Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A distributed variable density path search and simplification method for industrial manipulators with end-effector's attitude constraints*#

Jin WANG†‡[1,2], Shengjie LI[1,2], Haiyun ZHANG[3], Guodong LU[1,2], Yichang FENG[1,2], Peng WANG[1,2], Jituo LI[1,2]

[1]*State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China*

[2]*Engineering Research Center for Design Engineering and Digital Twin of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China*

[3]*School of Robotics, Ningbo University of Technology, Ningbo 315211, China*

†E-mail: dwjcom@zju.edu.cn

Received Aug. 21, 2022; Revision accepted Dec. 7, 2022; Crosschecked Mar. 2, 2023

Abstract: In many robot operation scenarios, the end-effector's attitude constraints of movement are indispensable for the task process, such as robotic welding, spraying, handling, and stacking. Meanwhile, the inverse kinematics, collision detection, and space search are involved in the path planning procedure under attitude constraints, making it difficult to achieve satisfactory efficiency and effectiveness in practice. To address these problems, we propose a distributed variable density path planning method with attitude constraints (DVDP-AC) for industrial robots. First, a position–attitude constraints reconstruction (PACR) approach is proposed in the inverse kinematic solution. Then, the distributed signed-distance-field (DSDF) model with single-step safety sphere (SSS) is designed to improve the efficiency of collision detection. Based on this, the variable density path search method is adopted in the Cartesian space. Furthermore, a novel forward sequential path simplification (FSPS) approach is proposed to adaptively eliminate redundant path points considering path accessibility. Finally, experimental results verify the performance and effectiveness of the proposed DVDP-AC method under end-effector's attitude constraints, and its characteristics and advantages are demonstrated by comparison with current mainstream path planning methods.

Key words: Path planning; Industrial robots; Distributed signed-distance-field; Attitude constraints; Path simplification
https://doi.org/10.1631/FITEE.2200353 **CLC number:** TP242

1 Introduction

Autonomous path planning of industrial robots is of great significance for improving their ease of use and making them meet the requirements of different tasks. When considering task requirements and constraints, the path planning of industrial robots in real applications becomes more complicated and challenging. First, the path planning of mobile robots is generally carried out in the two-dimensional space, but the search space of six-degree-of-freedom (6-DOF)

‡ Corresponding author

* Project supported by the Key R&D Program of Zhejiang Province, China (Nos. 2020C01025 and 2020C01026), the National Natural Science Foundation of China (No. 52175032), the "Pioneer" and "Leading Goose" R&D Program of Zhejiang Province, China (No. 2022C01054), and the Robotics Institute of Zhejiang University (Nos. K12107 and K11808)

Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2200353) contains supplementary materials, which are available to authorized users

ORCID: Jin WANG, https://orcid.org/0000-0003-3106-021X

industrial robots is three-dimensional or higher. Second, the constraint relationship between the links must be considered, and the whole robot cannot be regarded as a point, so collision detection is much more complex. In addition, the inverse kinematic solution of the robot will cause additional computation costs. This study will focus on efficient path planning in the Cartesian space.

Collision-free path planning has been widely used in automatic driving, unmanned aerial vehicle (UAV) navigation, game development, and other fields, and some classical theories and algorithms have been proposed, e.g., A* (Hart et al., 1968), rapidly-exploring random tree (RRT) (LaValle, 1998), and Dijkstra's algorithm (Dijkstra, 1959). On this basis, some new algorithms have been derived. Koenig and Likhachev (2005) proposed the D* lite algorithm based on the lifelong planning A* (LPA*) (Koenig et al., 2004) algorithm. Its main feature is incremental search, and the current position is regarded as the starting position after the environment changes, so as to ensure that the path is the shortest. On this basis, Sun et al. (2010) proposed an improved D* lite algorithm for moving targets, which makes up for the deficiency of the D* algorithm in moving target search. Grid-based search in the Cartesian space is a feasible method, but the traditional grid-based search limits the search direction of each step. Ferguson and Stentz (2006) therefore proposed the field D* algorithm, which adjusts the search direction based on linear interpolation and solves the problem of an excessive number of grids with multi-resolution grids. Inspired by the RRT algorithm, Persson and Sharf (2014) combined the classic A* algorithm with sampling and proposed sampling-based A* (SBA*), which improves the planning success rate. However, this method still requires many samples to ensure the planning success. Collision-free path planning based on artificial potential field methods is also an important direction (Adeli et al., 2011; Baziyad et al., 2021).

Based on the basic algorithm research and considering the characteristics of the industrial robot path planning problem, some related studies have been carried out. Ademovic and Lacevic (2014) proposed an evolutionary algorithm based on trees of bubbles of free C-space. This algorithm improves the reliability, but the redundant path is long. Fu et al. (2018) improved pre- and post-processing based on SBA*. It could improve the planning efficiency and solve the problem of redundant path points, but the planning efficiency may be reduced when post-processing does not work. Xie et al. (2020) found the defect of the distorted configuration space (DCS) method and presented an optimized IDCS method to shorten the generated path, but this method is not available in some cases. Qureshi and Ayaz (2016) combined RRT* with an artificial potential field to reduce memory utilization and improve the convergence rate. In view of the impact of collision detection on planning efficiency, Han et al. (2018) proposed an adaptive discrete collision detection method, which adjusts the collision model in the Cartesian space according to the distance between robots and obstacles to improve efficiency and accuracy. Hernández et al. (2014) proposed an improved adaptive A* (AA*) algorithm, named multipath adaptive A* (MPAA*), which can reuse paths found by previous A* searches. The path is proved to be optimal, and the search is carried out in the Cartesian space. Abele et al. (2016) defined multiple working points in the Cartesian space, solved the traveling salesman problem based on the A* algorithm, and realized time-optimal path point sequence planning.

To sum up, there are three main factors that affect planning efficiency: inverse kinematics, collision detection, and planning algorithm design. The current path planning research for industrial robots confronts the following problems in three aspects:

End-effector's attitude control: Most of the current mainstream motion planning algorithms, such as covariant Hamiltonian optimization for motion planning (CHOMP) (Zucker et al., 2013), RRT-connect (Kuffner and LaValle, 2000), and stochastic trajectory optimization for motion planning (STOMP) (Kalakrishnan et al., 2011), are solved in the configuration space, which makes it difficult to directly control the attitude of the end-effector. To cope with end-effector's attitude constraints in the Cartesian space, the conventional inverse kinematic analytical method is generally complicated and will generate multiple solutions, and the selection of solutions is time-consuming.

Excessive collision detection: Collision detection of industrial robots has a high time complexity. Besides, to ensure that the continuous trajectory is

reachable, it will be repeated in the planning process, resulting in a large computational load.

Unnecessary twists: The grid-based path search will generate unnecessary twists in the path, and a significant number of inverse kinematic solutions and collision detections are needed to perform path simplification to remove redundant path points, which has a great impact on efficiency.

In this paper, a variable density path-planning and simplification method based on a distributed signed-distance-field (DSDF) is proposed. The major work and contributions of this study are as follows:

Inverse kinematics: In tasks that require attitude constraints, accurate positioning is needed only at the starting and ending points, but not in the process of motion. Therefore, a position–attitude constraints reconstruction (PACR) inverse kinematic solution approach is proposed, which can satisfy the attitude constraints in the intermediate process and improve the efficiency of solving inverse kinematics.

Collision detection: For efficient collision detection, the models of DSDF and single-step safety sphere (SSS) are proposed. DSDF solves the problem of low reconstruction efficiency of the original SDF, and SSS can help reduce the number of collision detections during single-step search and path simplification.

Path simplification: Aiming at the inherent defect of grid-based search, a novel forward sequential path simplification approach (FSPS) is proposed to eliminate the redundant path points and speed up the process of path simplification.

2 Problem statement

We consider the handling task of a 6-DOF industrial robot with attitude constraints, which is very common in robot applications, such as industrial scenarios and scientific research, as shown in Fig. 1.

In a material handling task, the robot motion planning needs to meet the following requirements:

1. The robot needs to be precisely positioned at the starting and ending points, so as to grasp and release materials with the appropriate position and attitude (PA).

2. Obstacle avoidance constraints and joint limitations should be considered in the intermediate movement process, which is the premise of the feasible path.

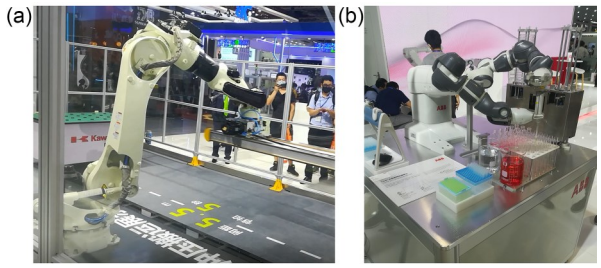

Fig. 1 Robot handling scenarios with attitude constraints: (a) material handling; (b) scientific research

3. Since the objects being moved are not always in a closed state, as shown in Fig. 2, it is necessary to ensure that the material will not overturn or fall during the movement, so attitude constraints for the end-effector are essential.

