Mech. Sci., 13, 41–53, 2022
https://doi.org/10.5194/ms-13-41-2022

Mechanical Sciences
Open Access

Evolutionary multi-objective trajectory optimization for a redundant robot in Cartesian space considering obstacle avoidance

Yong Liu[1], Xiang Li[1], Peiyang Jiang[1], Zhe Du[2], Zhe Wu[1], Boxi Sun[1], and Xinyan Huang[3]

[1]School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China
[2]Department of Mechanical Engineering, Kyushu University, Fukuoka 819-0395, Japan
[3]School of Mechanical Engineering, Zhejiang University, Hangzhou 310007, China

Correspondence: Yong Liu (xcliuyong@hfut.edu.cn)

Received: 13 October 2021 – Revised: 20 December 2021 – Accepted: 21 December 2021 – Published: 8 February 2022

Abstract. A method of end-effector trajectory planning in Cartesian space based on multi-objective optimization is proposed in this paper to solve the collision problem during the motion of the redundant manipulator. First, a cosine polynomial function is used to interpolate the trajectory of the end effector, enabling it to reach the desired pose at a specific time. Then, the joint trajectory of the manipulator is solved by inverse kinematics, and the null space term is introduced as the joint limit constraint in the inverse kinematics equation. During the operation of the manipulator, the collision detection algorithm is employed to calculate the distance between the obstacle and each arm in real time. Finally, a multi-objective, multi-optimization model of trajectory that considers the obstacle avoidance, joint velocity, joint jerk and energy consumption is established and optimized with a multi-objective particle swarm optimization algorithm. The simulation results demonstrate that the proposed method can effectively accomplish the trajectory planning task and avoid obstacles; the joint trajectories obtained are smooth and meet the limit constraints; the joint jerk and energy consumption are well suppressed.

1 Introduction

In many of today's vastly complex and difficult manufacturing applications, manipulators may have more degrees of freedom (DOFs) than essential due to the need of executing intended complicated jobs like human arms. Specifically, the end effector of redundant manipulator should satisfy a desired trajectory while its body avoids colliding with obstacles in the environment, contributing to finishing the job. The extremely crucial obstacle avoidance problem is that robotic manipulators are required to move from an initial pose to a specified final objective without colliding with any obstacles in the workspace.

To solve the trajectory planning problem considering obstacle avoidance, many researchers have focused on different methods. For example, Ma and Liang (2020) proposed an obstacle avoidance algorithm for a 6-DOF manipulator based on a sixth-degree polynomial, with limitations to the redundant manipulator. For the obstacle-avoiding motion of the end effector of the redundant robot, Shen et al. (2014) presented an obstacle avoidance algorithm based on the transition between the primary and the secondary tasks to acquire a smooth and continuous trajectory. Dalla and Pathak (2017) put forward a methodology that a limited number of joints are actuated and a curve-constrained link trajectory tracking is developed to control hyper-redundant space robots. However, these are only for planar robots. Shi et al. (2019) constructed a visual servoing system, which obtained the image errors by the ORB feature points and constructed a virtual repulsion torque to avoid obstacles.

In addition to the above methods, more researchers handle the obstacle avoidance problem by introducing the concept of potential field to construct the artificial potential field. Safeea et al. (2020) redefined the obstacle avoidance problem as a second-order optimization problem and constructed the potential function that is minimized by Newton's method to realize real-time trajectory calculation in joint space. It lacks the intuitiveness and efficiency compared with Cartesian space trajectory planning. Wang et al. (2018) realized trajectory planning and obstacle avoidance through the improved artificial potential field method with the consideration of both position and posture of manipulator end effectors. Park et al. (2020) employed a Jacobian matrix-based numerical method and an improved potential field method to achieve path planning and obstacle avoidance, both in redundant and non-redundant conditions.

The above-mentioned trajectory planning methods focus more on obstacle avoidance than kinematics or dynamics optimization. Nowadays, an optimized path should be found for robot motion regarding minimum time (Bourbonnais et al., 2015), minimum distance (Salaris et al., 2010), and minimum energy (Arakelian et al., 2011) given the robot optimization problem in obstacle avoidance planning. This suggests that multiple objective functions should be simultaneously optimized to obtain a suitable trajectory. Based on the above articles, some criteria for evaluating manipulator trajectory can be obtained:

1. Trajectories need to be predictable, accurate, and effective to both compute and execute.

2. Trajectories need to ensure that the joints of the robotic arm do not appear singular.

3. Trajectories should be found for a manipulator with respect to minimum time, minimum distance, and minimum energy.

Many researchers used an evolutionary algorithm to address multi-objective planning problems considering obstacle avoidance. For example, Saravanan et al. (2008) optimized two evolutionary algorithms of an elite non-dominant sorting genetic algorithm (NSGA-II) and differential evolution (DE) to handle the multi-objective problem and achieved the predetermined goal in the PUMA560 robot. Han et al. (2013) realized the obstacle avoidance path planning optimization of the robot arm in the environment with single or multiple obstacles based on genetic algorithms with execution time, space distance, and trajectory length as the optimization parameters. However, most of the above researchers have weighted each constraint as an objective function, making the result very dependent on the setting of the weight factor. Some researchers transform multi-objective problems into quadratic programming (QP) problems. Chen and Zhang (2017) proposed a hybrid multi-objective scheme including the specified primary task for the end effector, obstacle avoidance, joint-physical limits avoidance, and repetitive motion of redundant robot manipulators and transformed it into a dynamic quadratic program (DQP) problem. Zhang et al. (2020) studied two schemes of hybrid end-effector pose-maintaining and obstacle-limits avoidance (PM-OLA) for redundant robot motion planning based on QP framework to solve the end-effector drift problem in the planar task. Other researchers obtained the Pareto front in multi-objective problems, as well as the non-dominant solution. Guigue et al. (2010) employed the discrete dynamic programming (DDP) approximation method to obtain an approximate solution of the Pareto optimal set to handle multi-objective robotic trajectory planning problems. Marcos et al. (2012) combined a closed-loop pseudo-inverse method with a multi-objective genetic algorithm to control the joint position and choose a solution from the Pareto solution set. Although these researchers solved the multi-objective problem, they did not add energy consumption constraint to the multi-objective problem for obtaining a superior solution with more energy-saving or less joint jerk.

Besides, many researchers have improved the particle swarm optimization (PSO) algorithm to acquire the optimized trajectory of the robotic arm. Cao et al. (2016) adopted a hybrid algorithm combing crossover and mutation particle swarm optimization (CMPSO) with the interior point method (IPM) to determine the minimum-length motion path of robot end effector and the best movement posture of the robot arm. Lin (2014) proposed a k-means clustering method based on PSO to address the near-optimal solution of the minimum-jerk joint trajectory, making it easy to obtain the optimized trajectory control law. Kucuk (2017) developed an optimal trajectory generation algorithm (OTGA) for generating smooth motion trajectories with minimum time with the PSO algorithm. Under the diversification of robot optimization goals, some researchers used a multi-objective particle swarm optimization (MOPSO) algorithm (Coello Coello and Lechuga, 2002) to solve multi-objective optimization problems. For instance, Yan et al. (2020) presented multi-objective configuration optimization for the maximum maneuverability and minimum base disturbance in the pre-contact stage of dual-arm space robot by MOPSO. Liu and Zhang (2021) proposed an improved external archives self-searching multi-objective particle swarm optimization (EASS-MOPSO) algorithm to handle the multi-objective trajectory optimization problem.

In this paper, regarding obstacle avoidance, the change rate of joint torque, joint angular velocity, and energy consumption are minimized using the path planning method based on a multi-objective optimization algorithm. The main contributions of this paper are highlighted as follows.

1. A cosine polynomial function is used to interpolate the trajectory of the end effector, enabling it to reach the desired pose at a specific time, and the null space term is introduced as the joint limit constraint in the inverse kinematics equation.

2. During the operation of the manipulator, a collision detection algorithm is employed to calculate the distance between the obstacle and each arm in real time.

3. The trajectory optimization model of redundant manipulator was established to consider obstacle avoidance, joint velocity, joint jerk and energy consumption, and

the optimized trajectories were obtained by a multi-objective particle swarm optimization algorithm.

The remaining paper is organized as follows. In Sect. 2, the end trajectory of the robotic arm is interpolated using the cosine fifth-degree polynomial, allowing the end to reach the specified pose at the specified time. Then, the joint trajectory of the robotic arm is reversed through inverse kinematics. Section 3 presents the collision detection algorithm used in this paper. Afterward, the multi-objective optimization problem equations are described in Sect. 4. The obstacle avoidance trajectory planning based on multi-objective optimization is introduced in Sect. 5. Simulation results are analyzed in Sect. 6. Finally, conclusions are drawn in Sect. 7.

