Sensor-based skeleton modeling and MVEEs approach of the redundant manipulator for reaction motion

Lan Ye1,2, Genliang Xiong1,3, Hua Zhang1,3 and Cheng Zeng1

Abstract
With the wide application of redundant manipulators, sharing a working space with humans and dealing with uncertainty seems an inevitable problem, especially in the dynamic and unstructured domain. How to deal with obstacle avoidance is of particular importance that robots and humans/environments are safe interactions to fulfill the complex cooperating tasks. This paper aimed at solving the problem of multiple points avoidance for the reaction motion based on the skeleton algorithm in unstructured and dynamic environments. A method named “sensor-based skeleton modeling and MVEEs approach of the redundant manipulator for the reaction motion” is proposed. The extraction of skeleton information from image is obtained to calculate the distances of the multiple control points and establish the repulsion in this method. Afterward, the force Jacobian related to the priority weighting factors is calculated and then a reaction force with damping term is established, which is corresponding nominal torque commands. For the redundant manipulator, the joint angles are obtained through torque iteration instead of inverse kinematics to reduce calculation cost. Finally, the method was tested by a 7-DOF manipulator in the ROS framework. The obtained results indicate that the method in this method can realize dynamic obstacle avoidance and time cost reduction.

Keywords
Reactive collision avoidance, 7-DOF manipulator, skeleton algorithm, vision sensor, virtual end-effector

Date received: 30 July 2021; accepted: 23 December 2021

Handling Editor: Chenhui Liang

Introduction
Collaborative robots are finding many applications in fields such as manufacturing, health care, agricultural production, and social tasks. Compared to tightly structured environments, cooperation-type robots are made to interact closely with humans in collaborative contexts in an unknown and dynamic environment while robots have to detect and avoid obstacles. In this case, it is necessary for performing real-time reactive strategies to deal with all possible collisions instead of precisely following a predefined path. Some related researches include fixed control points,1 multiple-point control,2 modification of reaction paths.3

For high-dimensional configuration space (C-space) formed by the high degree of freedom (high-DOF)
robots, the problem of planning a feasible path in a dynamic environment is one of the most complex and time-consuming tasks in the field of robotics and it can cause easily a dimension explosion. Probabilistic and randomized approaches, such as Rapidly-Exploring Random Tree (RRT),4 Probabilistic Roadmap Method (PRM),5 have been used to resolve this problem of the dimensionality.6 When the changes in the environment are inevitable, the potential fields are commonly used to estimate the reactive motion intent of the robots modeled as particles.7 In particular, it has obvious advantages in dynamic obstacle avoidance8 and dynamic target tracking,9 however, these focused primarily on the mobile robots. When it comes to a redundant manipulator which is consists of a multiple-link structure with a certain volume and cannot be represented by a particle, the problem becomes challenging because there exist some intrinsic constraints such as local minima issues and fluctuations transferred to narrow passages.10 Many attempts have been performed to develop various ways to handle these limitations.11–15 Whereas these main studies are the target for collision avoidance at the end-effector of the manipulator from the perspective of kinematics. For the redundant manipulator composed of multiple-link, the collision of arbitrary points may occur in the process of planning, but it is seldom discussed.

For collision avoidance, it needs to regard the motion of the robot at the arbitrary point as a possible reactive movement, and the problem has been extensively described in the robotics literature. The complete framework and its implementation have been elaborated in Brock and Khatib,3 although the major effort remains on the computation of control points position and corresponding Jacobian matrix. Alejo et al.16 put forward a real-time reactive collision avoidance method for multiple aircraft which has been applied in industrial environments. From the control viewpoint, a more general reactive motion approach called the “replanning and deformation methods” has been described.17 While approaching an obstacle, the planner first attempts to apply deformation and the robot can continue to replan if only the path remains feasible. In Cui et al.,18 combining a reactive path-planned approach with the Dubins curves and the artificial potential field (APF) method solves the path planning of the robot in a dynamic environment.

For reactive collision avoidance in human-robot interaction (HRI), it is necessary to track the key parts of the objects while a control system can take corrective actions before collisions. One major way for the introduction in unstructured environments depends on sensors to compute the distances between possibly colliding parts. In this context, the reliability and stability of the sensor directly correlate with the safe interaction with robot and human/environment. The Point Cloud Library (PCL)19–21 is the most common approach for obstacles or environments modeling with the visual sensors. However, the main shortcoming is that this method cannot fully describe the obstacle information in the environment. In Wu et al.,22 a high-dimensional grid model was used to model the known static obstacles, and the potential field method was adopted to process the unmodeled dynamic obstacles for the manipulator trajectory planning. Flacco Fabrizio23 proposed a method based on depth space offline mapping to divide depth space into depth grids. These two methods require that the main camera must cover the entire robot workspace, which has low computing efficiency and requires parallel computing to accelerate. As a common characteristic, the idea of directly calculating the distance of the environment image was introduced in Kuhn and Henrich.24 To improve performance in the computational time cost, the authors25,26 proposed a new method to assess the distances between a general point in Cartesian space and an object perceived by the depth sensor, but it can only be applied to a single sensor. In Li et al.,27 presents an environment modeling method for the manipulator using the offline mapping between the depth image and OctoMap in a dynamic non-structured environment. However, when the distance evaluation is calculated, only the real-time obstacle avoidance research of the end-effector is regarded.