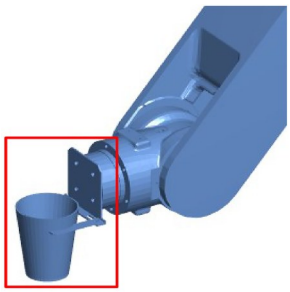

Fig. 2 Material handling without form closure

4. Unlike processing tasks, the intermediate process of handling tasks does not require precise positioning, which provides more possibilities for improving the efficiency of the planning algorithm.

For the above requirements, the attitude of the end-effector can be directly controlled based on inverse kinematics, which requires the construction and analysis of the kinematic model of the robot. According to the Denavit–Hartenberg (DH) parameters shown in Table 1, the robot kinematic model is defined (Fig. 3).

Table 1 Denavit–Hartenberg (DH) parameters of 6-DOF industrial robots

i	α_i (°)	a_i	d_i	θ_i
1	90.0	a_1	d_1	θ_1
2	0.0	a_2	0.0	θ_2+90°
3	90.0	a_3	0.0	θ_3
4	−90.0	0.0	d_4	θ_4
5	90.0	0.0	0.0	θ_5−90°
6	0.0	0.0	d_6	θ_6

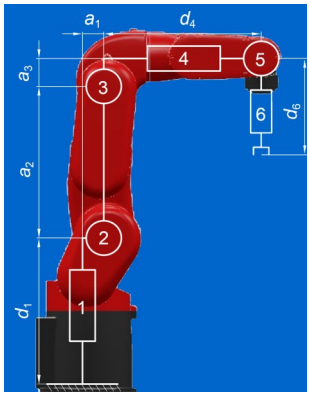

Fig. 3 Kinematic model of industrial robots

The path planning in the Cartesian space can be regarded as the planning of the PA sequence of the manipulator's end-effector. The PA of the manipulator can be described by a homogeneous matrix:

$$ {}^{0}_{6}\boldsymbol{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & p_z & p_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1} $$

where the rotation matrix in the left dashed box defines the attitude of the end-effector, and the position vector in the right dashed box defines the position of the end-effector. According to the requirements of the transportation task analyzed above, two assumptions are proposed:

1. Precise control of PA is necessary at the starting and ending points of the trajectory, while attitude control accuracy is more important than position control accuracy in the process of motion to avoid collision.

2. All the obstacles in the scene are rigid bodies.

Based on the above assumptions, the algorithm framework and details of this study are proposed and detailed in the following sections.

3 Methodology

3.1 Overview of DVDP-AC

The framework of the distributed variable density path planning method with attitude constraints (DVDP-AC) is inspired by D* lite. D* lite is a path planning algorithm with incremental search. It can use the information obtained from previous search steps to improve the efficiency of the subsequent planning (Koenig and Likhachev, 2005).

However, due to the limitations of grid-based search, the original D* lite suffers from two drawbacks. First, planning time increases dramatically as the number of grid cells increases. Second, the paths often have unnecessary twists, which makes the paths longer than desired. In addition, although great achievements have been made in the research of collision detection, the conventional collision-free path planning methods still spend most of their calculation time on collision detection.

Considering the above problems and characteristics of industrial robots, the framework of DVDP-AC is constructed, and the kinematic solution, collision detection, search method, and path simplification method are all improved. The steps are as follows:

1. Initialize the orthogonal variable density search space and collision models.

2. The improved kinematic model, collision models, and search space are used for path planning in the initial state.

3. If the robot does not reach the goal cell, search for the node with the lowest cost, then update the robot position, and the node information is updated according to the situation of obstacles.

4. If the robot reaches the goal cell, post-process the obtained path (path simplification) based on FSPS.

In Fig. 4, the yellow box represents the improvements adopted, which will be elaborated in the following subsections.

3.2 PACR inverse kinematics

At present, most industrial robots satisfy the Pieper criterion (Liu HS et al., 2015; Liu YY et al., 2021), which enables the inverse kinematic analytical solution of the robot. However, solving the inverse kinematics of a 6-DOF robot is still a complex problem, and since there are multiple solutions, it is necessary to consider how to choose the optimal one. In the planning process, inverse kinematics needs to be solved many times, so it is helpful to improve the efficiency. For a robot satisfying the Pieper criterion, it is generally believed that the end position is controlled mainly by the first three joints, and the end attitude is controlled mainly by the last three joints.

Based on this characteristic, the PACR inverse kinematic solution method is proposed.

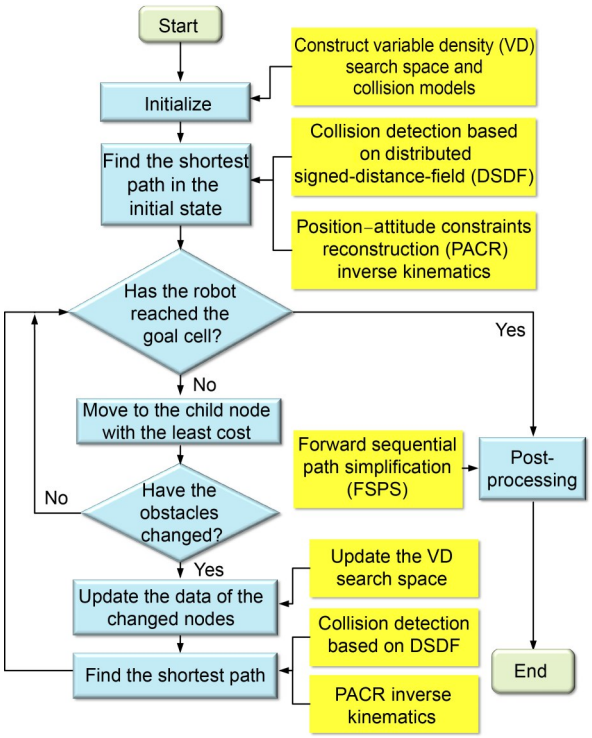

Fig. 4 Framework of DVDP-AC (References to color refer to the online version of this figure)

3.2.1 Deconstruction and reconstruction of the homogeneous position and attitude matrix

In general, path planning in the Cartesian space involves planning the PAs of the end-effector, including solving the robot configuration based on inverse kinematics and carrying out collision detection. Solving the inverse kinematics of a 6-DOF robot requires computing multiple solutions and selecting the optimal one. To improve the planning efficiency of intermediate path points, it is necessary to consider how inverse kinematics is applied in path planning.

As mentioned above, the end-effector homogeneous matrix contains attitude information and position information, and the terminal PA sequence of the intermediate path points describes the movement process of the manipulator. However, according to the assumption described in Section 2, the end position in the intermediate process does not require high precision, and only the PAs of the starting point and the ending point need to be highly accurate. Therefore, for the starting and ending points of the path, the inverse kinematics needs to be solved by the traditional method, while for the intermediate path points, the homogeneous matrix can be reconstructed, and the reconstructed matrix can be used to describe the movement process of the manipulator:

$$ {}^{0}_{6}\boldsymbol{T}_{\mathrm{D}}=\begin{bmatrix} n_{ex} & o_{ex} & a_{ex} & p_{wx} \\ n_{ey} & o_{ey} & a_{ey} & p_{wy} \\ n_{ez} & o_{ez} & a_{ez} & p_{wz} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2} $$

where ${}^{0}_{6}\boldsymbol{T}_{\mathrm{D}}\in\mathbb{R}^{4\times4}$, the rotation matrix in the left dashed box still describes the attitude of the end-effector, while the position vector in the right dashed box is changed to describe the wrist position of the robot, which means that in the middle waypoint planning, each point in the search space corresponds to the wrist rather than the end-effector. Thus, the position vector is related only to the first three joint angles:

$$ \begin{bmatrix} P_{wx} \\ P_{wy} \\ P_{wz} \end{bmatrix}=f_P(\boldsymbol{\theta}_{\mathrm{F}}), \tag{3} $$

where

$$ \boldsymbol{\theta}_{\mathrm{F}}=[\theta_1 \quad \theta_2 \quad \theta_3]^{\mathrm{T}}, \tag{4} $$

where θ_1, θ_2, and θ_3 are the first three joint angles.

As for attitude constraints, the rotation matrix is affected by all the joint angles as high precision control is still required:

$$ \begin{bmatrix} n_{ex} & o_{ex} & a_{ex} \\ n_{ey} & o_{ey} & a_{ey} \\ n_{ez} & o_{ez} & a_{ez} \end{bmatrix}=f_{\mathrm{A}}(\boldsymbol{\theta}), \tag{5} $$

where

$$ \boldsymbol{\theta}=[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6]^{\mathrm{T}}, \tag{6} $$

where θ_4, θ_5, and θ_6 are the last three joint angles.

For the above PA constraints, a novel PACR inverse kinematic solution method is proposed. Specifically, the robot applicable to the PACR method should conform to the Pieper criterion, and its structure and DH parameters are shown in Fig. 3 and Table 1, respectively. The principles of PACR are explained in detail below.