2 Trajectory planning of redundant manipulator

2.1 Trajectory parameterization of end effector in Cartesian space

A five-order cosine polynomial is adopted to describe the trajectory to ensure the position, orientation, velocity, and acceleration of the end effector change smoothly and continuously. With this method, the motion of the end effector follows a helical curve in Cartesian space (Zghal et al., 1990). The pose of the end effector is taken to compose a six-dimensional generalized coordinate vector $\boldsymbol{X}_\mathrm{e} = [X_1(t), \ldots, X_6(t)]^\mathrm{T}$ as the status variable of the system. The trajectory is parameterized as

$$\begin{aligned} X_j(t) = {} & \left(\lambda_{j5}t^5 + \lambda_{j4}t^4 + \lambda_{j3}t^3 + \lambda_{j2}t^2 + \lambda_{j1}t + \lambda_{j0}\right) \\ & \cdot \cos\left(\frac{2\pi t}{T_j} + \varphi_j\right) + X_{j0}, \end{aligned} \tag{1}$$

where X_{j0} denotes the initial pose of end effector, and λ, T and φ are trajectory parameters, $j = 1, \ldots, 6$.

By taking the differential of the equation above, the generalized velocity and acceleration of the end effector can be expressed as

$$\begin{aligned} \dot{X}_j(t) &= K_{j1}(t)\cos\left(\tfrac{2\pi t}{T_j} + \varphi_j\right) \\ &\quad - \tfrac{2\pi}{T_j}K_{j2}(t)\sin\left(\tfrac{2\pi t}{T_j} + \varphi_j\right) \\ \ddot{X}_j(t) &= K_{j3}(t)\cos\left(\tfrac{2\pi t}{T_j} + \varphi_j\right) \\ &\quad - \tfrac{4\pi}{T_j}K_{j1}(t)\sin\left(\tfrac{2\pi t}{T_j} + \varphi_j\right) \\ &\quad - \tfrac{4\pi^2}{T_j^2}K_{j2}(t)\cos\left(\tfrac{2\pi t}{T_j} + \varphi_j\right), \end{aligned} \tag{2}$$

where $K_j(t)$ can be described as follows:

$$\begin{aligned} K_{j1}(t) &= 5\lambda_{j5}t^4 + 4\lambda_{j4}t^3 + 3\lambda_{j3}t^2 + 2\lambda_{j2}t + \lambda_{j1} \\ K_{j2}(t) &= \lambda_{j5}t^5 + \lambda_{j4}t^4 + \lambda_{j3}t^3 + \lambda_{j2}t^2 + \lambda_{j1}t + \lambda_{j0} \\ K_{j3}(t) &= 20\lambda_{j5}t^3 + 12\lambda_{j4}t^2 + 6\lambda_{j3}t + 2\lambda_{j2}. \end{aligned} \tag{3}$$

The boundary conditions of the trajectory planning problem are

$$\begin{aligned} X_j(t_0) &= X_{j0},\ \dot{X}_j(t_0) = \ddot{X}_j(t_0) = 0 \\ X_j(t_\mathrm{f}) &= X_{j\mathrm{f}},\ \dot{X}_j(t_\mathrm{f}) = 0,\ \ddot{X}_j(t_\mathrm{f}) = 0, \end{aligned} \tag{4}$$

where t_0 and t_f represent the initial and final time, respectively; $X_{j\mathrm{f}}$ refers to the desired pose of end effector.

By substituting Eq. (4) into Eqs. (1) and (2), the following relationship can be obtained:

$$\begin{aligned} \lambda_{j0} &= \lambda_{j1} = \lambda_{j2} = 0 \\ \lambda_{j3} &= \frac{\begin{array}{c} 2\big(2\pi^2 D_j t_\mathrm{f}^2 \sin^2(\psi) + \pi^2 D_j t_\mathrm{f}^2 \cos^2(\psi) \\ -4\pi D_j T_j t_\mathrm{f} \sin(\psi)\cos(\psi) + 5 D_j T_j^2 \cos^2(\psi)\big) \end{array}}{T_j^2 t_\mathrm{f}^3 \cos^3(\psi)} \\ \lambda_{j4} &= \frac{\begin{array}{c} -\big[D_j\big(8\pi^2 t_\mathrm{f}^2 \sin^2(\psi) + 4\pi^2 t_\mathrm{f}^2 \cos^2(\psi) \\ -14\pi T_j t_\mathrm{f} \sin(\psi)\cos(\psi) + 15 T_j^2 \cos^2(\psi)\big)\big] \end{array}}{T_j^2 t_\mathrm{f}^4 \cos^3(\psi)} \\ \lambda_{j5} &= \frac{\begin{array}{c} 2D_j\big(2\pi^2 t_\mathrm{f}^2 \sin^2(\psi) + \pi^2 t_\mathrm{f}^2 \cos^2(\psi) \\ -3\pi T_j t_\mathrm{f} \sin(\psi)\cos(\psi) + 3 T_j^2 \cos^2(\psi)\big) \end{array}}{T_j^2 t_\mathrm{f}^5 \cos^3(\psi)}, \end{aligned} \tag{5}$$

where ψ and D_j can be described as

$$\begin{aligned} \psi &= \tfrac{2\pi t_\mathrm{f}}{T_j} + \varphi_j \\ D_j &= X_{j\mathrm{f}} - X_{j0}. \end{aligned} \tag{6}$$

According to Eq. (5), polynomial coefficients can be determined with parameters T_j and φ_j.

Then trajectories of end effector that satisfy the constraints can be obtained with the following parameters vector:

$$\boldsymbol{\xi} = [T_1, \varphi_1, T_2, \varphi_2, \ldots, T_6, \varphi_6]^\mathrm{T}. \tag{7}$$

2.2 Joint motion planning for redundant manipulator

The subject of this paper is a n-DOF redundant manipulator. Assuming the dimension of task space is $m (m \leq 6)$, a redundant manipulator satisfies $n > m$. The joint velocity vector of redundant manipulator is

$$\dot{\boldsymbol{q}} = \mathbf{J}^+ \dot{\boldsymbol{x}}_e + \left(\mathbf{I} - \mathbf{J}^+\mathbf{J}\right)\dot{\boldsymbol{\Phi}}, \tag{8}$$

where $\boldsymbol{x}_\mathrm{e} \in \mathbb{R}^m$ indicates the pose of end effector in Cartesian space; $\boldsymbol{q} \in \mathbb{R}^n$ denotes joint angle vector; $\mathbf{J} \in \mathbb{R}^{n\times m}$ represents the Jacobian matrix of manipulator; $\mathbf{I} \in \mathbb{R}^{n\times n}$ refer to unit matrix; $\dot{\boldsymbol{\Phi}} \in \mathbb{R}^n$ stands for an arbitrary n-dimensional vector; $(\mathbf{I} - \mathbf{J}^+\mathbf{J}) \in \mathbb{R}^{n\times n}$ designates the null space projection matrix; $\mathbf{J}^+ \in \mathbb{R}^{n\times m}$ means the generalized inverse of the Jacobian matrix, which is defined by

$$\mathbf{J}^+ = \mathbf{J}^\mathrm{T}\left(\mathbf{J}\mathbf{J}^\mathrm{T}\right)^{-1}. \tag{9}$$

For redundant manipulators, a specific trajectory of end effector can be obtained by multiple joint trajectories. Hence, the joint trajectory should be selected to ensure that the motion of the manipulator follows specific constraints. Null space (Zghal et al., 1990) term is introduced in this paper to enable the joint motion to satisfy joint limit constraints. The joint limit constraint function is established as

$$H(q) = \sum_{i=1}^{n} \frac{(q_{i\max} - q_{i\min})}{(q_{i\max} - q_i)(q_i - q_{i\min})}, \tag{10}$$

where q_i donates the angle of joint i; $q_{i\max}$ and $q_{i\min}$ represent upper and lower limits of joint i, respectively.

According to the gradient projection method, the joint angle limit constraints are resolved by substituting $k\nabla \boldsymbol{H}(\boldsymbol{q})$ for $\dot{\boldsymbol{\Phi}}$ in Eq. (8) as

$$\dot{\boldsymbol{q}} = \mathbf{J}^{+}\dot{\boldsymbol{x}}_{\mathrm{e}} + k(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\nabla \boldsymbol{H}(\boldsymbol{q}), \tag{11}$$

where the gradient vector $\nabla \boldsymbol{H}(\boldsymbol{q})$ is given by

$$\nabla \boldsymbol{H}(\boldsymbol{q}) = \left(\frac{\partial \boldsymbol{H}(\boldsymbol{q})}{\partial q_1}, \frac{\partial \boldsymbol{H}(\boldsymbol{q})}{\partial q_2}, \ldots, \frac{\partial \boldsymbol{H}(\boldsymbol{q})}{\partial q_n}\right). \tag{12}$$

$\nabla \boldsymbol{H}(\boldsymbol{q})$ reaches minimum value when $q_i = (q_{i\min} + q_{i\max})/2$; $\nabla \boldsymbol{H}(\boldsymbol{q})$ approaches to infinity when $q_i = q_{i\min}$ or $q_i = q_{i\max}$.

3 Collision detection algorithm

A collision detection algorithm should be established to achieve obstacle avoidance. A redundant manipulator is explored in this paper. Most of the links are like cuboids or cylinders. Moreover, obstacles are irregular geometric objects. The collision detection between the manipulator and obstacles is extremely complicated. Thus, sphere and cylindrical envelopes are applied in this paper to describe the manipulator links, joints, and obstacles.