In this paper, a method of sensor-based skeleton modeling and multiple virtual end-effectors (MVEEs) approach of the redundant manipulator for the reaction motion is proposed. In this framework, the geometry and architecture of the manipulator and objects are obtained by the extraction of the skeleton based on the three-dimensional (3D) sensor. Then the possible multiple collision points and distances can be calculated based on the skeleton model. Finally, the priority weighting factors are defined according to the virtual repulsive force and the reaction force is generated, resulting in corresponding nominal torque commands. Compared to the previous research, the primary contributions are listed as follows: (1) A method to get the skeleton online based on 3D vision sensor is proposed. (2) The bounding box was constructed and the distance calculation of the manipulator’s multiple control points and obstacles based on the real-time detection of skeleton model, which dramatically reduces the computational complexity. (3) For the redundant manipulator composed of multiple links, the MVEEs approach based on the skeleton model is realized to avoid the dynamic obstacle and the joint angles are obtained through torque iteration instead of inverse kinematics to reduce calculation cost.
Vision-based skeleton and bounding box modeling

A policy that automatically selects the control points in the whole process of motion, based on sensor information, could make it efficient for the application of human-robot interaction. It is important to note, however, that these control points should be able to be calculated quickly and result in simple distance computation. In this section, the skeleton algorithm and the bounding box modeling for dynamic objects for fast multiple-point control, are described.

Depth space model of the object

Figure 1 shows four coordinate systems that exist in the process of obtaining image information by the 3D sensor. The origin of the camera frame, which is described by the world frame, is defined as the optical center. The image pixel frame is on the top-left side of the image, \((u, v)\) are the columns and rows of the pixel, respectively. The focal length \(f\) is the distance between the imaging plane’s center and the camera optical center.

The vision sensor is modeled as a typical pinhole camera which is constituted of two main parts: the intrinsic parameters matrix \(K\) denoted the projection of a Cartesian point on the image plane, and the extrinsic parameters matrix \(e\) represented the coordinate transformation between the world and the sensor frame. Considering each point has the pixels given by the RGB camera and the distance in the depth space which is expressed as \(P_d = (u, v, d_p)\), the Cartesian point \(P_c = (x_c, y_c, z_c)\) in the sensor frame is given by:

\[
\begin{align*}
x_c &= \frac{u - c_x}{f_s} z_c \\
y_c &= \frac{v - c_y}{f_s} z_c \\
z_c &= d_p
\end{align*}
\]  

Where \(s_x\) and \(s_y\) are the pixel dimensions, \(c_x\) and \(c_y\) correspond to the image center coordinates in pixels.

Thus, we can calculate point \(P = (x_W, y_W, z_W)\) in the world frame as follows:

\[
\begin{bmatrix}
x_w \\
y_w \\
z_w
\end{bmatrix} =
K_e \begin{bmatrix}
x_W \\
y_W \\
z_W
\end{bmatrix} =
\begin{bmatrix}
f_s & 0 & c_x & 0 \\
0 & f_s & c_y & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
R' \\
T'
\end{bmatrix}
\begin{bmatrix}
x_W \\
y_W \\
z_W
\end{bmatrix}
\]

Where \(R'\) and \(T'\) represent the rotation matrix and the translation vector.

Building the skeleton

In this section, the skeleton extraction is based on the idea of thinning, that is, remove some points from the original image, but keep the original shape, until the skeleton of the image is obtained. All the definitions are followed by the Zhang-Suen method. Supposed a 3 × 3 window is applied and each element of the pixel \((i, j)\) is related to its eight neighboring elements as shown in Figure 2.

As well known that a pixel value at the \(n\)th iteration rests on the new pixel and its eight adjacent pixels at the \((n-1)\)th iteration in image processing. The variables \(A(p_1)\) and \(B(p_1)\) are the 0–1 patterns numbers and
nonzero neighbors numbers of \( p_1 \), respectively. To preserve the skeleton pixels in the 8-connectivity sense, the two sub-iterations are described as follows.

In the first sub-step, remove the pixel \( p_1 \) from the digital pattern if conditions (a)–(d) are all satisfied:

(a) \( 2 \leq B(p_1) \leq 6 \); (b) \( A(p_1) = 1 \); (c) \( p_2 \neq p_4 \neq p_8 = 0 \);
(d) \( p_4 \neq p_6 \neq p_8 = 0 \).

In the second sub-step, the last two of the four conditions are: (c’) \( p_2 \neq p_4 \neq p_8 = 0 \); (d’) \( p_4 \neq p_6 \neq p_8 = 0 \). Meanwhile, to keep the remaining pixels away from each other, the pixel \( p_1 \) is deleted if \( p_2 + p_3 + p_8 + p_9 \geq 1 \). Thus, it can be simplified the whole volumetric analysis of the manipulator/obstacle by taking into account a skeleton of the structure (Figure 3).