3.2.2 PACR: solving the first three joint angles based on the wrist position constraint

Since the wrist coordinates (x, y, z) are known during the search, the solution process of the first three joints can be greatly simplified and complex matrix operations can be saved. In other words, it can be solved with a geometric method. Note that there are two sets of the first three joint angles that allow the wrist to reach the same position, as shown in Fig. 5. However, considering the practical situation, the configuration of Fig. 5b is generally not adopted. Therefore, only the calculation method for the configuration of Fig. 5a is given.

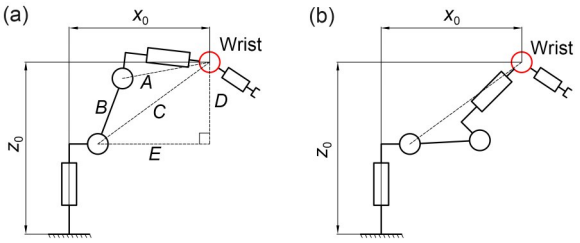

Fig. 5 Two configurations corresponding to the same position: (a) common situation; (b) special situation

The first three joint angles are calculated as follows:

$$\theta_1 = \begin{cases} \theta_{1\_1}, & \text{if abs}(\theta_{1\_1} - \theta_{1\text{pre}}) \le \text{abs}(\theta_{1\_2} - \theta_{1\text{pre}}), \\ \theta_{1\_2}, & \text{if abs}(\theta_{1\_2} - \theta_{1\text{pre}}) < \text{abs}(\theta_{1\_1} - \theta_{1\text{pre}}), \end{cases} \tag{7}$$

$$\theta_2 = \arctan(D, E\cdot\text{sign}(E_x\cos\theta_1 + E_y\sin\theta_1)) + \arccos\left(\frac{B^2 + C^2 - A^2}{2BC}\right) - \frac{\pi}{2}, \tag{8}$$

$$\theta_3 = -1\cdot\left\{\pi - \arccos\left[(A^2 + B^2 - C^2)/(2AB)\right] - \arctan(d_4, a_3)\right\}. \tag{9}$$

In Eq. (7), $\theta_{1\text{pre}}$ is the previous value of θ_1, and θ_{1_1} and θ_{1_2} are possible solutions of θ_1, which is chosen according to the principle of minimum joint movement. θ_{1_1} and θ_{1_2} are calculated as follows:

$$\begin{cases} \theta_{1\_1} = \arctan(y, x), \\ \theta_{1\_2} = -1\cdot\text{sign}(\theta_{1\_1})\cdot(\pi - \text{abs}(\theta_{1\_1})). \end{cases} \tag{10}$$

In Eqs. (8) and (9), A to E are defined as shown in Fig. 5a, and E_x and E_y are components of E along x and y axes, respectively. The calculation formulae of A to E are shown in Table 2. In addition, considering the initial value of the second joint, π/2 should be added when θ_2 is involved in subsequent steps.

Table 2 Brief introduction of each group

Symbol	Expression
A	$\sqrt{a_3^2 + d_4^2}$
B	a_2
C	$\sqrt{(x - a_1\cos\theta_1)^2 + (y - a_1\sin\theta_1)^2 + (z - d_1)^2}$
D	$z - d_1$
E	$\sqrt{(x - a_1\cos\theta_1)^2 + (y - a_1\sin\theta_1)^2}$
E_x	$x - a_1\cos\theta_1$
E_y	$y - a_1\sin\theta_1$

3.2.3 PACR: solving the last three joint angles based on the attitude constraint of the end-effector

According to the attitude of the end-effector of the starting and ending points, the attitude interpolation can be carried out in the form of a quaternion, and then the quaternion can be converted into a matrix form to solve the angles of the last three joints.

Since the angles of the first three joints are known, the homogeneous matrix of these joints can be calculated according to

$${}_i^{i-1}\boldsymbol{T} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

The homogeneous matrix of the above joints is shown in Fig. 6. Then the homogeneous matrix of the last three joints is obtained:

$${}_6^3\boldsymbol{T} = {}_3^2\boldsymbol{T}^{-1}\cdot{}_2^1\boldsymbol{T}^{-1}\cdot{}_1^0\boldsymbol{T}^{-1}\cdot{}_6^0\boldsymbol{T} = \begin{bmatrix} {}_6^3T_{1,1} & {}_6^3T_{1,2} & {}_6^3T_{1,3} & {}_6^3T_{1,4} \\ {}_6^3T_{2,1} & {}_6^3T_{2,2} & {}_6^3T_{2,3} & {}_6^3T_{2,4} \\ {}_6^3T_{3,1} & {}_6^3T_{3,2} & {}_6^3T_{3,3} & {}_6^3T_{3,4} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

Thus, θ_5 can be calculated as follows:

$$\theta_5 = \begin{cases} \theta_{5\_1}, & \text{if abs}(\theta_{5\_1} - \theta_{5\text{pre}}) \le \text{abs}(\theta_{5\_2} - \theta_{5\text{pre}}), \\ \theta_{5\_2}, & \text{if abs}(\theta_{5\_2} - \theta_{5\text{pre}}) < \text{abs}(\theta_{5\_1} - \theta_{5\text{pre}}), \end{cases} \tag{13}$$

where $\theta_{5\text{pre}}$ is the previous value of θ_5, and θ_{5_1} and θ_{5_2} are possible solutions of θ_5, which is chosen according to the principle of minimum joint movement. θ_{5_1} and θ_{5_2} are calculated as follows:

$$\begin{cases} \theta_{5\_1} = \arccos {}^3_6T_{3,3} + \pi/2, \\ \theta_{5\_2} = -\arccos {}^3_6T_{3,3} + \pi/2. \end{cases} \tag{14}$$

Similar to the second joint, $-\pi/2$ should be added when θ_5 is involved in subsequent steps.

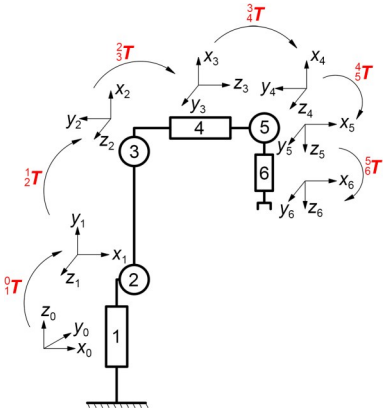

Fig. 6 Transformation matrices of the joints

Since matrix operations take more time, ${}^3_6T_{3,3}$ is usually computed by

$$\begin{aligned} {}^3_6T_{3,3} = {} & a_{ex}\cos\theta_1\sin(\theta_2 + \pi/2 + \theta_3) + a_{ey}\sin\theta_1\sin(\theta_2 \\ & + \pi/2 + \theta_3) - a_{ez}\cos(\theta_2 + \pi/2 + \theta_3), \end{aligned} \tag{15}$$

where a_{ex}, a_{ey}, and a_{ez} are the elements of the matrix in Eqs. (2) and (5). ${}^3_6T_{1,1}$, ${}^3_6T_{2,1}$, ${}^3_6T_{1,3}$, ${}^3_6T_{2,3}$, ${}^3_6T_{3,1}$, and ${}^3_6T_{3,2}$ can be calculated similary, and the specific equations will not be described here. If ${}^3_6T_{3,3} = \pm 1$, the wrist of the manipulator is in a singular state, and thus θ_4 and θ_6 are calculated as follows:

$$\theta_4 = \theta_{4\text{pre}}, \tag{16}$$

$$\theta_6 = \begin{cases} \arctan({}^3_6T_{2,1}, {}^3_6T_{1,1}) - \theta_4, & \text{if } {}^3_6T_{3,3} = 1, \\ -\arctan(-{}^3_6T_{2,1}, -{}^3_6T_{1,1}) + \theta_4, & \text{if } {}^3_6T_{3,3} = -1, \end{cases} \tag{17}$$

where $\theta_{4\text{pre}}$ is the previous value of θ_4.

In general, θ_4 and θ_6 can be solved according to θ_5:

$$\theta_4 = \arctan\left(\frac{{}^3_6T_{2,3}}{\sin(\theta_5 - \pi/2)}, \frac{{}^3_6T_{1,3}}{\sin(\theta_5 - \pi/2)}\right), \tag{18}$$

$$\theta_6 = \arctan\left(\frac{{}^3_6T_{3,2}}{\sin(\theta_5 - \pi/2)}, -\frac{{}^3_6T_{3,1}}{\sin(\theta_5 - \pi/2)}\right). \tag{19}$$

Since the motion process is required to be continuous, it is feasible and necessary to select the solution in advance when calculating θ_1 and θ_5. The multi-solution problem can be solved by using a pre-selection solution and posing a constraint separation, because other solutions will cause an abrupt attitude change of the end-effector, resulting in the material falling in the process of transportation. The proposed PACR method not only meets the practical requirement, but also reduces the calculation needed to find solutions and select the optimal one.