3.1 The description of space geometry relation

Considering that obstacles in three-dimensional space generally have irregular geometric shapes and various kinds, it is difficult to extract their boundary contours. Thus, regular body envelopment is adopted in this paper to simplify the obstacle model, so as to simplify the complexity of obstacle modeling. For example, the cylindrical envelopment model and sphere envelopment model are employed for slender rod and rounded objects, respectively. This modeling method expands the range of obstacles while guaranteeing the reliability of the collision detection algorithm.

The obstacle uses a sphere envelope, as illustrated in Fig. 1. The radius of the sphere is r_{S}. The cylindrical envelope is used for the links of the manipulator while the sphere envelope and cylindrical envelope are adopted to describe the joint. In this paper, a 7-DOF redundant manipulator (Fig. 5) is studied. As exhibited in Fig. 2. The two typical cases are described as follows.

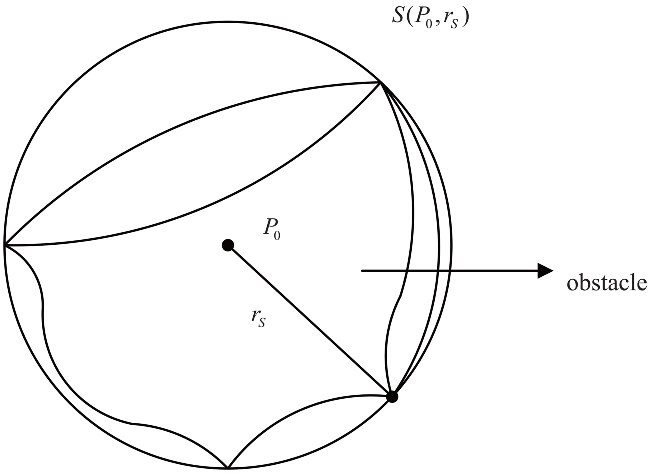


Figure 1. The obstacle uses sphere envelope.

1. For case I, when axes of adjacent links are vertical, straight connecting links (link $i-1$ and link i) are enveloped by cylinders whose axes are coplanar. The intersection points P_i of cylindrical axes are centers of joint envelope spheres. The radius of the cylinder for the link i is r_i; the radius of the sphere for the joint i is $r_i^j \geq \sqrt{r_i^2 + r_{i-1}^2}$. Hence, the distance between links and obstacles can be substituted as the distance from a line segment to a point in 3-dimensional space.

2. For case II, when axes of adjacent links are parallel, joint i and link $i-1$ are enveloped in the same way as case I considering the joint with offset. Hence, $r_i^j \geq \sqrt{r_{\mathrm{joint}}^2 + r_{i-1}^2}$, where r_{joint} indicates the radius of envelope cylinder for joint i. Two cylinders whose axes are coplanar and a sphere whose center is located in the intersection point (P_i) of the axes are applied to envelop it.

3.2 Collision detection algorithm

The space geometry (Guan et al., 2015) relations between manipulator's links and obstacle are divided into three classes, as exhibited in Fig. 3. Whether the links and obstacle collide depends on d_i, it denotes the minimum distance between the obstacle and manipulator's link/joint. If $d_i > 0$, the obstacle is avoided; otherwise, it collides with the manipulator. Therefore, the collision detection problem between cylinder and sphere can be reduced to the calculation of the shortest distance between a point and line segment in 3D space.

Assume the length of link i is L_i. Centers of the bottom are P_i and P_{i+1}. A vertical line of segment P_iP_{i+1} is drawn through P_0, and P_{vi} indicates the point of intersection to the cylindrical axis.

Based on the space geometry relation, the collision detection algorithm (Cheng et al., 1993) to achieve obstacle

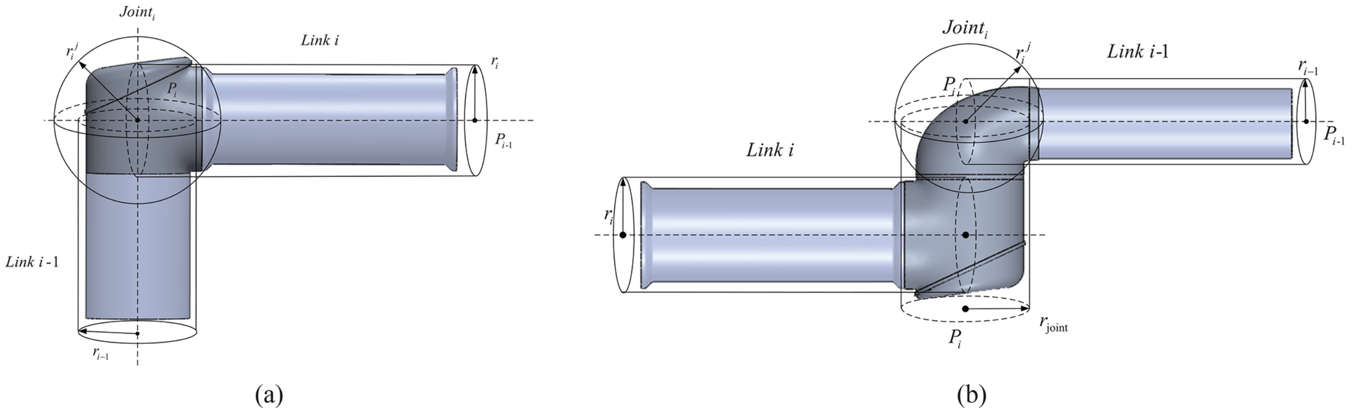


Figure 2. Links of the manipulator use cylindrical envelope and sphere envelope. **(a)** Case I. **(b)** Case II.

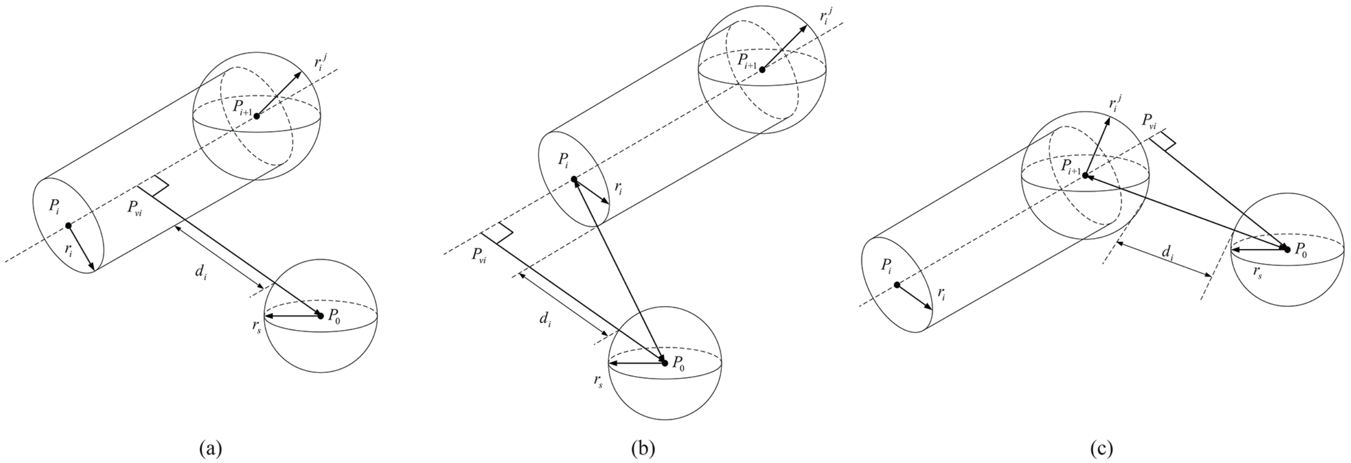


Figure 3. The space geometry relations between manipulator's links and obstacle. **(a)** Class I. **(b)** Class II. **(c)** Class III.

avoidance path planning for redundant manipulators is summarized as follows.

Step 1. Based on forwarding kinematics, the coordinates P_i of all nodes and joints are calculated with the current configuration and geometrical sizes of the manipulator.

The relative translation and rotation between link i coordinate and link $i-1$ coordinate can be described with a homogeneous transformation matrix T_{i-1}^{i}. The kinematics equation of n degree of freedom manipulators can be expressed as

$$T_i^0 = \prod\nolimits_{j=1}^{i} T_j^{j-1}. \tag{13}$$

Step 2. $\boldsymbol{P}_i\boldsymbol{P}_{vi}$ is defined as the projection vector of $\boldsymbol{P}_i\boldsymbol{P}_0$ on $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$, and the scale factor λ is calculated:

$$\lambda = \frac{\boldsymbol{P}_i\boldsymbol{P}_{vi}}{\boldsymbol{P}_i\boldsymbol{P}_{i+1}} = \frac{\boldsymbol{P}_i\boldsymbol{P}_0 \cdot \boldsymbol{P}_i\boldsymbol{P}_{i+1}}{\boldsymbol{P}_i\boldsymbol{P}_{i+1} \cdot \boldsymbol{P}_i\boldsymbol{P}_{i+1}}. \tag{14}$$

The three-dimensional coordinates of the perpendicular foot P_{vi} can be obtained with $P_{vi} = P_i + \lambda(P_{i+1} - P_i)$.