**Constructing bounding box used the geometric primitive**

The characteristic of the objects in the workspace can be described as simple geometric primitives using the shapes of the general objects, such as cuboid, cylindrical, conical, etc. As shown in Figure 4, \( \{O_bX_bY_bZ_b\} \) denotes the world coordinate system, and the arbitrary point \( P \) relative to \( \{O_bX_bY_bZ_b\} \) are represented by \( (x, y, z) \). When the satellite component assembly is the main application scenario, the main types of geometric primitives using super-quadric surface functions depiction can be defined in 3D space and the expression to depict a single geometry primitive is represented as follows.

(a) For a cylindrical surface, the super-quadric surface function is represented by:

\[
S(x, y, z) = \left( \frac{x}{r} \right)^2 + \left( \frac{y}{r} \right)^2 + \left( \frac{z}{h} \right)^8 - 1
\]  

(b) For a cuboid surface

\[
S(x, y, z) = \left( \frac{x}{a} \right)^8 + \left( \frac{y}{b} \right)^8 + \left( \frac{z}{c} \right)^8 - 1
\]
Based on the established skeleton, the parameters such as $a$, $b$, $c$ can be obtained by:

$$
\begin{align*}
    p_{\text{min}} &= \left[ x_{\text{min}}, y_{\text{min}}, z_{\text{min}} \right] \\
    p_{\text{max}} &= \left[ x_{\text{max}}, y_{\text{max}}, z_{\text{max}} \right]
\end{align*}
$$

(5)

Then the center of the bounding box can be calculated by:

$$
c = \frac{p_{\text{min}} + p_{\text{max}}}{2}
$$

(6)

We define dimension vector $s$ as the vector from point $p_{\text{min}}$ to point $p_{\text{max}}$, which contains the information on the length, width, and height of the bounding box:

$$
s = p_{\text{max}} - p_{\text{min}}
$$

(7)

Combined with the above analysis, the eight vertex coordinates of the bounding box can be easily obtained on the basis of the established skeleton (see Figure 5(a)). In general, the Cartesian position of each joint, computed by the D-H table described in Table 2, can constitute the endpoints of the skeleton segments.

Aim at safety collision avoidance, a cylindrical bounding box with the skeleton as a central axis, to which add two hemispherical bounding boxes, was constructed for the manipulator. Figure 5(b) shows a segment of the bounding box, which line AB represents the central axis of the skeleton, and $r$ is the bounding box radius. It can be obtained that points A and B are the intersections or endpoints of the skeleton, and the radius $r$ can be directly approximated in the SolidWorks model or calculated by the projection method (see Figure 5(c)). In view of the method for the bounding box construction based on the skeleton, it greatly provides convenience for the calculation of distance in the next section.

**The corresponding distance calculation of the multiple control points**

**Distance calculation for a control point**

Based on the description in the previous section, the control endpoints of the skeleton segments can be obtained. For any point $P (x, y, z)$, we can judge whether it is on the obstacle surface according to the calculation of equations (3) and (4).

$$
\begin{align*}
    S(x, y, z) &< 0, \quad P \text{ is inside the surface} \\
    S(x, y, z) &= 0, \quad P \text{ is on the surface} \\
    S(x, y, z) &> 0, \quad P \text{ is outside the surface}
\end{align*}
$$

(8)

As shown in Figure 6, this paper takes the case of a cuboid object as the research description. The value $S(x, y, z)$ from equation (8) is proportional to the proximity of point $P$ to the obstacle, that is the larger $|S(x, y, z)|$, the farther away $P$ is from the surface. Thus, as commonly-used distance metric is the Euclidean distance, $S(x, y, z)$ called the pseudo distance can be used to assess the proximity.

$$
\tilde{d}_p(P) = S(x, y, z)
$$

(9)

**Distance calculation for skeleton segments**

As well known that a segment is the most likely to collide at the minimum distance, which should also be considered as a moving control point on the basis of...
endpoints of segments. If all the points on the segment which is denoted as link \( S_1S_2 \) as an example are regarded, The calculation of the real minimum value is generally very complex. In this paper, we can first judge whether the control endpoints \( S_1, S_2 \) are external according to equation (8). The distances of the control points \( S_1, S_2 \) as shown in Figure 7(b), can be approximately calculated according to equation (9), which is represented by \( d_t(P_{S_1}), d_t(P_{S_2}) \). Then when the segment \( S_1S_2 \) is located on the same plane, obviously the distance between \( S_1 \) or \( S_2 \) is the minimum distance point. However, another case is that the segment \( S_1S_2 \) spans the two planes, as shown in Figure 7(a). The minimum distance of the segment is not at the endpoint, but a point in the middle of the segment. Thus, we can get the plane in which each endpoint is located based on the angle between the bounding box normal vector \( \overrightarrow{n} \) (red solid line) and the normal vector \( \overrightarrow{n}_0 \) (red dashed line) composed of the endpoint vector \( \overrightarrow{v}_1, \overrightarrow{v}_2 \) and the plane segment \( \overrightarrow{v}_1, \overrightarrow{v}_2 \). In this way, we judge that the control endpoints \( S_1, S_2 \) are related to the intersecting line segment of the two adjacent planes such as \( (v_1, v_2, v_3, v_7) \) and \( (v_{1a}, v_1, v_2, v_3) \), with the minimum distance between the segment \( \overrightarrow{v}_1, \overrightarrow{v}_2 \) and segment \( S_1S_2 \) (Figure 7(c)).