3.3 Collision detection acceleration based on DSDF and SSS

Collision detection takes up much time in the planning process. In recent years, collision detection methods based on triangular mesh have been continuously improved. For example, collision detection based on bounding volume hierarchy (BVH) mesh screening is the current mainstream (Li et al., 2018; Tan et al., 2020). In addition, for path planning, there is a more efficient collision detection approach, which is based on SDF. Generally, the sphere-chain collision model is built for robots and SDF is built for the surrounding environment:

1. The original model of the robot is abstracted into a collision model formed by a sphere chain. The initial sphere chain is generated based on an oriented bounding box (Gottschalk et al., 1996), and the radii and center coordinates of all balls are adjusted based on the genetic algorithm (Harik et al., 1999) to minimize the total volume of all bounding spheres. The value of C_c can be defined according to the radius of the smallest sphere of the sphere chain.

2. Collision models of obstacles are constructed based on SDF. SDF consists of a series of sampling point clouds, and each point of SDF has a signed number, whose absolute value is the shortest distance from the obstacles. If the sampling point is outside the obstacle, the signed number is positive; otherwise, it is negative. In this way, collision detection can be

carried out quickly by comparing the value of the sampling point with the radius of the collision sphere.

An obvious problem with traditional SDF is that it needs to be rebuilt when the surrounding obstacles change, and the reconstruction process is often time-consuming, making SDF-based collision detection difficult to apply in a dynamic scenario. To solve this problem, the concept of DSDF is proposed, based on the second assumption in Section 2. The principle is that DSDF is constructed in advance for the obstacle, and when the obstacle moves or rotates, its DSDF moves and rotates accordingly. Because the obstacles are rigid bodies, complex reconstruction is avoided.

The range of DSDF is enlarged by d_e on the basis of axis-aligned bounding boxes (AABB) (Xing et al., 2010). d_e is the radius of the maximum sphere of the moving parts, as shown in Fig. 7. The collision models of robot and obstacle are shown in Fig. 8. Some papers have proposed the method of constructing SDF (Huo et al., 2014; Klingensmith et al., 2015), so the construction process will not be repeated here.

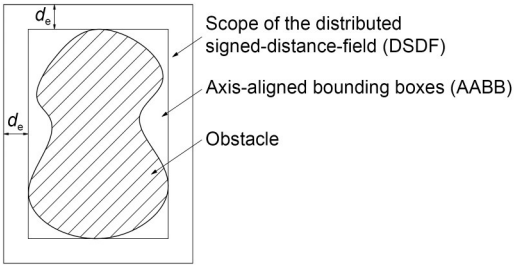

Fig. 7 Scope of DSDF for a single obstacle

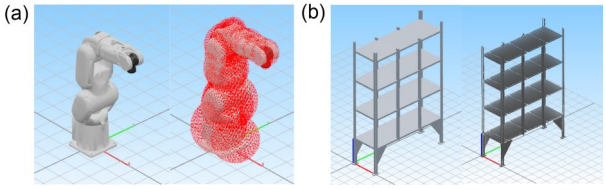

Fig. 8 The robot collision model based on sphere chain (a) and obstacle collision model based on DSDF (b)

In this paper, the collision detection of a robot is carried out based on its collision spheres. First, it is determined whether the center of the collision sphere is within the sampling range of DSDF. If not, the robot is deemed to be collision-free; if so, the coordinates of the nearest point from the center of the ball can be found with

$$\begin{cases} x_s = x_{\min} + \text{round}\left(\dfrac{x_c - x_{\min}}{d_i}\right) \cdot d_i, \\ y_s = y_{\min} + \text{round}\left(\dfrac{y_c - y_{\min}}{d_i}\right) \cdot d_i, \\ z_s = z_{\min} + \text{round}\left(\dfrac{z_c - z_{\min}}{d_i}\right) \cdot d_i, \end{cases} \tag{20}$$

where (x_c, y_c, z_c) is the center of the bounding ball, (x_s, y_s, z_s) is the sampling point closest to the center of the ball, $(x_{\min}, y_{\min}, z_{\min})$ is the point with the smallest coordinate values in all directions among the sampling points of DSDF, and d_i is the sampling interval. If the signed number of the sampling point is greater than the radius of the bounding box, it is deemed to be a collision.

In addition, to improve the efficiency of single-step search and reduce the computation of continuous collision detection, a safety zone model named SSS is proposed, as shown in Fig. 9.

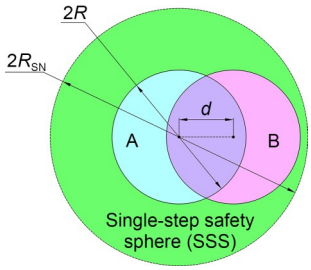

Fig. 9 Diagram of safe state determination

In Fig. 9, A is the collision sphere before movement, B is the collision sphere after movement, d is the displacement, R_{SN} is the signed value of the sampling point closest to the center of the collision sphere, and it is also the radius of the SSS. The principle of continuous collision detection acceleration is to directly judge whether the single-step movement is collision-free, by evaluating whether the sweep area of the collision sphere is covered by SSS. If the single-step sweep area of the collision ball cannot be covered by SSS, continuous collision detection is required.

To ensure accurate judgment, there are two special cases that need to be considered, as shown in Fig. 10. Fig. 10a presents the situation in which the collision sphere breaks through the SSS at the end of the single step, and Fig. 10b presents the situation in

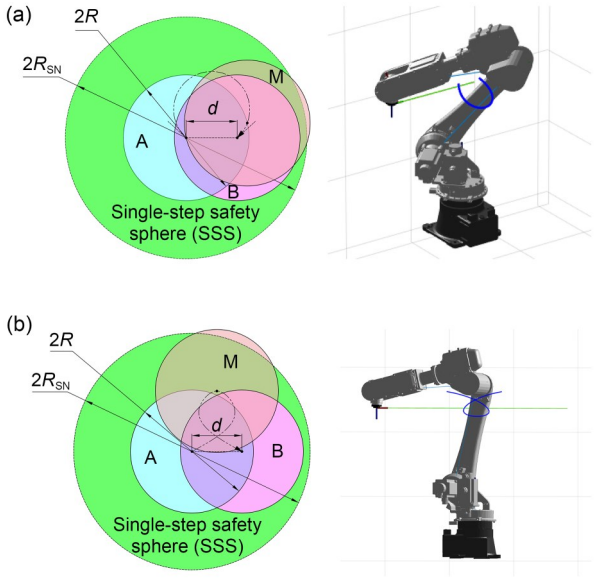

Fig. 10 The unsafe state at the end (a) and intermediate (b) segments of single-step movement (References to color refer to the online version of this figure)

which the collision sphere breaks through the SSS during the intermediate stage of the single step. In Fig. 10, the green line is the path of the end flange, and the blue curve is the moving track of the center of the elbow collision sphere.

For the first case, it can be inferred from inequality (21) whether the motion trend of the sphere at the end segment is safe:

$$d_{n-1} \le d, \tag{21}$$

where d_{n-1} represents the distance between the penultimate interpolation point of the single-step trajectory and the center of the sphere in state A. If the above formula does not hold, continuous interpolation is required.

For the second case, Algorithm 1 can be used to deduce whether the intermediate stage is safe.

With the exception of the above two cases, most cases are shown in Fig. 11, in which the single-step movement of wrist is linear and the step size is small. Then the single-step safety state can be determined by

$$d + R < R_{SN}. \tag{22}$$

If inequality (22) is true, the single-step search is safe.

Specifically, for DSDF, the collision sphere is safe under the following conditions:

Algorithm 1 Safety state discrimination based on DSDF-SSS

Input: collision sphere radius, R; configurations of the starting and ending points of a local path; radius of SSS, R_{SN}

1: According to the preset interpolation interval, the local path of single-step motion is interpolated, and the number of interpolation points is n;
2: **if** $n \le 6$
3: return **false**;
4: **end**
5: According to the configurations of the first three points, the center coordinates of the collision spheres (P_A, P_2, P_3) are calculated;
6: According to the configurations of the last three points, the center coordinates of the collision spheres (P_{n-2}, P_{n-1}, P_B) are calculated;
7: Compute vectors $\overrightarrow{P_AP_B}$, $\overrightarrow{P_AP_2}$, $\overrightarrow{P_2P_3}$;
8: Calculate the angle θ_{A1} between $\overrightarrow{P_AP_B}$ and $\overrightarrow{P_AP_2}$ as well as θ_{A2} between $\overrightarrow{P_AP_B}$ and $\overrightarrow{P_2P_3}$;
9: **if** $\theta_{A1} \le \theta_{A2}$ or $\theta_{A1} > \pi/2$
10: return **false**;
11: **end**
12: Compute vectors $\overrightarrow{P_BP_A}$, $\overrightarrow{P_BP_{n-1}}$, $\overrightarrow{P_{n-1}P_{n-2}}$;
13: Calculate the angle θ_{B1} between $\overrightarrow{P_BP_A}$ and $\overrightarrow{P_BP_{n-1}}$ as well as the angle θ_{B2} between $\overrightarrow{P_BP_A}$ and $\overrightarrow{P_{n-1}P_{n-2}}$;
14: **if** $\theta_{B1} \le \theta_{B2}$ or $\theta_{B1} > \pi/2$
15: return **false**;
16: **else**
17: return **true;**
18: **end**