Step 3. The minimal distance from the links/joints to the obstacles is calculated. There are three possible positions of perpendicular foot P_{vi} related to $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$:

- For class I, P_{vi} is on $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$ when $\leq \lambda \leq 1$.
- For class II, P_{vi} is on the reverse extension of $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$ when $\lambda < 0$.
- For class III, P_{vi} is on the extension of $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$ when $\lambda > 1$.

Therefore, the minimum distance between P_0 and $\boldsymbol{P}_i\boldsymbol{P}_{i+1}$ is

$$d_i = \begin{cases} \|\boldsymbol{P}_0\boldsymbol{P}_{vi}\| - r_i - r_{\text{S}}, \lambda < 0 \\ \min\Big\{ \|\boldsymbol{P}_0\boldsymbol{P}_{i+1}\| - r_i^j \\ -r_{\text{S}}, \|\boldsymbol{P}_0\boldsymbol{P}_{vi}\| - r_i - r_{\text{S}} \Big\}, 0 \leq \lambda \leq 1 \\ \|\boldsymbol{P}_0\boldsymbol{P}_{i+1}\| - r_i^j - r_{\text{S}}, \lambda > 1 \end{cases}. \tag{15}$$

Step 4. If $\min\{d_1, d_2, \ldots, d_n\} < 0$, a collision occurs. Then, the collision information is outputted. According

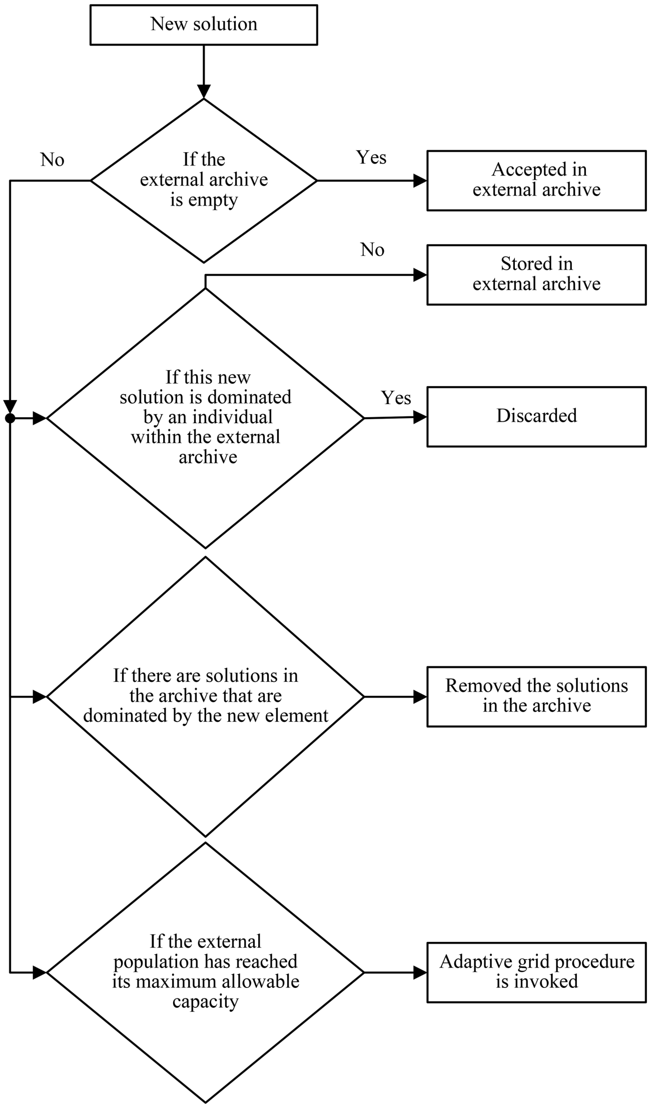

New solution
No
If the external archive is empty
Yes
Accepted in external archive
No
Stored in external archive
If this new solution is dominated by an individual within the external archive
Yes
Discarded
If there are solutions in the archive that are dominated by the new element
Removed the solutions in the archive
If the external population has reached its maximum allowable capacity
Adaptive grid procedure is invoked

Figure 4. The maintenance process of external repository.

to the collision condition, obstacles can be avoided if $\min\{d_1, d_2, \ldots, d_n\} > 0$.

4 Description of multi-objective trajectory planning problem

In this study, minimization of total energy, joint jerks, and avoidance of obstacles are considered together to build a multi-criterion cost function. Singularity involved in null space and joint limits is considered a constraint.

4.1 Establishment of objective functions

1. For a given trajectory of the end effector, the smaller the joint velocity is, the less likely the joint angle mutation occurs. Therefore, the optimization of joint angular velocity has the effect of avoiding singularity. Meanwhile, the null space term is used to constrain the joint limit when solving inverse kinematics. Since the collision detection algorithm is also the kinematic solution of the manipulator in essence, the obstacle avoidance term is taken as a penalty function in this paper to fuse with the joint angular velocity optimization objective. The specific expression is

$$\mathrm{obt} = \begin{cases} \lambda_1 \cdot \max\{-d_1, -d_2, \ldots, -d_n\}, \\ \min\{d_1, d_2, \ldots, d_n\} > 0 \\ \lambda_2, \min\{d_1, d_2, \ldots, d_n\} < 0 \end{cases}, \tag{16}$$

where obt denotes obstacle avoidance term; λ_1 and λ_2 are scale factors.

Combining with Eq. (11), the kinematic optimization term can be expressed as

$$f_1 = \min \sum\nolimits_{i=1}^{n} (|\dot{q}_i|) + \mathrm{obt}. \tag{17}$$

Due to the scale factors, joint angular velocity is the major term when obstacles are avoided. Besides, the objective function is larger when the links failed to avoid the obstacle. This is convenient for choosing feasible solutions in the process of trajectory optimization.

2. For redundant manipulators, the frequent changes in the joint jerk would cause a great impact. Reducing the joint jerk can indirectly limit the change rate of joint torque and thus make the movement of the manipulator more stable. This effectively reduces the execution error of the manipulator and resonance caused by the movement and wear on the joint mechanism. The jerk is the third derivative of the joint angle regarding time, and its optimization objective function is

$$f_2 = \min \sum\nolimits_{i=1}^{n} \left(|{q'''}_i| \right). \tag{18}$$

3. Total energy consumption may significantly increase when the manipulator is trying to find an obstacle avoidance trajectory with a small joint jerk. This may result in frequent acceleration and deceleration, which should also be avoided in the manipulator's motion. The optimization objective function is

$$f_3 = \sum\nolimits_{i=1}^{n} \int_{t_0}^{t_f} [\dot{q}_i(t)\tau_i(t)]^2 \mathrm{d}t, \tag{19}$$

where τ_i denotes the torque of joint i. Joint torque is calculated by the following dynamics equations (Saramago and Junior, 2000).

$$\begin{aligned} \tau_i = & \sum\nolimits_{j=1}^{n} D_{ij}(q)\ddot{q}_j \\ & + \sum\nolimits_{j=1}^{n} \sum\nolimits_{k=1}^{n} C_{ijk}(q)\dot{q}_j\dot{q}_k \\ & + g_i(q), i = 1, \ldots, n, \end{aligned} \tag{20}$$

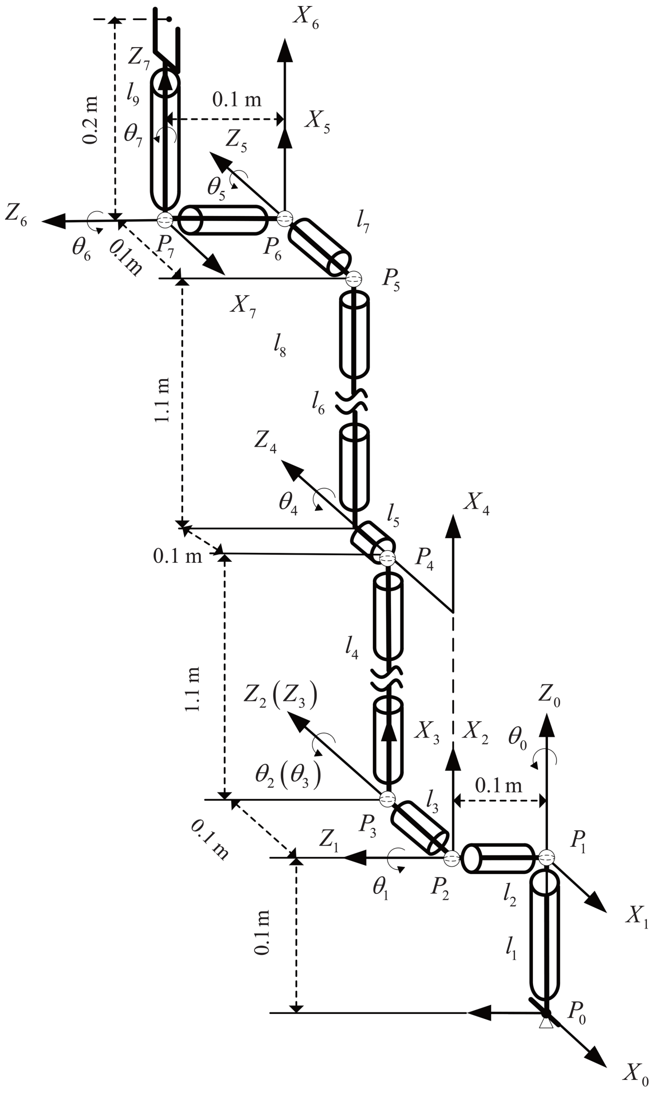


Figure 5. 7-DOF redundant manipulator.

where $C_{ijk} = \frac{\partial D_{ij}}{\partial q_k} - \frac{\partial D_{jk}}{2\partial q_i}$; $\sum_{j=1}^{n} D_{ij}(q)\ddot{q}_j$ indicates the acceleration term; D_{ij} represents inertial system matrix. $\sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}(q)\dot{q}_j\dot{q}_k$ stands for quadratic velocity term; C_{ijk} refers to Coriolis and centripetal forces matrix; $g_i(q)$ designates a configuration-dependent term that depends on the gravity.