Let \( \overrightarrow{P_a} \) and \( \overrightarrow{P_b} \) represent the generic point position vectors in the two segments, whose extreme points are \( v_1, v_3, \) and \( S_1, S_2 \), respectively.

\[
\overrightarrow{P_a} = \overrightarrow{v}_1 + t_1 \overrightarrow{u}_1 \\
\overrightarrow{P_b} = \overrightarrow{s}_2 + t_2 \overrightarrow{u}_2
\]  

Where \( \overrightarrow{u}_1 \) and \( \overrightarrow{u}_2 \) are calculated by:

\[
\overrightarrow{u}_1 = \frac{\overrightarrow{v}_3 - \overrightarrow{v}_1}{\|\overrightarrow{v}_3 - \overrightarrow{v}_1\|} \tag{12}
\]

\[
\overrightarrow{u}_2 = \frac{\overrightarrow{s}_2 - \overrightarrow{s}_1}{\|\overrightarrow{s}_2 - \overrightarrow{s}_1\|} \tag{13}
\]

and \( \{t_1, t_2\} \) is scalar values, with \( t_1 / \|\overrightarrow{v}_3 - \overrightarrow{v}_1\| \in (0, 1), t_2 / \|\overrightarrow{s}_2 - \overrightarrow{s}_1\| \in (0, 1) \).

When the distance is minimal, the points on the two segments are denoted the collision points \( p_{ac} \) and \( p_{bc} \), respectively. Thus, the common normal line can be calculated as:

\[
\overrightarrow{P_{ac}P_{bc}} = \left\| \overrightarrow{s}_1 - \overrightarrow{v}_1 + t_2 \overrightarrow{u}_2 - t_1 \overrightarrow{u}_1 \right\| \tag{14}
\]

\[
\overrightarrow{P_{ac}P_{bc}} \times \overrightarrow{u}_1 = 0 \tag{15}
\]

\[
\overrightarrow{P_{ac}P_{bc}} \times \overrightarrow{u}_2 = 0 \tag{16}
\]

When \( t_1 \) and \( t_2 \) are both determined, the points \( p_{ac} \) and \( p_{bc} \) generating minimum distances within the two segments can be obtained. Thus, this analytical approach is adopted to compute the distance of the points \( p_{ac} \) and \( p_{bc} \).

\[
d_M = \left\| \overrightarrow{P_{ac}P_{bc}} \right\| \tag{17}
\]

Thus, the minimum distance of a segment is obtained as:

\[
\tilde{d}_{\text{link}}(S_1S_2) = \min (\tilde{d}_p(P_{S_1}), \tilde{d}_p(P_{S_2}), d_M) \tag{18}
\]

Figure 7. Schematic diagram of human-robot interaction: (a) the segment \( S_1S_2 \) spans the two planes, (b) the distances of the control endpoints \( S_1, S_2 \) and (c) the minimum distance between the two segments.
The distances of all segments can be calculated based on the equation (18), thus the minimum distance between the robot and the obstacle can be described by:

\[ d_{\text{arm}} = \min(d_{\text{link}1}, d_{\text{link}2}, \ldots, d_{\text{link}n}) \]  

(19)

When there is no collision at the minimum distance, the manipulator can safely achieve collision avoidance. In the next section, details on MVEEs approach are given.

The MVEEs approach of the redundant manipulator for reactive motion

Figure 8 shows the system flow chart of collision avoidance. After constructing the bounding box based on the skeleton algorithm to calculate the distance of the multiple control points in real-time, the multiple points of each segment are equivalent to MVEEs. Then the priority weighting factors are defined according to the virtual repulsive force value and the force Jacobian matrix is obtained. Next, the velocity mapping of the joint angle to the minimum distance control point is realized by Jacobian and the reaction force, combined with repulsive force and reverse resistance (damping term) induced by velocity under the regulation of the parameter \( D_a \), is established. Based on the above analysis, the torque generated by a reaction force to collide avoidance is derived, then the actual trajectory can be calculated by between kinematics and dynamic of the manipulator.

Generating reaction force in MVEEs

Given the framework of Figure 8, the reaction force to avoid collision consists of two parts: the repulsive force and the reverse resistance. To compute the repulsive field intensity, we calculate the distance between the multiple points of the robot skeleton and the object from the 3D sensor. The principle of constructing a repulsion field is: the robot and the obstacle are mutually repulsive and do not touch each other; meanwhile the repulsive force should gradually decrease as soon as the robot is away from the object. The safety index (defined as \( d_{\text{safe}} \)) is derived from the obstacle and the manipulator bounding box within which the repulsive field is activated. The repulsive force function can be expressed as:

\[ U_{\text{rep}}(q) = \begin{cases} 0 & d_i(q) > d_{\text{safe}} \\ \frac{1}{2} \eta_i \left( \frac{1}{d_i(q)} - \frac{1}{d_{\text{safe}}} \right)^2 \frac{d_i(q)}{1 + \frac{d_i(q)}{\sigma}} & d_i(q) \leq d_{\text{safe}} \end{cases} \]  