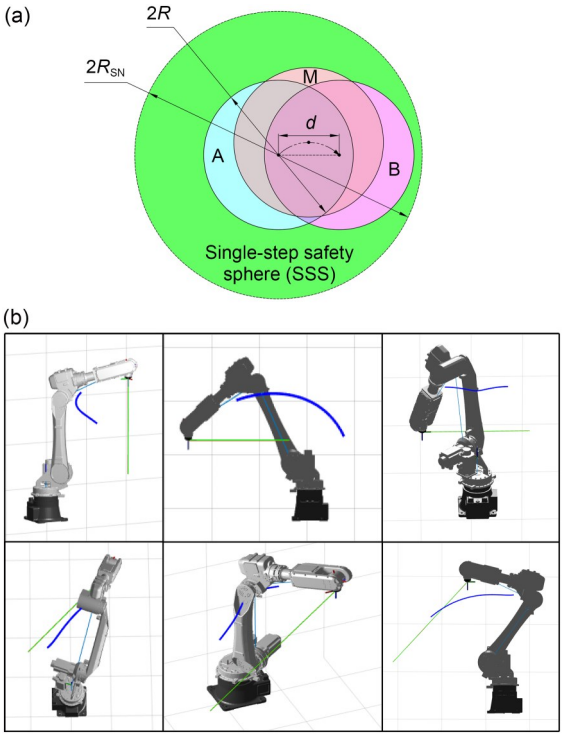

Fig. 11 The safe state of single-step movement (a) and corresponding motion process in general (b)

1. The sphere is outside DSDF and the distance from DSDF is greater than R.

2. Before the single-step motion, the sphere is in DSDF and the signed number of the sampling point closest to its center is greater than the safe distance (d+R).

3.4 Variable density path-planning

In robot working scenarios, the distribution of obstacles is usually uneven. That is to say, there are densely distributed areas and sparsely distributed areas. To ensure security, it is necessary to increase the mesh density of the search space. However, for the area where obstacles are sparsely distributed, an over-dense grid will reduce the search efficiency. Therefore, in this study we adopt a search space generation method that adaptively adjusts the mesh density according to the distribution of obstacles. Its principle is introduced below.

The octree is generated based on the robot workspace and the search space is initially established. When the manipulator is close to the obstacles, more precise planning is needed to ensure that collision is avoided. Therefore, the collision clearance C_c and the minimum scale in the planning process should first be defined. Then the AABB of the obstacles is constructed which can delineate the area of an obstacle at a low computational cost. The grid cells intersecting with AABB should be further refined until the size of the leaf node is not greater than the collision clearance C_c, as shown in Fig. 12.

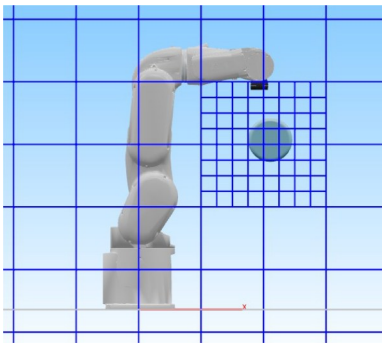

Fig. 12 Variable density grid based on octree

The subdivided area is called the intensive area, and the grid cells without collision are defined as safe area cells. Thus, the construction of the variable density search space is preliminarily completed. In the subsequent planning process, if a collision occurs in the unrefined areas, the robot will return to the state of the previous step and search after the mesh is refined. Due to the presence of variable density areas, it is infeasible to search at a constant step size, so the search is carried out in a segmented way. According to the grid density, the search space can be divided into a safe area and an intensive area. No matter whether the next step is in the safe area or the intensive area, Eq. (23) can be used to quickly calculate the location of the center point:

$$\begin{cases} x'_{\mathrm{s}} = x'_{\min} + \mathrm{round}\left(\dfrac{x_{\mathrm{m}} - x'_{\min}}{d_{\mathrm{s}}}\right) \cdot d_{\mathrm{s}}, \\ y'_{\mathrm{s}} = y'_{\min} + \mathrm{round}\left(\dfrac{y_{\mathrm{m}} - y'_{\min}}{d_{\mathrm{s}}}\right) \cdot d_{\mathrm{s}}, \\ z'_{\mathrm{s}} = z'_{\min} + \mathrm{round}\left(\dfrac{z_{\mathrm{m}} - z'_{\min}}{d_{\mathrm{s}}}\right) \cdot d_{\mathrm{s}}, \end{cases} \tag{23}$$

where $(x_{\mathrm{m}}, y_{\mathrm{m}}, z_{\mathrm{m}})$ are the coordinates after moving one step, $(x'_{\min}, y'_{\min}, z'_{\min})$ is the point with the smallest coordinate values in all directions among the sampling points in the area in which $(x_{\mathrm{m}}, y_{\mathrm{m}}, z_{\mathrm{m}})$ falls, and d_{s} is the search step size. Note that using variable density search sometimes results in longer paths because large grids may lengthen local paths, as shown in Fig. 13.

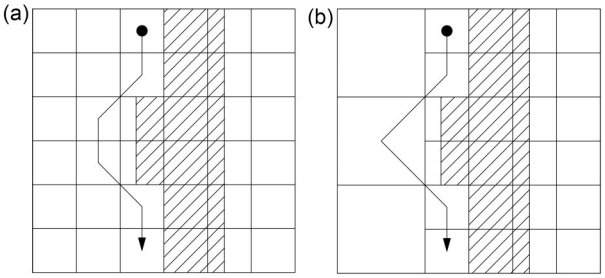

Fig. 13 The difference of paths caused by uniform density search (a) and variable density search (b)

To solve this problem, according to the positions of the child nodes, additional path points are added to the local paths that pass through only the grid nodes of the safe area, and then the path is further simplified. In the next subsection, a novel approach will be introduced.

3.5 Forward sequential path simplification

Since the path planning in this study is based on grid-based search, there are redundant bends in the path. To achieve optimal planning results, it is necessary to remove the redundant path points. Existing methods tend to start with the farthest nodes in terms of connection relationships, such as the method of Fu et al. (2018):

1. Define node p_i at the starting point.

2. Establish the local paths between p_i and subsequent non-adjacent path points p_j.

3. Find the collision-free local path which contains node p_j with the maximum subscript j, use p_i–p_j to replace the corresponding segment of the initially planned path, and then assign j to i.

4. If there is no collision-free local path, i=i+1.

5. Repeat steps 2–4 until p_j reaches the last point.

It can be seen that the simplification order in Fu et al. (2018) is from back to front, and the simplification method proposed in this study is improved in two aspects:

First, the method of FSPS is proposed. The details are as shown in Algorithm 2.

Algorithm 2 Forward sequential path simplification (FSPS)

Input: the initial path, interpolation interval (I_{FSPS})

```
Define a pair of nodes (N_A and N_B), and at first N_B is defined at the first node, and N_A is two nodes ahead of N_B;
while N_A does not reach the goal node do
    N_A traversal forward;
    if the local path N_A–N_B is collision-free do
        Replace the corresponding segment of the initially planned path with N_A–N_B;
    else
        Move N_B to the previous node of N_A;
    end
end
```

The above process is shown in Fig. 14. The red dashed lines are the local paths with collision, the green dashed lines are the local paths without collision, and the solid green lines are the local paths adopted. The novel simplification method can effectively reduce the number of redundant path points. Moreover, the time complexity is $O(n)$ in the worst case, which is lower than that of the existing method, $O(n^2)$. In addition, this method can reduce collision detection of long local paths, such as linear paths from the starting point to the goal point. The longer the distance, the more times collision detection is performed, and avoiding collision detection of long paths can further improve efficiency. Therefore, the proposed method has smaller time fluctuation in different situations and can improve efficiency. In addition, the idea of variable density can be used for collision detection and reachability detection, that is, to reduce the number of interpolation points of the local path in the safe area.

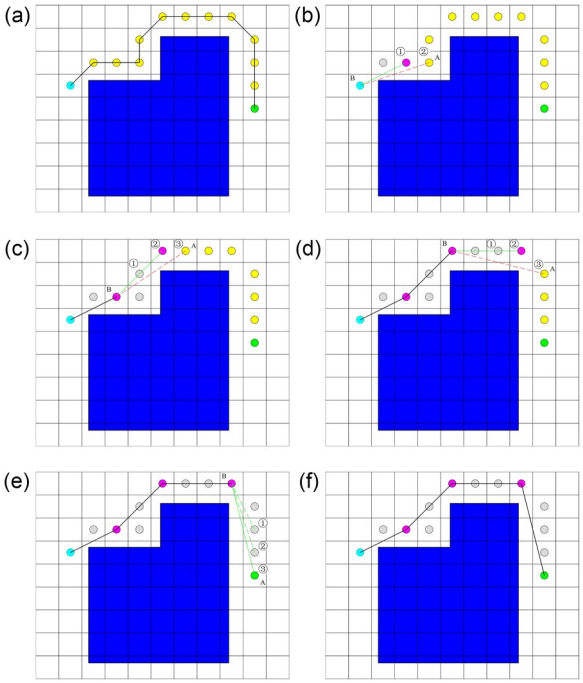

Fig. 14 The process of forward sequential path simplification: (a) original path; (b) finding the second path point; (c) finding the third path point; (d) finding the fourth path point; (e) finding the fifth path point; (f) completing the simplification (References to color refer to the online version of this figure)

Second, some redundant points can be eliminated by checking whether they are collinear, which takes little time to reduce the number of collision detections, as shown in Fig. 14d. In addition, to solve the problem shown in Fig. 13, a condition needs to be added: if only the collinear redundant path points are eliminated in a certain step, and the path is not shortened, this step of path simplification should be rejected, and the process should continue from the next point.