4.2 Description of multi-objective trajectory planning problem

The trajectory of the manipulator's end effector is only decided by $(T_j\varphi_j)$ according to Eqs. (3) and (5). Different trajectories can be obtained by changing the values of $(T_j\varphi_j)$ in the cosine polynomial when initial conditions are given. Hence, T_j and φ_j are selected as the decision variables in the following optimization. The range of them is defined as the feasible region:

$$\begin{aligned}&\forall \boldsymbol{\xi} \in \boldsymbol{\Omega} : \left(T_{\min} \leq T_j \leq T_{\max}\right)\\ &\quad \wedge \left(\varphi_{\min} \leq \varphi_j \leq \varphi_{\max}\right), j = 1, 2, \ldots, 6.\end{aligned} \tag{21}$$

For redundant manipulators, the core of the multi-objective trajectory planning problem is to determine the decision vector $\boldsymbol{\xi}$ that will optimize the following objective vector:

$$\boldsymbol{F}(\boldsymbol{\xi}) = \left[f_1(\boldsymbol{\xi}), f_2(\boldsymbol{\xi}), f_3(\boldsymbol{\xi})\right]^{\mathrm{T}}. \tag{22}$$

$\boldsymbol{\xi}$ is defined in Eq. (7), which generates the trajectory of the end effector in Cartesian space. The objective vector $\boldsymbol{F} = [f_1, f_2, f_3]^{\mathrm{T}}$ is composed of the objective functions' values.

The constraints in the optimization problem include the initial and final.

During the trajectory planning process, the joint limit constraints are

$$q_{i\min} \leq q_i(t) \leq q_{i\max}. \tag{23}$$

$q_i(t)$ is angle of joint i at time t.

5 Trajectory planning based on multi-objective optimization

It is difficult to find a global minimum solution for multi-objective optimization problems (MOPs). In MOP (Liu et al., 2015), a vector $\boldsymbol{\xi}$ dominates $\boldsymbol{\xi}^*(\boldsymbol{\xi} \succ \boldsymbol{\xi}^*)$ when the following Pareto dominance principle is satisfied.

$$\begin{aligned}&f_i(\boldsymbol{\xi}) \leq f_i\left(\boldsymbol{\xi}^*\right), \quad i = 1, 2, 3, \ldots, N\\ &f_i(\boldsymbol{\xi}) < f_i\left(\boldsymbol{\xi}^*\right), \quad \exists i \in \{1, 2, 3, \ldots, N\}.\end{aligned} \tag{24}$$

The solution satisfying Eq. (24) is defined as a non-dominated solution of Eq. (22).

Respecting the given multi-objective trajectory optimization problem, the Pareto optimal solution set S is defined as

$$\{S\} = \left\{\boldsymbol{\xi} \in \boldsymbol{\Omega} \middle| \neg\exists \boldsymbol{\xi}^* \in \boldsymbol{\Omega}, \boldsymbol{\xi}^* \succ \boldsymbol{\xi}\right\}. \tag{25}$$

The mapping of S is a three-dimensional surface called Pareto front. Therefore, discovering as many non-dominated solutions as possible to cover all features of MOP is the key to solving the proposed trajectory planning problem.

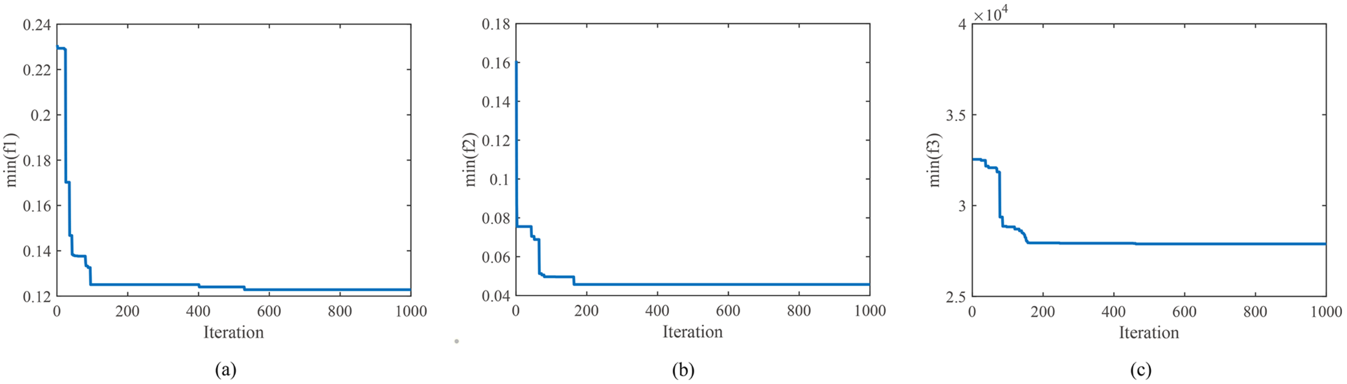
MOPSO algorithm is adopted to solve the multi-objective trajectory optimization problem. This algorithm employs an external repository of particles as guidance to guide the evolution of the whole population toward the Pareto front. Only non-dominated solutions can be stored in the repository according to the following process.

The velocity formula of particle's flight consists of not only the position and velocity of the current particle ($P[k]$, $V[k]$) but also the best position that all the particles ever had in the repository ($R[h]$):

$$\begin{aligned}V[k] = \chi \cdot \big[&V[k] + C_1R_1\left(P_{\text{best}}[k] - P[k]\right)\\ &+ C_2R_2\left(R[h] - P[k]\right)\big].\end{aligned} \tag{26}$$

Table 1. Mass parameters a 7-DOF redundant manipulator.

Part	m_i (kg)	The position of the bar center of mass in the joint coordinate system (m)	The inertia tensor in joint coordinates (kg m^2)
Link 1	75	[0, −0.05, 0]	Diag (0.0733, 0.0630, 0.0733)
Link 2	75	[0, 0.05, 0]	Diag (0.0733, 0.0360, 0.0733)
Link 3	25	[0.55, 0, 0]	Diag (0.1750, 14.2045, 14.2045)
Link 4	28	[0.391, 0, −0.054]	$\begin{pmatrix} 0.143 & 0 & -0.014 \\ 0 & 3.416 & 0 \\ -0.014 & 0 & 3.416 \end{pmatrix}$
Link 5	5	[0, 0, 0.05]	Diag (0.017, 0.017, 0.025)
Link 6	5	[0, 0, 0.05]	Diag (0.017, 0.017, 0.025)
Link 7	5	[0, 0, 0.1]	Diag (0.025, 0.079, 0.079)



(a) (b) (c)

Figure 6. Minimum value of objective functions during iteration. **(a)** Angular velocity. **(b)** Joint jerk. **(c)** Energy.

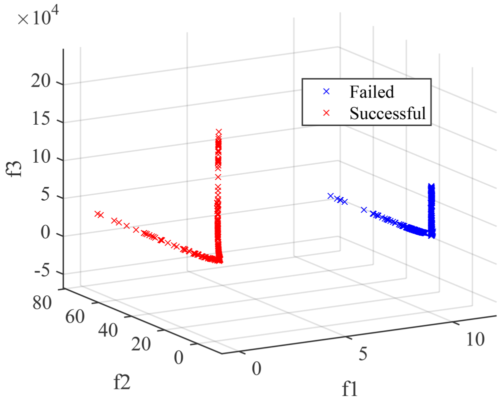


Figure 7. Pareto front.

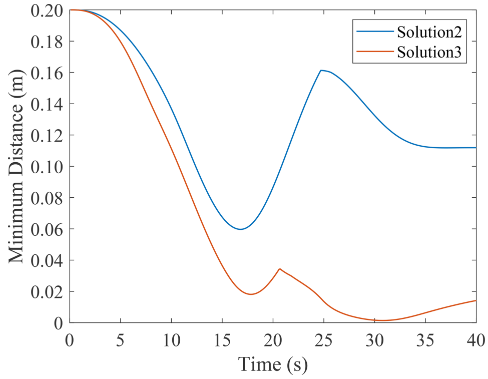


Figure 8. The minimum distances between center of obstacle and manipulator's links.