(20)

Where \( \eta_i \) is a positive weight coefficient, \( d_i(q) = \| p_i(q) - p_i(q_{\text{obstacle}}) \| \), which represents the nearest distance between the \( i \)th point and the obstacle. The distance index \( d_{\text{safe}} = \frac{L \times d_i(q)}{d_j(q)} \) namely the ratio of the distance from the control point to the goal and that to the obstacle. When the robot is far from the obstacle, that is, \( d_{\text{safe}} \gg 1 \), the expression \( \frac{d_i(q)}{1 + \frac{d_i(q)}{\sigma}} \) is equivalent to approximately 1 and the repulsive field is equivalent to amplification. When the robot is ready for the objective, that is \( d_{\text{safe}} \ll 1 \), the expression \( \frac{d_i(q)}{1 + \frac{d_i(q)}{\sigma}} \approx d_i(q) \). That is to say, the repulsive field is weaker and \( d_{\text{safe}} \) plays a critical role in the weakening of the repulsive field.

Figure 9 shows the repulsive force consisted of radial force \( f_{\text{rep1}} \) and tangential force \( f_{\text{rep2}} \), as be given the role of \( d_{\text{safe}} \). The repulsive force of any control point is given by:

\[ f_{\text{rep}}(q) = \begin{cases} 0 & d_i(q) > d_{\text{safe}} \\ f_{\text{rep1}n_{\text{cr}}} + f_{\text{rep2}n_{\text{rg}}} & d_i(q) \leq d_{\text{safe}} \end{cases} \]  

(21)
Where the unit normal vector $n_{ac}$ indicates the direction from the obstacle to the robot control point and $n_{rg}$ denotes the direction of the manipulator to the objective. The $f_{rep1}$ component is represented by the negative repulsive field gradient as follows:

$$f_{rep1} = \eta_i \left( \frac{1}{d_i(q)} - \frac{1}{d_{safe}} \right) \frac{d^n g}{d^2 l(q)} \frac{d^n g}{d^2 g}$$  \hspace{1cm} (22)$$

In fact, the tangential repulsive force from obstacle $f_{rep2}$ is equivalent to attraction in the direction of the goal, represented by taking the gradient of $d_g$ as follows:

$$f_{rep2} = \frac{1}{2} \eta_i \left( \frac{1}{d_i(q)} - \frac{1}{d_{safe}} \right)^2 \frac{d^n g}{d^2 g}$$  \hspace{1cm} (23)$$

The control point $p_{ac}$, which defined the minimum distance of the manipulator, is obtained according to equation (19). When we ensure that the minimum distance place does not collide, then the manipulator can safely achieve collision avoidance. Thus, the reaction force of adding the damping term is expressed as follows:

$$f_{ac} = f_{rep}(q) - D_q p_{ac}$$  \hspace{1cm} (24)$$

Where $f_{rep}(q)$ is the repulsion of the control point $p_{ac}$, $\dot{p}_{ac}$ is the velocity of the control point $p_{ac}$, and $D_q$ is a suitable tuning parameter based on fuzzy rules in which the input variables are respectively the velocity $\dot{p}_{ac}$ and the repulsive force $f_{rep}(q)$. $\dot{p}_{ac}$ has a differential mapping relationship with the responding the joint velocities $\dot{q}$ through the Jacobian matrix $J_{ac}$:

$$\dot{p}_{ac} = J_{ac} \dot{q} = J_s \begin{bmatrix} \dot{p}_{v1} \\ \dot{p}_{v3} \end{bmatrix} = J_s \begin{bmatrix} J_{v1} \\ J_{v3} \end{bmatrix} \dot{q}$$  \hspace{1cm} (25)$$

Where $J_{ac}$ depends on the two endpoints of the responding segment in which $p_{ac}$ is located, that is $v1, v3$. $J_{v1}, J_{v3}$ are the Jacobians of the two endpoints of the segment, respectively, and $\dot{q}$ from the force Jacobian based on priority weighting factors in the next section.

**Definition of force Jacobian matrix based on priority weighting factors in MVEEs**

As shown in Figure 10, the presence of the virtual repulsive force makes the robot offset, and the greater the force, the greater the offset of the corresponding control point. Thus, we can get the displacement change by the proportion of the force generated on each control point, that is, the priority value of each point is defined as follows:

$$\mathcal{W} = \begin{cases} \frac{|F_j|}{\text{const}} & \text{if } F_j \neq 0 \\ 0 & \text{if } F_j = 0 \end{cases}$$  \hspace{1cm} (26)$$

Where $n$ represents the control point numbers and $l$ is the displacement directions caused by corresponding points. $F_j$ is the $j$th elements of control point force $F$. From equation (26), it can be seen that the weighting
factors of each control point are changed when the force \( F_i \) are different. Thus, the priority function \( W \) is defined as:

\[
W = \text{diag}[W_1, ..., W_1, ..., W_n, ..., W_n] / C_{138}^2
\]

Where \( W_{ij} \) represents weighting factors element of \( i \)th control point and \( j \)th displacement direction. According to the next section, it is easy to reach the desired displacement when the weight factor value of the corresponding point is high.