4 Validation and discussion

4.1 Ablation study

For a preliminary analysis of the effect of the proposed DVDP-AC method, a series of simulation cases were set up for testing. The computer used for the simulation was equipped with an AMD Ryzan 3600 CPU with 16 GB memory, whose base frequency was 3.6 GHz, and a GTX1070 GPU was used to accelerate the construction of DSDF. The simulation environment was built based on C++ and OpenGL in Windows 10. Six groups were set for each case to analyze the contribution of different measures, as shown in Table 3.

Table 3 Brief introduction of each group

Group	Inverse kinematics	Collision detection	Search space	Post-processing
OBUO	O	B	U	O
OSUO	O	S	U	O
OSVO	O	S	V	O
MSVO	M	S	V	O
MSVM	M	S	V	M
MSVN	M	S	V	N

O: original; M: modified; B: bounding volume hierarchy (BVH) based; S: signed-distance-field (SDF) based; U: uniform; V: variable; N: none

MSVM uses the DVDP-AC method proposed in this study, and OBUO uses the original D* lite algorithm (Koenig and Likhachev, 2005). The simulation cases consist of a robot, a conveyor belt, and two shelves, simulating the processes of taking out the workpiece from the shelf and placing it on the conveyor belt, as illustrated in Fig. 15. Based on this, the starting point, ending point, and positions of obstacles are changed to form 10 different scenes. The calculation time and the length of the cases are shown in Table 4.

To verify the effectiveness of the proposed algorithm in the real environment, tests were carried out in a static scenario (case 1) and a dynamic scenario (case 2). The setups of these two cases are shown in Fig. 16. The path planning was repeated 10 times in each case to reduce the error. In case 1, there is a robot, a computer numerical control (CNC) milling machine, a couple of aluminum brackets (obstacles), and a table. The robot was ABB IRB120, required to avoid obstacles and move to the CNC milling machine

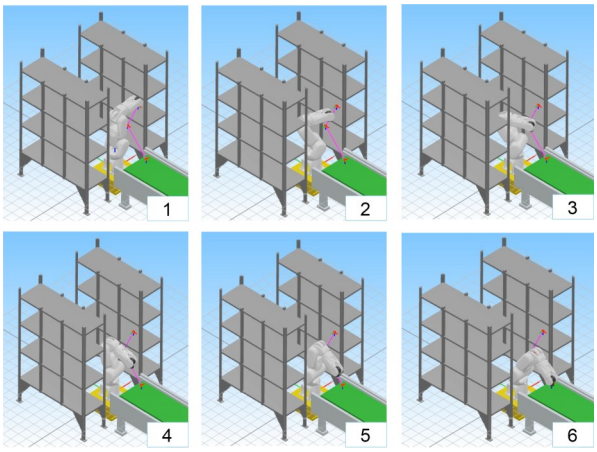

Fig. 15 Motion process of the simulation cases

Table 4 Performance comparison in the simulation cases

Group	Average time (ms)	Maximum time (ms)	Minimum time (ms)	Average length (mm)
OBUO	38.24	65.65	28.74	751.31
OSUO	32.65	53.92	26.13	756.17
OSVO	30.69	50.43	24.94	756.17
MSVO	26.18	45.57	19.67	761.43
MSVM	19.99	31.04	14.74	737.11
MSVN	17.79	28.24	14.27	917.17

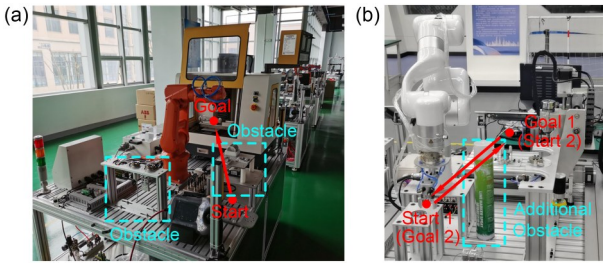

Fig. 16 The setups of case 1 (a) and case 2 (b)

(Fig. 16a). In case 2, the robot was required to complete loading and unloading actions. The robot was xArm 6, which does not satisfy the Pieper criterion, so the simplified kinematic model was not adopted. In this case, five groups were reset for each path, as shown in Table 5, where the meanings of the letters are the same as those in Table 3. While waiting for processing, an obstacle was added so that the manipulator cannot return along the original path (Fig. 16b). According to the framework of DVDP-AC, if the obstacles were found to be changed, the search space will be rebuilt and the planning will be carried out again. The calculation time and path length are shown in Tables 6 and 7.

Table 5 Brief introduction of each group in case 2

Group	Inverse kinematics	Collision detection	Search space	Post-processing
OBUO	O	B	U	O
OSUO	O	S	U	O
OSVO	O	S	V	O
OSVM	O	S	V	M
OSVN	O	S	V	N

O: original; M: modified; B: bounding volume hierarchy (BVH) based; S: signed-distance-field (SDF) based; U: uniform; V: variable; N: none

Table 6 Performance comparison of case 1

Group	Average time (ms)	Maximum time (ms)	Minimum time (ms)	Length (mm)
OBUO	196.37	206.83	180.54	1006.08
OSUO	111.10	115.33	108.91	1006.08
OSVO	100.57	104.39	98.62	1006.08
MSVO	92.74	97.06	89.54	1172.55
MSVM	78.37	82.04	75.67	1049.53
MSVN	55.57	58.18	53.95	1295.52

Table 7 Performance comparison of case 2

Group	Average time (ms)	Maximum time (ms)	Minimum time (ms)	Length (mm)
OBUO	2748.51	2764.42	2719.25	2233.58
OSUO	2001.04	2009.75	1994.36	2219.94
OSVO	1990.14	1998.12	1975.50	2219.94
OSVM	1739.87	1754.54	1722.70	2221.03
OSVN	1356.66	1367.09	1337.79	3174.12

Since four paths need to be planned in case 2, the path length in Table 7 is the sum of the four paths, and the average time, maximum time, and minimum time are all calculated based on the total planning time. In Tables 6 and 7, there is a big difference in planning time, which is related mainly to the path length, obstacle distribution, robot structure, and other factors. Among them, the path length will affect the number of search steps, the obstacle distribution will affect the time required for collision detection, and the robot structure will affect the efficiency of an inverse kinematic solution. In addition, as mentioned before, Table 7 shows the total planning time of the four paths, and thus the gap is further increased.

4.2 Baseline comparison

CHOMP (Zucker et al., 2013), RRT-connect (Kuffner and LaValle, 2000), and Bi-Fast-Marching-Tree algorithm (BFMT*) (Starek et al., 2015) were chosen as the benchmarks to measure the performance of the proposed DVDP-AC. CHOMP and RRT-connect are both mainstream algorithms in the field of manipulator motion planning. CHOMP can efficiently plan a reachable path in a configuration space based on covariant Hamiltonian optimization. RRT-connect is known for its bidirectional search, which can effectively improve the efficiency of an RRT algorithm. RRT-connect can solve the feasible path not only with a high success rate in a narrow search space, but also with a high efficiency in a wide search space. BFMT* is a bidirectional version of FMT*, which can quickly solve complex motion planning problems in high-dimensional configuration spaces (Janson et al., 2015). In this study, BFMT* needs only to find variable paths, so additional optimization will be reduced to further improve its efficiency. The hardware conditions of this test were the same as the previous ones. The simulation environment was built based on ROS Noetic, which ran on an Ubuntu 20.04 operating system.

To demonstrate the performance of the above algorithms in terms of planning efficiency and end-attitude constraints, a 6-DOF manipulator was adopted, which operated in a scene that included some obstacles such as cups and tables, as shown in Fig. 17. In case 3, the robot was instructed to deliver four cups to points 1 to 4 in sequence. Since cups usually contain liquid, attitude control is necessary. For the above algorithms, the parameters are taken as follows.

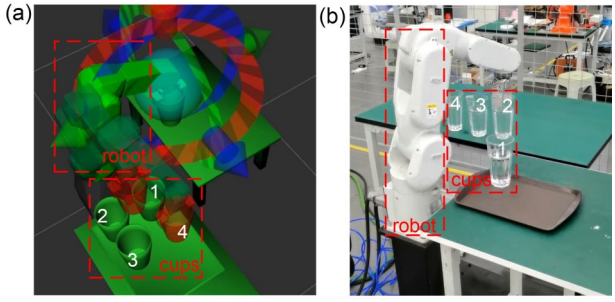

Fig. 17 The setup of case 3: (a) simulation environment; (b) experimental environment

4.2.1 DVDP-AC

The parameters of DVDP-AC and their introduction are shown in Table 8.