$\chi = 2/(|2-\phi-\sqrt{\phi(\phi-4)}|)$ represents the constriction factor; $\phi = C_1 + C_2$, where C_1 and C_2 refer to the learning factors regulating the step size of the particles flying to P_{best} (the best position a particle has ever explored) and the best position that all the particles ever had, respectively; R_1 and R_2 indicate independent random numbers in the range [0, 1].

Combining with the trajectory planning algorithm and MOPSO mentioned above, the obstacle avoidance trajectory optimization can be concluded as follows.

1. The population containing n_{P} particles is initialized, and the range of decision variables is defined as $[T_{\min}, T_{\max}]$ and $[\varphi_{\min}, \varphi_{\max}]$. Decision variables $\boldsymbol{\xi}$ are randomly

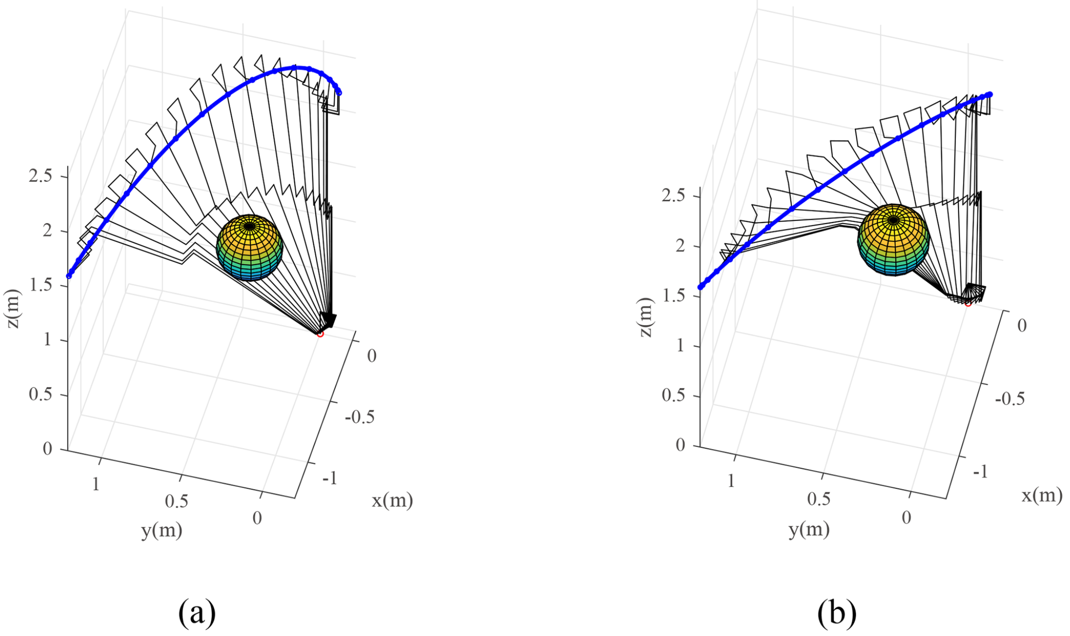


Figure 9. The configuration change of the manipulator and its geometric relationship with the obstacle. **(a)** Solution 2. **(b)** Solution 3.

generated, and the trajectory of the end effector is calculated by Eq. (1).

2. The joint trajectories are solved according to inverse kinematics, and then the values of three objective functions are obtained following kinematics and dynamics calculation results.

3. The repository is initialized, and its maximum allowable number is restricted to n_{R}. Next, the positions of particles that represent non-dominated decision vectors in the initial population are stored. Then, the best positions that the particles have had are initialized.

4. While the maximum number of $I_{\max}$ has not been reached, evolutionary computations are performed to update and maintain the repository. The velocities of particles are calculated using Eq. (26), and the position of each particle is updated. Besides, particles that failed to avoid the obstacle are eliminated. The reserved solutions are added into the repository according to the process as shown in Fig. 4. The loop counter is incremented until the maximum number is reached.

5. The non-dominated solutions are obtained from the repository. The particles constitute the set of non-dominant solution S in this paper.

6 Simulation results and analysis

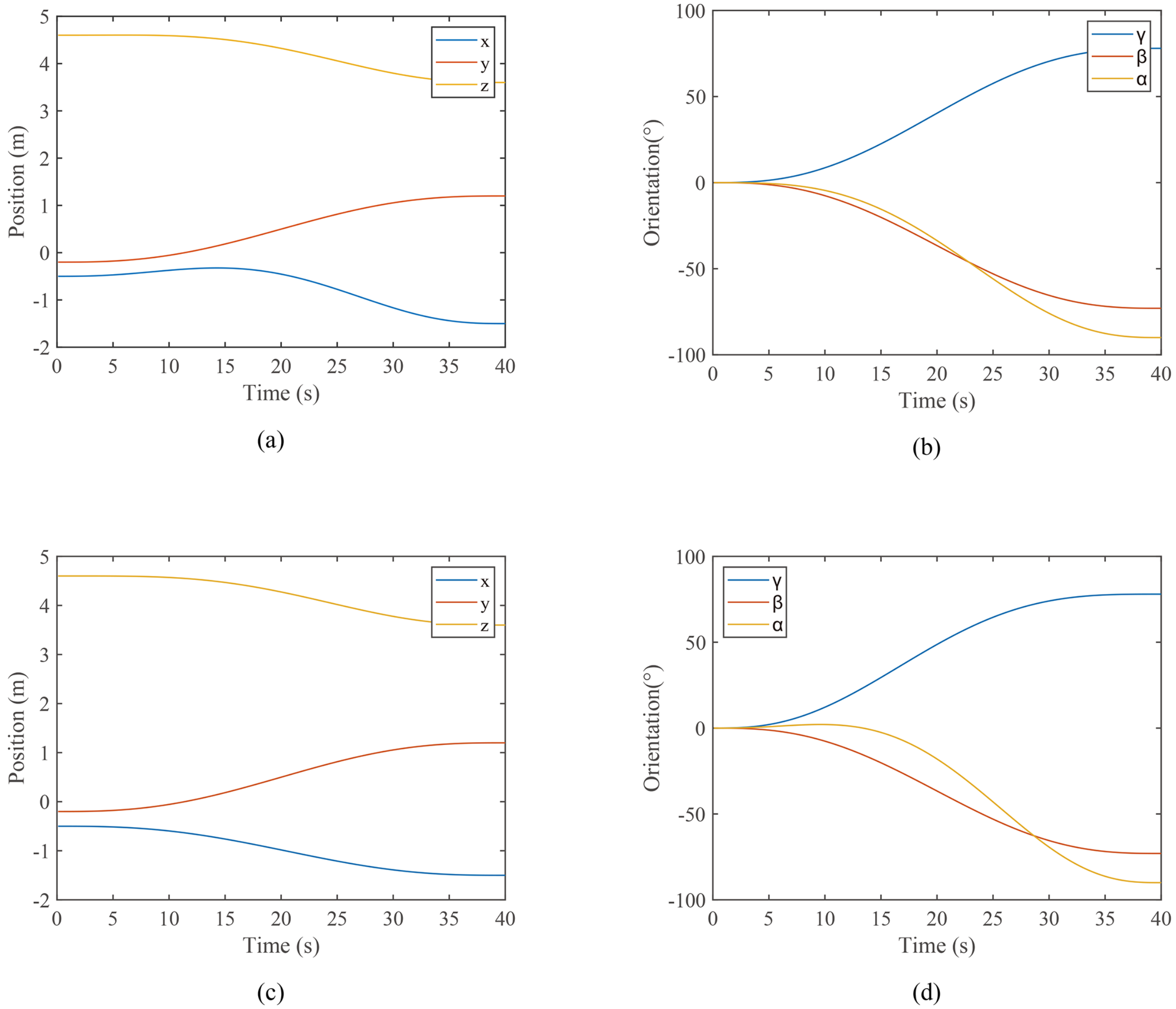
In this paper, a 7-DOF redundant manipulator is adopted to simulate the approached algorithm. The parameters are exhibited in Fig. 5. Mass parameters are provided in Table 1.

The initial configuration of the manipulator is $\boldsymbol{q}_{\mathrm{ini}} = [0, 0, 0, 0, 0, 0, 0°]$. The corresponding pose of the end effector is $\boldsymbol{X}_{\mathrm{ini}} = [-0.5, -0.2, 4.6\,\mathrm{m}, 0, 0, 0°]$. The desired pose of the end effector is $\boldsymbol{X}_{\mathrm{des}} = [-1.50, 1.20, 3.60\,\mathrm{m}, 78, -73, -90°]$. The total time of motion is $t_{\mathrm{f}} = 40\,\mathrm{s}$. The upper and lower limits of joint $i(i = 1, 2, \ldots, 7)$ are $q_{i\min} = -100$ and $q_{i\max} = 100°$, respectively. The radius of link i is $r_i = 0.1\,\mathrm{m}$. The radius of joint envelope sphere is $r_i^j = 0.1\,\mathrm{m}$. The obstacle is a sphere whose radius is $r_{\mathrm{S}} = 0.2\,\mathrm{m}$. The center of the obstacle is located in $\boldsymbol{Z}_0 = [-0.5, 0.3, 1.25\,\mathrm{m}]$. Scale factors in Eq. (16) are $\lambda_1 = 0.01$ and $\lambda_2 = 10$. The parameters of MOPSO are set as $[\lambda'_{\min1}, \lambda'_{\max1}] = [+10 \times 10^2, +10 \times 10^3]$, $[\lambda'_{\min2}, \lambda'_{\max2}] = [-\pi, +\pi]$, $n_{\mathrm{P}} = 1000$; $n_{\mathrm{R}} = 1500$; $C_1 = C_2 = 2.05$.