Using the decided \( W \), the force Jacobian matrix \( J_F \), resulting from the virtual repulsive forces at the control points, can be expressed as follows:

\[
J_F = WJ
\]

Where \( J = [J^1, J^2, J^3, ..., J^m]^T \) is the concatenated Jacobian matrix. \( J^i \in R^l \times m \) is the Jacobian matrix of the \( CP_i \) and \( m \) is the degree of freedom of the manipulator. When the column of \( J_F \) has a full rank, there exists a left-inverse matrix and thus \( J_F^+ \), used to represent the pseudo-inverse of the Jacobian matrix \( J_F \), can be achieved by:

\[
J_F^+ = (J^TWJ)^{-1}J^TW
\]

Since the manipulator is redundant, the corresponding joint-angular vectors associated with the VEE position vector based on the differential kinematics mapping can be described as follows:

\[
\dot{q} = J_F^+ \dot{p}
\]

Based on the above analysis, the velocity mapping relationship between the control points of each section can be obtained. Thus, the velocity-related terms can be mapped to compute the force value of equation (24), which provides a basis for the torque calculation in the next section.

### Computing avoidance torques

As introduced, it is necessary to convert the desired trajectories to joint velocities or torques and the related torque is tracked in system dynamics. The avoidance torques generated by the reaction forces via the transpose of the Jacobian matrix defined in equation (25) are given:

\[
\tau_c = J_{ac}^T f_{ac}
\]

Hence, we compute joint angles through incremental iteration of the torque. Here we use a batch update rule:

\[
q(t + 1) = q(t) + \beta \tau_c
\]

Where \( \beta \) is an iterative step in joint space. The pseudo-code for the proposed approach in this paper is shown in the following Table 1.

**Simulation and results analysis**

To verify the effectiveness of the proposed method in this paper, the 7-DOF manipulator is applied in the scenario of satellite component assembly via Gazebo and ROS software. As shown in Figure 11(a), human-robot collaboration in the shared narrow space to complete the work of satellite component assembly meanwhile the collision is needed to avoid. Figure 11(b) shows the simulation environment implemented through the ROS software.

![Simulation validation scenario: (a) Satellite component assembly scene and (b) simulation structure used ROS and Gazebo.](image-url)
Table 2. D-H parameters of the manipulator.

| Frame | Link | \( \alpha_{i-1} \) | \( a_{i-1} \) (m) | \( d_i \) (m) | \( \theta_i \) (°) |
|-------|------|-----------------|-----------------|----------------|-----------------|
| 0–1   | 1    | 0               | 0               | 0              | \( \theta_1 \)  |
| 1–2   | 2    | 90              | 0               | 0              | \( \theta_2 \)  |
| 2–3   | 3    | -90             | 0               | 0.4495         | \( \theta_3 \)  |
| 3–4   | 4    | 90              | 30              | 0              | \( \theta_4 \)  |
| 4–5   | 5    | -90             | -30             | 0.4495         | \( \theta_5 \)  |
| 5–6   | 6    | 90              | 0               | 0              | \( \theta_6 \)  |
| 6–7   | 7    | -90             | 0               | 0              | \( \theta_7 \)  |

framework for this task. The detailed D-H parameters of the manipulator are presented in Table 2.

The simulation procedure is described as follows: When a human (obstacle) is shared with the robot in the workspace, the Kinect sensor is used for the 3D image acquisition and the skeleton is extracted according to the Gazebo virtual environment. We set the coordinates of the target point to be \((0, -0.65, 0.45)\) m. The initial position of the manipulator is also the minimum (Figure 12(a)). Thus, a control point \( CP_2 \) occurs at approximately 28 s and the maximum repulsive force of the obstacle \( Z \) direction can be obtained in the upper graph of Figure 12(a). It can be seen that the distance fluctuated. After 30 s, the reaction force is 0, indicating that the manipulator has been far away from the obstacle, so it follows the original path. It can be seen that the distance approaches zero and the manipulator reaches the goal as the gradual movement to the target point.

Discussion

As it can be seen in the simulation results section, the MVEEs approach can achieve obstacle avoidance for a manipulator with higher degrees of freedom. In view of the time cost and the superiority considerations, the graph-based basic method \( A^* \), the classical method artificial potential field (APF) for multi-DOF manipulator, and the optimization-based approach ant colony APF (AC-APF) are selected to be compared with the method MVEEs in this paper. Figure 12(g) shows the distance from the end-effector of the robot to the target point. It could be shown that the manipulator kept getting close to the goal in the first 10 s. The dynamic obstacle appeared after that time and considering the influence of reaction force, the distance from the target fluctuated. After 30 s, the reaction force is 0, indicating that the manipulator has been far away from the obstacle, so it follows the original path. It can be seen that the distance approaches zero and the manipulator reaches the goal as the gradual movement to the target point.

As shown in Figure 12(d) and (e), the priority weighting factors \( W_{Cp1}, W_{Cp2}, W_{Cp3} \) of each control point in the \( Y \) and \( Z \) direction are generated, respectively. The weighting factors are changed with the proportion of each repulsive force according to equation (21).

