International Journal of Control, Automation and Systems 15(2) (2017) 815-826
http://dx.doi.org/10.1007/s12555-015-0455-7
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Avoidance of Multiple Moving Obstacles during Active Debris Removal Using a Redundant Space Manipulator

Zonggao Mu, Wenfu Xu*, and Bin Liang

Abstract: During the operation of space manipulators for debris removal, the obstacles moving in the workspace must be avoided. We propose a unified modelling framework for multiple moving obstacles and a collision-free trajectory planning method for a redundant space manipulator. The complete properties of an obstacle, including its shape, dimension, pose (position and orientation), and velocity (linear and angular), are defined in the model. The obstacle surface is represented by a super quadratic function whose parameters are adjusted to describe different shapes and dimensions. Pseudo-distance is defined to evaluate the proximity extent between the manipulator and an obstacle. Considering multiple different obstacles, we present an approach to normalize the pseudo-distances. The self-motion of the redundant manipulator was used to optimize the normalized pseudo-distance by adaptive redundancy resolution. By ensuring that the pseudo-distance was always larger than the safety threshold value, collisions with the obstacles were avoided. The proposed method solved the problem for which the Euclidean distance was difficult, or even impossible, to calculate for 3-D cases. When handling multiple different obstacles, the proposed method was much easier and had higher computational efficiency than previous methods. The proposed method was verified by the simulation of typical missions.

Keywords: Active debris removal, multiple moving obstacles, obstacle avoidance, redundant manipulator, space robot.

1. INTRODUCTION

The growing amount of space debris poses a threat to operational spacecraft and the long-term sustainability of activities in outer space. Presently, many organisations and scholars are studying active debris removal technology [1–4]. A redundant space manipulator [5] has great advantages in singularity handling, obstacle avoidance, torque optimisation, and manipulability enhancement. The well-known manipulators launched to the international space station (ISS), including the Dexter manipulator , the European Robotic Arm (ERA), and the Space Station Remote Manipulator System (SSRMS, also called Canadarm2) [6], are all 7-DOF serial manipulators. Some free-flying space robots for future on-orbit servicing, such as FREND and TECSAS/DEOS, are also redundant. These space manipulators perform complex and diverse tasks such as target capturing, replacement of orbital replaceable units (ORU), and payload transfer. To ensure safety, the potential danger of collisions with nearby obstacles or with the arm itself must be avoided.

Obstacle avoidance approaches can be divided into off-line and on-line methods [7, 8]. Glass *et al*. [9] proposed a real-time collision avoidance scheme for redundant manipulators, which was implemented at the inverse kinematics level using the damped-least-squares formulation of the configuration control approach. Yoshida *et al*. [10] proposed a practical planning framework for generating 3-D collision-free motions in complex environments that was based on an iterative two-stage planning scheme. Perdereau *et al*. [11] proposed a method to represent obstacles using super-quadratic surface functions, which handled the collision avoidance problem by iteratively solving the inverse geometric model.

Obstacle modelling is the key to real-time control algorithms. To reduce computational load and improve efficiency, simple geometric primitives are generally used to represent a manipulator and its working environment. Colbaugh *et al*. [12] dealt with the case of a planar manipulator by representing the objects using circles sur-

Manuscript received December 5, 2015; revised April 4, 2016; accepted May 17, 2016. Recommended by Associate Editor Juhoon Back under the direction of Editor Myo Taeg Lim. This work was supported in part by the National Natural Science Foundation of China (61573116, U1613227), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China51521003), and the Basic Research Program of Shenzhen (JCYJ20160427183553203; JCYJ20150529141408781).