Figure 6 illustrates the minimum value of three objective functions during the iteration, which demonstrates the excellent astringency and stability of the proposed algorithm. The Pareto front is exhibited in Fig. 7. The particles are divided into two categories owing to the penalty function. Red particles satisfy the obstacle avoidance constraint while blue ones do not. Although the screening process can be executed to remove all blue solutions in each iteration, they are reserved to demonstrate the effectiveness of trajectory optimization for obstacle avoidance in this paper. According to the obstacle avoidance constraint, all the remaining non-dominant

Table 2. Decision vectors and objective vectors of S_2 and S_3.

Non-dominant solution	Decision vector	Objective vector
S_2	$[3.58,5.42,5.87,3.55,8.01,5.78] \times 10^3$ $[-1.59, -0.51, -1.58, 1.26, 0.64, -1.56]$	$[0.19, 0.05, 2.08 \times 10^5]$
S_3	$[3.29, 5.12, 6.80, 2.90, 8.05, 6.25] \times 10^3$ $[-1.39, -0.27, -1.57, 1.51, 0.86, -1.58]$	$[10.23, 58.08, 2.79 \times 10^4]$



(a)

(b)

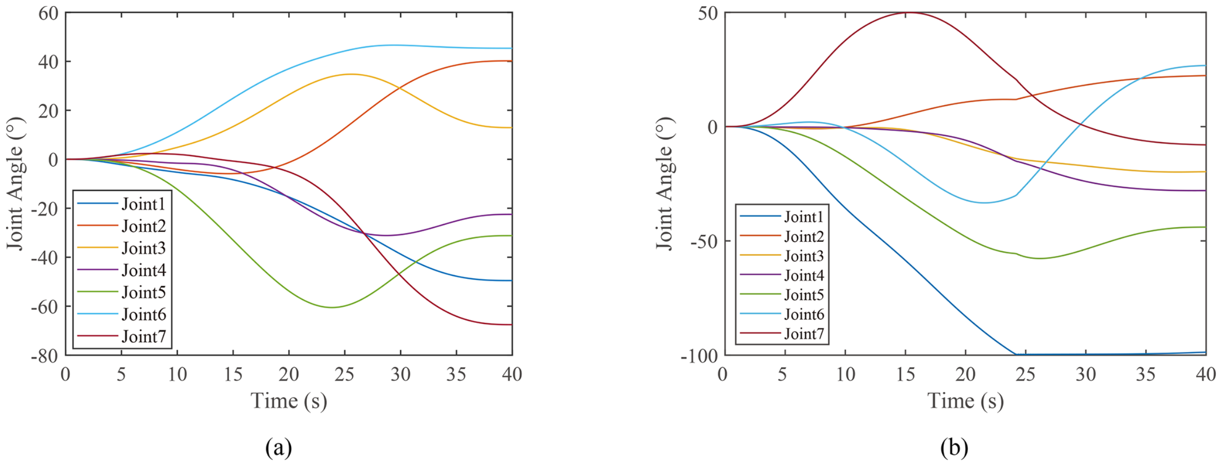
(c)

(d)

Figure 10. The position and orientation change of end effector. **(a)** The position changes of end effector in solution 2. **(b)** The orientation changes of end effector in solution 2. **(c)** The position changes of end effector in solution 3. **(d)** The orientation changes of end effector in solution 3.

solutions successfully avoid the obstacle after the particles that fail to avoid the obstacle are removed. This reflects that the proposed algorithm has a good performance in obstacle avoidance.

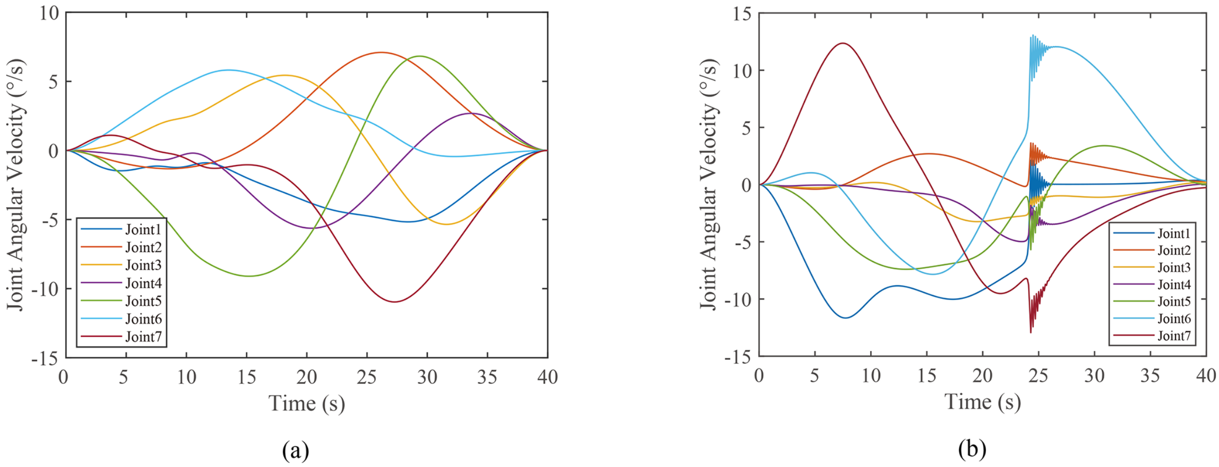
The first objective function is merely adopted to avoid singularity and obstacles. It will not affect the other physical performance of the redundant manipulator. Therefore, only f_2 and f_3 are considered in the solution selection. The extreme values of cost functions are $f_2^{\min} = 0.06\% f_2^{\max}$ and $f_3^{\min} = 13.41\% f_3^{\max}$, the specific values are $f_2^{\min} = 0.05$, $f_2^{\max} = 76.22$, $f_3^{\min} = 2.79 \times 10^4$ and $f_3^{\max} = 2.08 \times 10^5$. The jerks and energy costs corresponding to different end



(a)

(b)

Figure 11. The change of joint angular. **(a)** Solution 2. **(b)** Solution 3.



(a)

(b)

Figure 12. The change of joint angular velocities. **(a)** Solution 2. **(b)** Solution 3.

trajectories significantly vary, verifying the necessity of trajectory optimization.

Given no optimal solution to the multi-objective optimization problem, all the solutions are non-dominant. In this paper, the non-dominant solutions corresponding to the minimum values of f_2 and f_3 are selected for comparative analysis. It is assumed that the non-dominant solutions are S_2 and S_3. The decision vectors and objective vectors of S_2 and S_3 are presented in Table 2.

The minimum distances d_i between the center of obstacles and the manipulator's links are illustrated in Fig. 8. The minimum distances for S_2 and S_3 are 0.0597 and 0.0014 m, respectively. The trajectory obtained in S_2 and S_3 successfully avoided the obstacle. The configuration change of the manipulator and its geometric relationship with the obstacle are displayed in Fig. 9. The position and orientation change in the end effector are shown in Fig. 10. The manipulator reached the desired pose with different changing processes.

The change in joint angular over time is exhibited in Fig. 11. Since the joint angular velocity is one of the optimization objectives, no mutation of joint angle occurs. The change in joint angle is smooth. Moreover, the variation range of joint angle is also inhibited with the suppression of joint angular velocity. For solution S_3, the joint angle of link i is still suppressed due to the null-space term, though it approaches the limitation at 24 s. Figure 12 provides the change in joint angular velocities. For S_3, the joint angular velocity changes abruptly when the joint angle approaches the joint limit, forcing the joint angle of the manipulator to return to the range of the joint limit. This also validates the effectiveness of the null-space term as the joint angle constraint in this paper. The shaking process endured for 2 s and finally stabilized. Besides, lower energy cost always demands higher joint jerk, according to Fig. 7. It takes more energy to achieve a smooth angular velocity change, which reduces the impact on the joint mechanisms for redundant manipulators.

7 Conclusions

In this paper, a method of obstacle avoidance trajectory planning based on a multi-objective optimization algorithm in Cartesian space is proposed to avoid obstacles during the redundant manipulator's motion. On the premise that the end effector reaches the desired position and orientation at a specific time, obstacle avoidance can be achieved successfully, and joint jerk and energy consumption can also be inhibited. The results are described as follows.