Table 8 Brief introduction of the parameters of DVDP-AC

Symbol	Meaning	Value (m)
C_c	Minimum distance required to avoid obstacles	0.01
s_{min}	Step size in the dense grid search space	0.01
s_{max}	Step size in the sparse grid search space	0.04
I_{FSPS}	Interpolation interval used for the collision detection in path simplification	0.002

4.2.2 CHOMP

The parameters of CHOMP and their introduction are shown in Table 9.

Table 9 Brief introduction of the parameters of CHOMP

Symbol	Meaning	Value
W_s	Weight of smoothness cost	0.1
W_o	Weight of obstacle cost	1.0
R_l	Learning rate	0.01
C_c	Minimum distance required to avoid obstacles	0.2 m
T_c	Threshold required to terminate the iteration according to the collision-free condition	0.07
n_i	Maximum number of iterations	200
n_{acf}	Maximum number of iterations after finding a feasible path	5

4.2.3 RRT-connect

The parameters of RRT-connect and their introduction are shown in Table 10. The larger the s_r, the higher the exploration efficiency. However, when s_r is too large, the trajectory quality and success rate will also be affected. The smaller the s_r, the finer the trajectory, but the lower the exploration efficiency.

Table 10 Brief introduction of the parameter of RRT-connect

Symbol	Meaning	Value
s_r	Step size in the configuration search space	0.075 rad

Since the obstacles of case 3 are sparse, a larger s_r can be adopted to improve efficiency.

4.2.4 BFMT*

The parameters of BFMT* and their introduction are shown in Table 11. K_o is used to determine the conditions for the end of the algorithm. When K_o is true, the algorithm ends when the best path is found; otherwise, the algorithm ends when the first viable path is found.

Table 11 Brief introduction of the parameters of BFMT*

Symbol	Meaning	Value
n_s	Number of samples taken by the planner	150
K_o	Does it have to find the best path?	False
r_m	Multiplier used for the nearest neighbor search radius	30

To enhance the reliability of the conclusions, the planning of each path was repeated 10 times, and the experimental data were recorded and processed, as shown in Table 12. According to Table 12, the DVDP-AC method showed high efficiency under variable and known obstacle distribution, and performed better in both planning efficiency and end-attitude controllability. In the process of planning the above four trajectories, the DVDP-AC algorithm reduced the average time by more than 39% compared with

Table 12 Comparison of planning time

Path	Time (ms)	DVDP-AC	CHOMP	Decrease rate 1	RRT-connect	Decrease rate 2	BFMT*	Decrease rate 3
1	Average	34.9	109.0	67.98%	49.3	29.21%	48.8	28.48%
	Maximum	40.0	111.0	63.96%	55.0	27.27%	52.0	23.08%
	Minimum	34.0	108.0	68.52%	45.0	24.44%	45.0	24.44%
2	Average	63.7	123.6	48.46%	103.0	38.16%	88.9	28.35%
	Maximum	77.0	128.0	39.84%	117.0	34.19%	96.0	19.79%
	Minimum	59.0	119.0	50.42%	92.0	35.87%	82.0	28.05%
3	Average	69.5	118.3	41.25%	95.8	27.45%	76.3	8.91%
	Maximum	72.0	121.0	40.50%	103.0	30.10%	82.0	12.20%
	Minimum	69.0	116.0	40.52%	84.0	17.86%	70.0	1.43%
4	Average	65.1	126.4	48.50%	101.1	35.61%	71.2	8.57%
	Maximum	69.0	132.0	47.73%	122.0	43.44%	78.0	11.54%
	Minimum	63.0	122.0	48.36%	86.0	26.74%	67.0	5.97%

CHOMP. Specifically, in the scenario of the first path, DVDP-AC saved more time because of the sparse distribution of obstacles, finding the feasible path within 40 ms, while CHOMP took more than 100 ms. Compared with RRT-connect and BFMT*, DVDP-AC can maintain a significant efficiency advantage, reducing the average planning time by 8% to 38%. What is more, by observing the motion process, it can be found that the algorithms of the three control groups failed to maintain the proper attitude of the end-effector, as shown in Fig. 18.

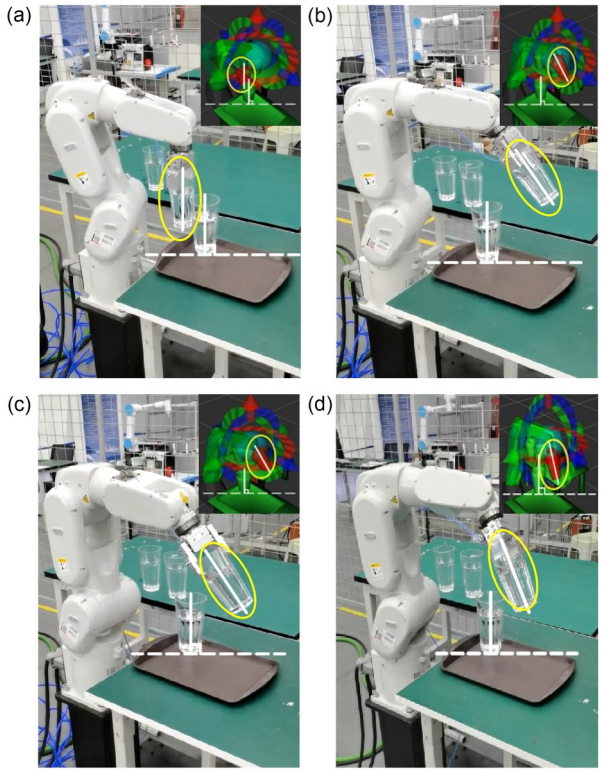

Fig. 18 Effect comparison among four methods of the attitude constraint: (a) DVDP-AC; (b) CHOMP; (c) RRT-connect; (d) BFMT*

4.3 Discussion

According to the above experimental results, in all test scenarios, the above improvement measures (DSDF, PACR, and FSPS) can improve the planning efficiency. The effects of the measures are discussed in detail below.

4.3.1 DSDF-based collision detection

Based on DSDF and SSS, DVDP-AC can quickly determine whether the local path is safe, which greatly reduces the number of collision detections, thus effectively improving the overall efficiency.

According to Tables 4, 6, and 7, in test cases, the average calculation time was reduced by 14.63% to 27.20% by adopting DSDF and SSS. The analysis results of the ablation study are shown in Fig. 19. In addition, by using SSS to reduce the number of collision detections, DVDP-AC achieved an efficiency improvement in baseline comparison.

4.3.2 PACR inverse kinematics

As a basic improvement, PACR inverse kinematics improved efficiency in all scenarios. Although the original analytical solution is already efficient, the inverse kinematic solution needs to be carried out many times in path planning, so simplified inverse kinematics can effectively improve the efficiency. In the test cases, the average calculation time was reduced by 7.79% to 14.68% by adopting the PACR method. The analysis results are shown in Fig. 20. Moreover, in baseline comparison, although DVDP-AC requires an additional inverse kinematic solution, which affects the computational efficiency, the PACR method can effectively reduce this impact.

4.3.3 FSPS

The effects of FSPS are as follows: first, it reduces the computational complexity when there are few redundant path points; second, it directly eliminates the collinear path points; third, it reduces the number of feasibility detections of the safety zone. Therefore, the effect is clear when there are sparse obstacles, few redundant path points, or many collinear path points. In the test cases, the average calculation time can be reduced by 12.58% to 23.64% due to FSPS, as shown in Fig. 21.