Zonggao Mu and Wenfu Xu are with Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China (e-mails: muzonggaook@163.com, wfxu@hit.edu.cn). They are also with the State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China. Bin Liang is with Department of Automation, School of Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: bliang@mail.tsinghua.edu.cn).
* Corresponding author.

rounded by a surface of influence and modelling the links by straight lines. Rahmanian-Shahri and Trocha [13] presented an on-line collision recognition method in which every link of a redundant robot and every obstacle in the workspace were modelled as a boundary ellipse. For 3-D workspace, spheres, ellipsoids, convex polyhedrons, and cylinders are generally used as the geometric primitives. Bonner and Kelley [14] proposed the successive spherical approximation to represent 3-D objects, which facilitated the planning of collision-free paths. [15] represented manipulator links with spheres and cylinders, and objects with spheres. Choi and Kim [16] represented the obstacles and links of a robot using spheres and ellipsoids, respectively. Hwang and Ju [17] also modelled the objects in a workspace as ellipsoids to simplify the mathematical representation and reduce the computational complexity for collision detection. For some objects, such as rods, a cylinder is a more appropriate representation of the surface. Therefore, Patel *et al*. [7] proposed a compact method of cylinder-cylinder collision detection and distance calculations using the notion of dual vectors and angles. For mobile obstacle avoidance, Saramago and Junior [18] presented a methodology to plan an off-line optimal trajectory by adding penalty functions.

However, there is little literature dealing with the modelling of multiple moving obstacles and avoidance in the 3-D workspace for serial manipulators. For different obstacles, or the same obstacle with different poses (position and orientation) relative to the manipulator links, the computational equations of the Euclidean distance are different and cannot be formed in a unified manner. Therefore, it will be much more complex, or even impossible, to extend the previous methods to avoid multiple different moving obstacles. In this paper, we propose both a unified framework to model obstacles and a 3-D collision-free trajectory planning method for redundant space manipulators to avoid collisions with multiple moving obstacles. Unlike the previous methods, the normalised pseudo-distance is defined as the cost function for the redundancy resolution.

The remainder of this paper is organised as follows. Section 2 introduces the configuration of a redundant manipulator for on-orbit servicing and derives two types of kinematic equations for it. Section 3 analyses the shape characteristics of typical spacecraft devices and presents a general framework for modelling objects in a 3-D workspace. In Section 4, a collision-free trajectory planning method is proposed to avoid multiple moving obstacles through the optimisation of the objective function constructed by the normalised pseudo-distances. Section 5 develops a 3-D simulation system with which the simulation of typical on-orbital tasks is performed. The last section presents the summary and conclusion.

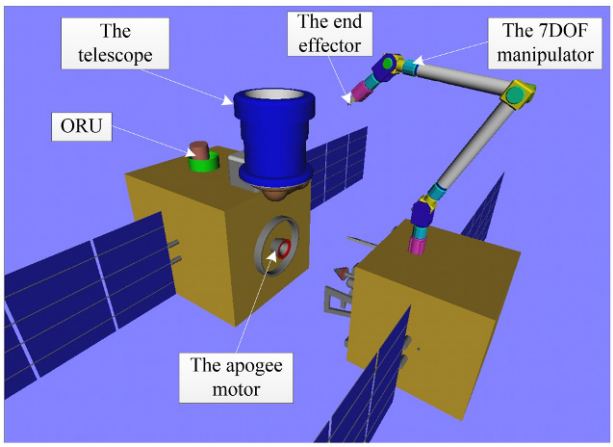


Fig. 1. System components.

Table 1. Redundant manipulator D-H parameters.

link i	θ_i (°)	α_i (°)	a_i (m)	d_i (m)
1	0	−90	0	$d_1 = 1.177$
2	0	90	0	0
3	0	−90	0	$d_3 = 2.93$
4	0	90	0	$d_4 = 0.55$
5	0	−90	0	$d_5 = 2.93$
6	0	90	0	0
7	0	0	0	$d_7 = 1.177$

2. MODELLING REDUNDANT SPACE MANIPULATORS

The designed space robot consists of a redundant manipulator and a spacecraft platform. It is used to implement the space tasks, such as visual monitoring, target capturing, and ORU replacement. The system components are shown in Fig. 1.The redundant manipulator is composed of seven revolute joints, which are arranged in an S-R-S configuration, i.e., the shoulder and wrist have three perpendicular joints forming spherical motion constraints and the elbow joint has a revolute joint. The D-H parameters are shown in Table 1.

