. Wang MM, Luo JJ, and Walter U. Trajectory planning of free-floating space robot using particle swarm optimization (PSO). *Acta Astronaut* 2015; 112: 77–88.

26. Ahmed F and Deb K. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Comput* 2013; 17: 1283–1299.

27. Saravanan R, Ramabalan S, and Balamurugan C. Evolutionary optimal trajectory planning for industrial robot with payload constraints. *Int J Adv Manuf Tech* 2008; 38: 1213–1226.

28. Coello CAC, Pulido GT, and Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans Evolut Comput* 2004; 8: 256–279.

29. Huang PF, Xu WF, Liang B, et al. Configuration control of space robots for impact minimization. In: *2006 IEEE international conference on robotics and biomimetics, Vols I-3*, Kunming, China, 17–20 December 2006, p. 357. IEEE.

30. Wang X, Yang GL, and Zhang FS. An inverse kinematics algorithm of the general 6-DOF robot based on Newton-Raphson iteration. *Transducer Microsyst Technol* 2010; 29: 116–118.