As shown in Figure 12(f), there are two graphs. The inset graph represents the corresponding parameter \( D_a \) based on the fuzzy inference system (FIS) is variable according to the velocity of the minimum distance control point and the responding repulsive force, which is described as estimation of interactive intention in our previous study. The fuzzy field parameters are \((-0.015, 0.015), (-5, 5), (10, 20)\), respectively. Thus the reaction forces in the \( Y \) and \( Z \) direction can be obtained by equation (24).

Figure 12(g) shows the distance from the end-effector of the robot to the target point. It could be shown that the manipulator kept getting close to the goal in the first 10 s. The dynamic obstacle appeared after that time and considering the influence of reaction force, the distance from the target fluctuated. After 30 s, the reaction force is 0, indicating that the manipulator has been far away from the obstacle, so it follows the original path. It can be seen that the distance approaches zero and the manipulator reaches the goal as the gradual movement to the target point.
Figure 12. Simulation results of 7-DOF redundant robot: (a) Minimum distance of segments, (b) force of control points in the $Y$ direction, (c) force of control points in the $Z$ direction, (d) priority weighting factor $W$ of control points in the $Y$ direction, (e) priority weighting factor $W$ of control points in the $Z$ direction, (f) reaction force in the $Y$ and $Z$ direction, (g) distance to goal, and (h) comparison test of some obstacle avoidance methods.
with the complexity and size of the environment and results in the computing time and storage capacity increasing significantly.

The collision-avoidance path for manipulator using APF shows the effective simulation results in a dynamic environment and avoids obstacles in real-time. It only takes 7.4804 s in the APF method and the distance between the end-effector is approximately 148.7 cm. However, the multi-DOF manipulator is trapped in local minima, especially in a narrow passage, due to the attraction to the desired goal configuration and the repulsion to the configuration space obstacles almost equal. Aim to this problem, some optimization-based approaches are applicable to APF. The AC_APF algorithm to improve the planning of manipulator end-effector is adopted for obstacle avoidance planning. The computing time (121.6748 s) decreased by half when compared to method A* and the distance is almost 0.81 cm.

In our work, the collision-avoidance method MVEEs for the manipulator has been carried out using deterministic methodologies, and therefore the defects of probabilistic methods can be avoided. It can be seen that two methods, which take respectively 121.6748 and 245.8535 s, are at the expense of computing time compared to our method which takes 52.4186 s.

To authors’ knowledge, there is no document in the literature supporting the method MVEEs to plan multiple control points collision-avoidance trajectory for the manipulator. Meanwhile, there are fewer parameters to be adjusted only the damping term parameter $D_s$, which is based on the interactive intention to guarantee deterministic collision avoidance. This may be a significant advantage as it opens the possibility of finding a collision-avoidance path, similar to the use of the resistance behavior of the damping term for the higher-DOF manipulator. Therefore, we ensure that our present work is worth research for the manipulator in the field of reaction motion.

The limitation of the proposed method and future research should be focused on the following aspects:

(i) The method of obstacle bounding box modeling based on the geometric primitive is a convex polygon in this paper, more work on concave polygon and complex obstacles in this respect should be considered.

(ii) When the multi-control point obstacle avoidance, facing multi-dynamic obstacles sharing a narrow space with the robot, is inevitable, the compliant contact control of some parts is also a worthy exploring problem.

(iii) Extending the proposed method to modeling and simulation of nonlinear systems with greater degrees of freedom robot.

Conclusion

In summary, a method of MVEEs control based on a skeleton model with the 3D-sensor is proposed to solve the problem of multiple control points avoidance for the reaction motion. The proposed approach is used to get the multiple possible collision points and distances between the manipulator and obstacles in the unstructured dynamic environment. Then, it greatly reduces the time and calculation cost through the Jacobian matrix related to the priority weighting factors. Meanwhile, a new form of a reaction force, which is corresponding nominal torque commands, to realize the joint angles iteration instead of inverse kinematics. The simulation results mean that this method can be successfully applied for the manipulator’s obstacle avoidance from beginning to end. Consequently, the MVEEs proposed in this paper could be widely applied in the field of human-robot interactions and provide a way to implement the non-contact compliant obstacle avoidance. However, it is noted that the multi-control point obstacle avoidance, facing multi-dynamic obstacles sharing a narrow space with the robot, sometimes is inevitable. In the future, based on the MVEEs, the contact compliant control of some parts deserve consideration. Also, further practical experiments will be needed to verify the simulation effect in the next step.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.
Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was financially supported by the National Key Research and Development Program of China under Grant No. 2018YFB1305300, the National Natural Science Foundation of China under Grant No. 61763030, and the Science and technology project of Education Department of Jiangxi Province under Grant No. GJJ212501.