For spacecraft-referenced end-point motion control [19], the motion of the space manipulator is controlled with respect to its own base. Its differential kinematic equation can then be written as:

$$\begin{bmatrix} {}^0\boldsymbol{v}_e \\ {}^0\boldsymbol{\omega}_e \end{bmatrix} = \left({}^0\boldsymbol{J}_m\right)\dot{\boldsymbol{\Theta}}, \tag{1}$$

where ${}^0\boldsymbol{v}_e$ and ${}^0\boldsymbol{\omega}_e$, respectively, denote the linear and angular velocities of the manipulator's end-effector, $\boldsymbol{\Theta} = [\theta_1, \theta_2, \cdots, \theta_7]$ is a vector formed by all joint angles of the manipulator, and ${}^0\boldsymbol{J}_m \in R^{6\times 7}$ is the Jacobian matrix, establishing the relationship between the joint rates and the end-effector velocities. The left-superscript '0' denotes

the base frame of the manipulator, i.e., frame $\{x_0, y_0, z_0\}$ is chosen as the reference frame. According to the definition of the Jacobian matrix, ${}^0\boldsymbol{J}_{\mathrm{m}}$ is independent on the mass parameters, i.e., the singularities of ${}^0\boldsymbol{J}_{\mathrm{m}}$ are kinematic singularities.

When the motion of the end-effector is described in the inertia frame, the differential kinematic equation of a space robotic system can be written as [19]:

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}_b \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \boldsymbol{J}_m \dot{\boldsymbol{\Theta}} \tag{2}$$

where $\boldsymbol{J}_b \in R^{6\times 6}$ and $\boldsymbol{J}_m \in R^{6\times 7}$ are the Jacobian matrices dependent on the base and the manipulator, respectively. When no external forces and torques are acting on the free-floating system, its linear momentum and angular momentum are conserved. With the assumption that their initial values are zero, the kinematic equation of a free-floating space robot is:

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}_g(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i)\dot{\boldsymbol{\Theta}} \tag{3}$$

where, $\boldsymbol{J}_g(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i) \in R^{6\times 7}$ is the generalised Jacobian matrix [20].

3. UNIFIED FRAME FOR MODELLING MULTIPLE MOVING OBSTACLES

3.1. Multiple moving obstacles environment

The working environment of a space manipulator with multiple moving obstacles is shown in Fig. 2, in which the obstacles are respectively denoted as Obstacle 1, Obstacle 2, ..., Obstacle N.

The ith obstacle (denoted as Obstacle i) moves with a linear velocity $\boldsymbol{v}_i$ and angular velocity $\boldsymbol{\omega}_i$. The manipulator is commanded to make its end-effector frame $\{O_eX_eY_eZ_e\}$ attain the target frame $\{O_tX_tY_tZ_t\}$ without colliding with any of the obstacles. To model the environment, the following properties of each obstacle should be considered:

(i) The geometries, including the (a) shape, which can be spherical, cylindrical, conical, combined, or others; and the (b) dimension, which is the geometric size of the obstacle;

(ii) The position and orientation with respect to the reference frame; and

(iii) The linear and angular velocities with respect to the reference frame.

3.2. Geometry modelling of typical devices on spacecraft

3.2.1 Shape characteristics of typical devices

It is well known that most spacecraft, such as the International Space Station, Hubble Space Telescope, and other artificial satellites, are composed of modular and standardised devices. Modularisation and standardisation are very important for long-life satellites and especially future serviceable spacecraft. The typical devices of a spacecraft have the following shape characteristics:

(i) The fuel tank's exterior contour is generally spherical;

(ii) The optical camera's shape comprises a cylinder and cube;

(iii) The star sensors' combined shape comprises a cone and cuboid;

(iv) The ORU's shape is cuboid;

(v) The communication antenna's shape is cylindrical;

(vi) The reaction thruster's its shape is conical;

(vii) Other devices' shapes combine one or more regular shapes.

After intensive investigation and analysis of space robots and their working environment, the characteristic of the objects in its workspace can be summarised as follows. The shape of a device can be described as one or several simple geometric primitives, such as cuboid (ORU devices), cylindrical (manipulator links, optical cameras, equipment brackets), conical (thruster nozzles, star trackers), and spherical or ellipsoid (fuel tanks). These 3-D geometric primitives can be defined to represent the shapes of general object, which are spheres or ellipsoids, cylinders or cones, and cuboids.