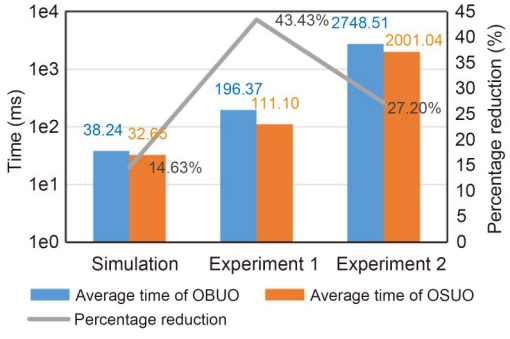

Fig. 19 Effect of DSDF-based detection

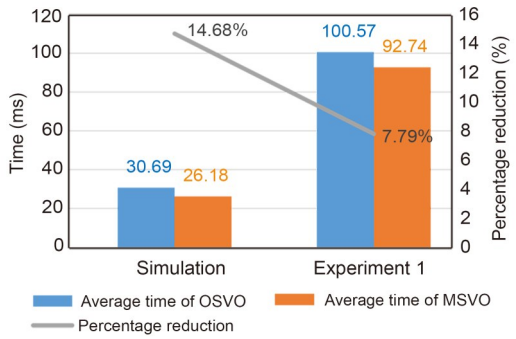

Fig. 20 Effect of PACR inverse kinematics

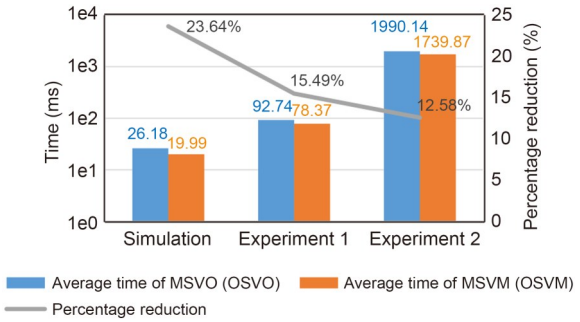

Fig. 21 Effect of FSPS

5 Conclusions

In this paper, a distributed variable density path planning method named DVDP-AC is proposed. In our proposed method, the inverse kinematic solution, collision detection, and post-processing are modified. First, collision detection based on distributed signed-distance-field (DSDF) and a single-step safety sphere (SSS) generally improves efficiency, especially in the scenarios where obstacles are densely distributed. In the ablation study cases, the computation time is reduced by 14.63% to 27.20% due to DSDF-based collision detection. Second, the position–attitude constraints reconstruction (PACR) inverse kinematics can further improve efficiency while meeting the requirements of the handling task. In the ablation study cases, the computational time is reduced by 7.79% to 14.68% due to PACR. Third, the adoption of forward sequential path simplification (FSPS) further improves the quality and efficiency of path simplification. In the ablation study cases, FSPS reduces the computational time by 12.58% to 23.64%. In baseline comparison, DVDP-AC shows advantages in efficiency and attitude constraints compared with existing mainstream algorithms.

The most important contribution of this study is to realize efficient path planning with terminal attitude constraints, which have important application value in industrial robot handling, stacking, spraying, and other tasks. The proposed method can be compiled into a planning tool and combined with three-dimensional modeling techniques to become an important part of offline programming software for industrial robots. We hope to implement these in future work.

Contributors

Jin WANG designed the research. Shengjie LI, Yichang FENG, and Peng WANG processed the data. Shengjie LI drafted the paper. Jin WANG and Guodong LU helped organize the paper. Shengjie LI, Haiyun ZHANG, and Jituo LI revised and finalized the paper.

Compliance with ethics guidelines

Jin WANG, Shengjie LI, Haiyun ZHANG, Guodong LU, Yichang FENG, Peng WANG, and Jituo LI declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abele E, Haehn F, Pischan M, et al., 2016. Time optimal path planning for industrial robots using STL data files. *Proc CIRP*, 55:6-11. https://doi.org/10.1016/j.procir.2016.08.038

Adeli H, Tabrizi MHN, Mazloomian A, et al., 2011. Path planning for mobile robots using iterative artificial potential field method. *Int J Comput Sci Iss*, 8(4):28-32.

Ademovic A, Lacevic B, 2014. Path planning for robotic manipulators via bubbles of free configuration space: evolutionary approach. Proc 22nd Mediterranean Conf on Control and Automation, p.1323-1328. https://doi.org/10.1109/MED.2014.6961559

Baziyad M, Saad M, Fareh R, et al., 2021. Addressing real-time demands for robotic path planning systems: a routing protocol approach. *IEEE Access*, 9:38132-38143. https://doi.org/10.1109/ACCESS.2021.3058121

Dijkstra EW, 1959. A note on two problems in connexion with graphs. *Numer Math*, 1(1):269-271. https://doi.org/10.1007/BF01386390

Ferguson D, Stentz A, 2006. Using interpolation to improve path planning: the field D* algorithm. *J Field Robot*, 23(2): 79-101. https://doi.org/10.1002/rob.20109

Fu B, Chen L, Zhou YT, et al., 2018. An improved A* algorithm for the industrial robot path planning with high success rate and short length. *Robot Auton Syst*, 106:26-37. https://doi.org/10.1016/j.robot.2018.04.007

Gottschalk S, Lin MC, Manocha D, 1996. OBBtree: a hierarchical structure for rapid interference detection. Proc 23rd Annual Conf on Computer Graphics and Interactive Techniques, p.171-180. https://doi.org/10.1145/237170.237244

Han D, Nie H, Chen JB, et al., 2018. Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection. *Robot Comput-Integr Manuf*, 49:98-104. https://doi.org/10.1016/j.rcim.2017.05.013

Harik GR, Lobo FG, Goldberg DE, 1999. The compact genetic algorithm. *IEEE Trans Evol Comput*, 3(4):287-297. https://doi.org/10.1109/4235.797971

Hart PE, Nilsson NJ, Raphael B, 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern*, 4(2):100-107. https://doi.org/10.1109/TSSC.1968.300136

Hernández C, Baier JA, Asín R, 2014. Making A* run faster than D*-lite for path-planning in partially known terrain. Proc 24th Int Conf on Automated Planning and Scheduling, p.504-508.

Huo XJ, Liu YW, Jiang L, et al., 2014. Inverse kinematic optimizations of 7R humanoid arms based on a joint parameterization. IEEE Int Conf on Mechatronics and Automation, p.113-118. https://doi.org/10.1109/ICMA.2014.6885681

Janson L, Schmerling E, Clark A, et al., 2015. Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions. *Int J Robot Res*, 34(7):883-921. https://doi.org/10.1177/0278364915577958

Kalakrishnan M, Chitta S, Theodorou E, et al., 2011. STOMP: stochastic trajectory optimization for motion planning. IEEE Int Conf on Robotics and Automation, p.9-13. https://doi.org/10.1109/ICRA.2011.5980280

Klingensmith M, Dryanovski I, Srinivasa S, et al., 2015. CHISEL: real time large scale 3D reconstruction onboard a mobile device using spatially-hashed signed distance fields. Proc Robotics: Science and Systems, Article 11. https://doi.org/10.15607/RSS.2015.XI.040

Koenig S, Likhachev M, 2005. Fast replanning for navigation in unknown terrain. *IEEE Trans Robot*, 21(3):354-363. https://doi.org/10.1109/TRO.2004.838026

Koenig S, Likhachev M, Furcy D, 2004. Lifelong planning A*. *Artif Intell*, 155(1-2):93-146. https://doi.org/10.1016/j.artint.2003.12.001

Kuffner JJ, LaValle SM, 2000. RRT-connect: an efficient approach to single-query path planning. Proc IEEE Int Conf on Robotics and Automation, p.995-1001. https://doi.org/10.1109/robot.2000.844730

LaValle SM, 1998. Rapidly-Exploring Random Trees: a New Tool for Path Planning. Technical Report, TR98-11, Department of Computer Science, Iowa State University, Ames, USA.

Li SP, Wang ZJ, Zhang Q, et al., 2018. Solving inverse kinematics model for 7-DoF robot arms based on space vector. Int Conf on Control and Robots, p.1-5. https://doi.org/10.1109/ICCR.2018.8534498

Liu HS, Zhang Y, Zhu SQ, 2015. Novel inverse kinematic approaches for robot manipulators with Pieper-Criterion based geometry. *Int J Contr Autom Syst*, 13(5):1242-1250. https://doi.org/10.1007/s12555-013-0440-y

Liu YY, Xi JL, Bai HF, et al., 2021. A general robot inverse kinematics solution method based on improved PSO algorithm. *IEEE Access*, 9:32341-32350. https://doi.org/10.1109/ACCESS.2021.3059714

Persson SM, Sharf I, 2014. Sampling-based A* algorithm for robot path-planning. *Int J Robot Res*, 33(13):1683-1708. https://doi.org/10.1177/0278364914547786

Qureshi AH, Ayaz Y, 2016. Potential functions based sampling heuristic for optimal path planning. *Auton Robots*, 40(6):1079-1093.

Starek JA, Gomez JV, Schmerling E, et al., 2015. An asymptotically-optimal sampling-based algorithm for bi-directional motion planning. IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.2072-2078. https://doi.org/10.1109/IROS.2015.7353652

Sun XX, Yeoh W, Koenig S, 2010. Moving target D* lite. Proc 9th Int Conf on Autonomous Agents and Multiagent Systems, p.67-74.

Tan T, Weller R, Zachmann G, 2020. Compressed bounding volume hierarchies for collision detection & proximity query. https://doi.org/10.48550/arXiv.2012.05348

Xie YM, Zhou R, Yang YS, 2020. Improved distorted configuration space path planning and its application to robot manipulators. *Sensors*, 20(21):6060. https://doi.org/10.3390/s20216060

Xing YS, Liu XP, Xu SP, 2010. Efficient collision detection based on AABB trees and sort algorithm. 8th IEEE Int Conf on Control and Automation, p.328-332. https://doi.org/10.1109/ICCA.2010.5524093

Zucker M, Ratliff N, Dragan AD, et al., 2013. CHOMP: covariant Hamiltonian optimization for motion planning. *Int J Robot Res*, 32(9-10):1164-1193. https://doi.org/10.1177/0278364913488805

List of supplementary materials