1. The cosine polynomial function was employed to realize the task requirements of the end effector reaching the desired pose at a specific time. Then, the inverse kinematics and null space inverse solution were introduced to meet the joint angle limit constraint.

2. The multinomial weighted objective function is established in a combination of joint velocity and obstacle avoidance function. By optimizing the proposed function, the manipulator avoids not only the obstacle, but also the joint singularity.

3. Multi-objective particle swarm optimization was performed to optimize joint jerk and energy consumption simultaneously. The clear physical meaning of the objective functions is beneficial for the selection of non-dominant solutions.

The trajectory optimization method proposed in this paper possesses the strong flexibility and diversity. Obstacle avoidance trajectories can be obtained by inputting the initial and expected pose of the end effector. Several non-dominant solutions can be provided for selection. Therefore, this method has an acceptable prospect in engineering applications. However, the collision detection in this paper was implemented based on static obstacles, with all known environment variables. Thus, how to realize dynamic obstacle avoidance in an unknown environment needs to be investigated in the future.

Data availability. The data are available upon request from the corresponding author.

Author contributions. YL, XL, PJ, and ZD planned the campaign; PJ, ZD, BS, and ZW wrote code and performed the simulations; XL, XH, YL, and ZW analyzed the data; XL, BS, XH, and PJ wrote the manuscript draft; YL, XL, ZW, and ZD reviewed and edited the manuscript.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Financial support. This research has been supported by the National Natural Science Foundation of China (grant no. 52175013) and the Fundamental Research Funds for the Central Universities (grant no. JZ2020HGTB0034).

Review statement. This paper was edited by Zi Bin and reviewed by three anonymous referees.

References

Arakelian, V., Baron, J.-P. L., and Mottu, P.: Torque minimisation of the 2-DOF serial manipulators based on minimum energy consideration and optimum mass redistribution, Mechatronics, 21, 310–314, https://doi.org/10.1016/j.mechatronics.2010.11.009, 2011.

Bourbonnais, F., Bigras, P., and Bonev, I. A.: Minimum-Time Trajectory Planning and Control of a Pick-and-Place Five-Bar Parallel Robot, IEEE/ASME T. Mech., 20, 740–749, https://doi.org/10.1109/TMECH.2014.2318999, 2015.

Cao, H., Sun, S., Zhang, K., and Tang, Z.: Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm, Optik, 127, 9974–9983, https://doi.org/10.1016/j.ijleo.2016.07.078, 2016.

Chen, D. and Zhang, Y.: A Hybrid Multi-Objective Scheme Applied to Redundant Robot Manipulators, IEEE T. Autom. Sci. Eng., 14, 1337–1350, https://doi.org/10.1109/TASE.2015.2474157, 2017.

Cheng, F. T., Chen, T. H., Wang, Y. S., and Sun, Y. Y.: Obstacle avoidance for redundant manipulators using the compact QP method, Proceedings IEEE International Conference on Robotics and Automation (Cat. No. 93CH3247-4), Atlanta, GA, USA, 2–6 May 1993, 263, 262–269, https://doi.org/10.1109/ROBOT.1993.292186, 1993.

Coello Coello, C. A. and Lechuga, M. S.: MOPSO: a proposal for multiple objective particle swarm optimization, Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, 12–17 May 2002, 1052, 1051–1056, https://doi.org/10.1109/CEC.2002.1004388, 2002.

Dalla, V. K. and Pathak, P. M.: Curve-constrained collision-free trajectory control of hyper-redundant planar space robot, P. I. Mech. Eng. I-J. Sys., 231, 282–298, https://doi.org/10.1177/0959651817698350, 2017.

Guan, W., Weng, Z. X., and Zhang, J.: Obstacle avoidance path planning for manipulator based on variable-step artificial potential method, 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, 23–25 May 2015, 4325–4329, https://doi.org/10.1109/CCDC.2015.7162690, 2015.

Guigue, A., Ahmadi, M., Langlois, R., and Hayes, M. J. D.: Pareto Optimality and Multiobjective Trajectory Planning for a 7-DOF Redundant Manipulator, IEEE T. Robot., 26, 1094–1099, https://doi.org/10.1109/TRO.2010.2068650, 2010.

Han, T., Wu, H.-Y., Du, Z.-J., and Zheng, X.-J.: Research on path planning for manipulator to avoid real-time obstacle based on genetic algorithms, Appl. Res. Comput., 30, 1373–1376, https://doi.org/10.3969/j.issn.1001-3695.2013.05.023, 2013.

Kucuk, S.: Optimal trajectory generation algorithm for serial and parallel manipulators, Robot. Cim.-Int. Manuf., 48, 219–232, https://doi.org/10.1016/j.rcim.2017.04.006, 2017.

Lin, H.-I.: A Fast and Unified Method to Find a Minimum-Jerk Robot Joint Trajectory Using Particle Swarm Optimization, J. Intell. Robot. Syst., 75, 379–392, https://doi.org/10.1007/s10846-013-9982-8, 2014.

Liu, Y. and Zhang, X.: Trajectory optimization for manipulators based on external archives self-searching multi-objective particle swarm optimization, P. I. Mech. Eng. C-J. Mec., 0954406221997486, https://doi.org/10.1177/0954406221997486, 2021.

Liu, Y., Jia, Q., Chen, G., Sun, H., and Peng, J.: Multi-Objective Trajectory Planning of FFSM Carrying a Heavy Payload, Int. J. Adv. Robot. Syst., 12, 118 pp., https://doi.org/10.5772/61235, 2015.

Ma, Y. and Liang, Y.: An Obstacle Avoidance Algorithm for Manipulators Based on Six-Order Polynomial Trajectory Planning, Journal of Northwestern Polytechnical University, 38, 392–400, https://doi.org/10.3969/j.issn.1000-2758.2020.02.021, 2020.

Marcos, M. D. G., Tenreiro Machado, J. A., and Azevedo-Perdicoúlis, T. P.: A multi-objective approach for the motion planning of redundant manipulators, Appl. Soft. Comput., 12, 589–599, https://doi.org/10.1016/j.asoc.2011.11.006, 2012.

Park, S.-O., Lee, M. C., and Kim, J.: Trajectory Planning with Collision Avoidance for Redundant Robots Using Jacobian and Artificial Potential Field-based Real-time Inverse Kinematics, Int. J. Control Autom., 18, 2095–2107, https://doi.org/10.1007/s12555-019-0076-7, 2020.

Safeea, M., Bearee, R., and Neto, P.: Collision Avoidance of Redundant Robotic Manipulators Using Newton's Method, J. Intell. Robot. Syst., 99, 673–681, https://doi.org/10.1007/s10846-020-01159-3, 2020.

Salaris, P., Fontanelli, D., Pallottino, L., and Bicchi, A.: Shortest Paths for a Robot With Nonholonomic and Field-of-View Constraints, IEEE T. Robot., 26, 269–281, https://doi.org/10.1109/TRO.2009.2039379, 2010.

Saramago, S. F. P. and Junior, V. S.: Optimal trajectory planning of robot manipulators in the presence of moving obstacles, Mech. Mach. Theory, 35, 1079–1094, https://doi.org/10.1016/S0094-114X(99)00062-2, 2000.

Saravanan, R., Ramabalan, S., and Balamurugan, C.: Evolutionary collision-free optimal trajectory planning for intelligent robots, Int. J. Adv. Manuf. Tech., 36, 1234–1251, https://doi.org/10.1007/s00170-007-0935-x, 2008.

Shen, H., Wu, H., Chen, B., Ding, L., and Yang, X.: Obstacle Avoidance Algorithm for Redundant Robots Based on Transition between the Primary and Secondary Tasks, Robot, 36, 425–429, https://doi.org/10.13973/j.cnki.robot.2014.0425, 2014.

Shi, H., Chen, J., Pan, W., Hwang, K., and Cho, Y.: Collision Avoidance for Redundant Robots in Position-Based Visual Servoing, IEEE Syst. J., 13, 3479–3489, https://doi.org/10.1109/JSYST.2018.2865503, 2019.

Wang, W. R., Zhu, M. C., Wang, X. M., He, S., He, J. P., and Xu, Z. B.: An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators, Int. J. Adv. Robot. Syst., 15, 1729881418799562, https://doi.org/10.1177/1729881418799562, 2018.

Yan, L., Xu, W., Hu, Z., and Liang, B.: Multi-objective configuration optimization for coordinated capture of dual-arm space robot, Acta Astronaut., 167, 189–200, https://doi.org/10.1016/j.actaastro.2019.11.002, 2020.

Zghal, H., Dubey, R. V., and Euler, J. A.: Efficient gradient projection optimization for manipulators with multiple degrees of redundancy, P. IEEE Int. C. Robot. Auto., Cincinnati, OH, USA, 13–18 May 1990, 1002, 1006–1011, https://doi.org/10.1109/ROBOT.1990.126123, 1990.

Zhang, Z., Chen, S., Zhu, X., and Yan, Z.: Two Hybrid End-Effector Posture-Maintaining and Obstacle-Limits Avoidance Schemes for Redundant Robot Manipulators, IEEE T. Ind. Inform., 16, 754–763, https://doi.org/10.1109/TII.2019.2922694, 2020.