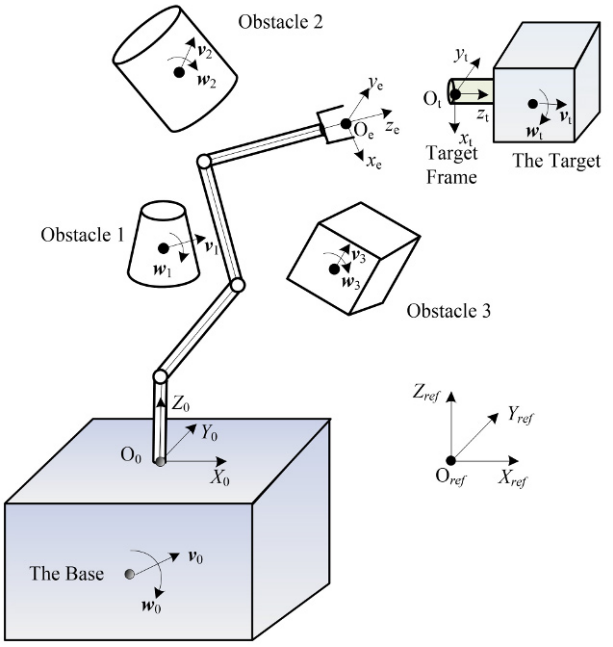
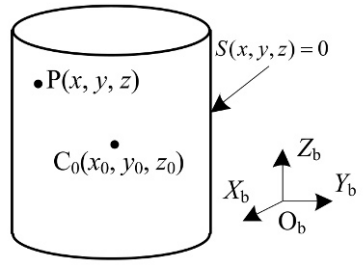


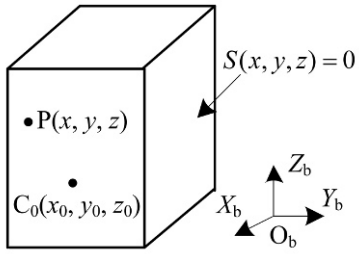
Fig. 2. Working environment with multiple obstacles.

3.2.2 Surface functions of the geometric primitives

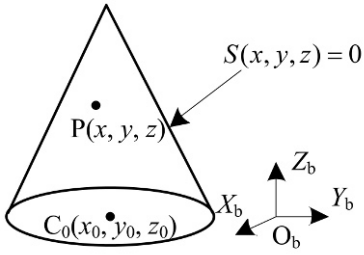
As the envelope of an obstacle can be represented by geometric primitives, the function to describe a single geometry primitive is defined first. Inspired by the work of



(a) Cylindrical object.



(b) Cuboid object.



(c) Conical object.

Fig. 3. Definition of the surface of an object with respect to its body-fixed frame.

[21], we use super-quadric surface functions to describe an object with an arbitrary geometry in 3-D space [22]. The definition of the surface of an object is shown in Fig. 3, in which $\{O_bX_bY_bZ_b\}$ denotes the body-fixed frame of the object and represent the coordinates of the geometry centre C_0.

If the coordinates of an arbitrary point P on the surface with respect to $\{O_bX_bY_bZ_b\}$ are denoted by (x,y,z), the surface function of an object has the following form:

$$^{b}S(x,y,z) = 0, \tag{4}$$

where the left-superscript 'b' denotes the body-fixed frame of the obstacle. The following super-quadric function can be used to define the surface of a geometric primitive:

$$^{b}S(x,y,z) = \left(\frac{(x-x_0)}{h_1}\right)^{2m} + \left(\frac{(y-y_0)}{h_2}\right)^{2n} + \left(\frac{(z-z_0)}{h_3}\right)^{2p} - 1. \tag{5}$$

In (5), h_1, h_2, and h_3 are the parameters determining the volume of the object and $m \geq 1$, $n \geq 1$, and $p \geq 1$ are the powers of the exponents defining the shape of the object.

If the geometry centre C_0 is chosen as the origin of the body-fixed frame, $x_0 = y_0 = z_0 = 0$, then equation (5) can be simplified as:

$$^{b}S(x,y,z) = \left(\frac{x}{h_1}\right)^{2m} + \left(\frac{y}{h_2}\right)^{2n} + \left(\frac{z}{h_3}\right)^{2p} - 1. \tag{6}$$

Using different values of these parameters, we can define various types of geometric primitives with different sizes.