ORCID iDs
Lan Ye https://orcid.org/0000-0002-2043-8734
Genliang Xiong https://orcid.org/0000-0002-1303-6980

References
1. Seto F, Kosuge K, Suda R, et al. Self-collision avoidance motion control for human Robot cooperation system using RoBE. In: 2005 IEEE/RSJ international conference on intelligent robots and systems, Edmonton, AB, Canada, 2–6 August 2005. pp. 3143–3148.
2. De Santis A, Pierro P and Siciliano B. The multiple virtual end-effectors approach for human-robot interaction. In: Lemarcič J and Roth B (eds) Advances in robot kinematics. Dordrecht: Springer, 2006. pp. 133–144.
3. Brock O and Khatib O. Elastic strips: a framework for motion generation in human environments. Int J Rob Res 2002; 21: 1031–1052.
4. Xu JJ, Duindam V, Alterovitz R, et al. Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring random trees and backchaining. In: 4th IEEE conference on automation science and engineering Key Bridge Marriott, Washington, DC, USA, 23–26 August 2008. pp. 41–46.
5. Kavraki LE, Svestka P, Latombe JC, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans Robot Autom 1996; 12: 566–580.
6. Lindemann SR and LaValle SM. Current issues in sampling-based motion planning. In: 11th international symposium of robotics research, Siena, 2003.
7. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. Int J Rob Res 1986; 5: 90–98.
8. Tzafestas SG, Tzamtsi MP and Rigatos GG. Robust motion planning and control of mobile robots for collision avoidance in terrains with moving objects. Math Comput Simul 2002; 59: 279–292.
9. Ge SS and Cui YJ. Dynamic motion planning for mobile robots using potential field method. Auton Robots 2002; 13: 207–222.
10. Weerakoon T, Ishii K and Nassirraeli AAF. An artificial potential field based mobile robot navigation method to prevent from deadlock. J Artif Intell Soft Comput Res 2015; 5: 189–203.
11. Zhu Z, Jing S, Zhong J, et al. Obstacle avoidance path planning of space redundant manipulator based on a collision detection algorithm. J Northwest Polytech Univ 2020; 38: 183–190.
12. Badawy A. Dual-well potential field function for articulated manipulator trajectory planning. Alex Eng J 2016; 55: 1235–1241.
13. Huber L, Billard A and Slotine JJ. Avoidance of convex and concave obstacles with convergence ensured through contraction. IEEE Robot Autom Lett 2019; 4: 1462–1469.
14. Zhu J and Yang MY. Path planning of manipulator to avoid obstacle based on improved artificial potential field method. Comput Meas Control 2018; 26: 205–210.
15. Xie L and Liu S. Dynamic obstacle-avoiding motion planning for manipulator based on improved artificial potential filed. Control Theory Appl 2018; 35: 1239–1249.
16. Alejo D, Cobano JA, Heredia G, et al. A reactive method for collision avoidance in industrial environments. J Intell Robot Syst 2016; 84: 745–758.
17. Eiichi Y and Fumio K. reactive robot motion using path replanning and deformation. In: 2011 IEEE international conference on robotics and automation, Shanghai, China, 9–13 May 2011, pp. 5456–5462.
18. Cui P, Yan W and Wang Y. Reactive path planning approach for docking robots in unknown environment. J Adv Transport 2017; 2017: 1–11.
19. Satyavolu S, Bruder G, Willemsen P, et al. Analysis of IR-based virtual reality tracking using multiple Kinects. In: IEEE virtual reality workshops, Orange County, CA, USA, 4–8 March 2012. pp. 149–150.
20. Nicolai P, Raczkowsky J and Wörn H. A novel 3D camera based supervision system for safe human-robot interaction in the operating room. J Autom Control Eng 2015; 3: 410–417.
21. Ling C, Wei HX and Li ZY. Environment exploration and obstacle avoidance of 6-DOF robot arm based on robot operating system. Mech Eng Autom 2019; 3: 175–180.
22. Wu XJ, Tang J, Li Q, et al. Development of a configuration space motion planner for robot in dynamic environment. Robot Comput Integr Manuf 2009; 25: 13–31.
23. Fabrizio F and De Luca A. Real-time computation of distance to dynamic obstacles with multiple depth sensors. IEEE Robot Autom Lett 2017; 2: 56–63.
24. Kuhn S and Henrich D. Fast vision-based minimum distance determination between known and unknown objects. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, San Diego, CA, USA, 29 October–2 November 2007. pp. 2186–2191.
25. Fabrizio F, Kroeger T, De Luca A, et al. A depth space approach to human-robot collision avoidance. In: IEEE international conference on robotics and automation, Saint Paul, MN, USA, 14–18 May 2012. pp. 338–345.
26. Flacco F, Kroeger T, De Luca A, et al. A depth space approach for evaluating distance to objects. J Intell Robot Syst 2015; 80: 7–22.
27. Li YL, Zhao YW, Wang Z, et al. Environment modelling method for manipulator robot based offline-mapping between depth image and OctoMap. J Control Decis 2020; 35: 1537–1546.
28. Zhang TY and Suen CY. A fast parallel algorithm for thinning digital patterns. Commun ACM 1984; 27: 236–239.
29. Mu Z, Liu T, Xu W, et al. A hybrid obstacle-avoidance method of spatial hyper-redundant manipulators for servicing in confined space. *Robotica* 2019; 37: 998–1019.

30. Ye L, Xiong G, Zeng C, et al. Trajectory tracking control of 7-DOF redundant robot based on estimation of intention in physical human-robot interaction. *Sci Prog* 2020; 103: 1–23.