(i) For a spherical surface

$$^{b}S(x,y,z) = \left(\frac{x}{R}\right)^{2} + \left(\frac{y}{R}\right)^{2} + \left(\frac{z}{R}\right)^{2} - 1 \tag{7}$$

(ii) For a cylindrical surface

$$^{b}S(x,y,z) = \left(\frac{x}{R}\right)^{2} + \left(\frac{y}{R}\right)^{2} + \left(\frac{z}{H}\right)^{8} - 1 \tag{8}$$

(iii) For a cuboid surface

$$^{b}S(x,y,z) = \left(\frac{x}{a}\right)^{8} + \left(\frac{y}{b}\right)^{8} + \left(\frac{z}{c}\right)^{8} - 1 \tag{9}$$

(iv) For a conical surface

$$^{b}S(x,y,z) = \left(\frac{x}{z+d}\right)^{2} + \left(\frac{y}{z+d}\right)^{2} + \left(\frac{z}{c}\right)^{8} - 1 \tag{10}$$

3.2.3 Envelope representation of general objects using combined geometric primitives

The shape of typical spacecraft equipment, which generally comprises one or more regular geometric primitives, approximately, can be represented using one or several geometric primitives. Taking the star sensor and the optical camera as examples, we can represent their envelopes using one or two geometric primitives:

(i) A single spherical envelope: The device is enveloped by a single spherical primitive. It is the simplest form, requires the least computing time, and is suitable for equipment with an approximately spherical shape. For the star sensor or the optical camera, however, two additional areas, which do not belong to the device, will be sacrificed.

(ii) A single cylinder envelope: The device is enveloped by a single cylinder primitive. It is suitable for equipment with an approximately cylindrical shape. The computational load is larger than that of a single spherical envelope, but the workspace sacrificed is smaller.

(iii) A single cuboid envelope: The device is enveloped by a single cuboid primitive. It is suitable for equipment with an approximately cuboid shape. The computational

load is larger than that of a cylinder envelope, but the workspace sacrificed is smaller.

(iv) An envelope with combined geometric primitives: The device is enveloped by a cuboid primitive and a cylinder primitive. The workspace sacrificed is the least, but the computational load is larger than that of the other shapes. In this case, the surface can be defined with the following functions:

$$ {}^{\mathrm{b}}S(x,y,z) = \begin{cases} \left(\frac{x-x_{01}}{a}\right)^8 + \left(\frac{y-y_{01}}{b}\right)^8 + \left(\frac{z-z_{01}}{c}\right)^8 - 1, \\ \qquad\qquad |z-z_{01}| \le c, \\ \left(\frac{x-x_{02}}{R}\right)^2 + \left(\frac{y-y_{02}}{R}\right)^2 + \left(\frac{z-z_{02}}{H}\right)^8 - 1, \\ \qquad\qquad c < z-z_{01} \le c+2H, \end{cases} \tag{11} $$

where $C_{01}(x_{01},y_{01},z_{01})$ and $C_{02}(x_{02},y_{02},z_{02})$ are the respective coordinates of the geometry centres of the cuboid and cylinder primitives, described in the body-fixed frame. For practical applications, two main factors, 'computational efficiency' and 'representation accuracy', are comprehensively considered to determine whether to use one, several, or even many geometric primitives to define the envelope of a device.

3.3. Defining pose and velocity for an obstacle

In the previous section, the surface of an object is described by a super-quadric function. Moreover, each obstacle can possibly move with respect to the base of the manipulator. Therefore, the multiple moving obstacles in the working environment should be described in a unified reference frame, such as the base frame of the space manipulator.

For the ith obstacle, whose body-fixed frame is denoted as $\{O_{bi}X_{bi}Y_{bi}Z_{bi}\}$, the position and orientation are

$$ {}^{ref}\boldsymbol{p}_{bi} = \begin{bmatrix} x_{bi} \\ y_{bi} \\ z_{bi} \end{bmatrix}, \quad {}^{ref}\boldsymbol{\psi}_{bi} = \begin{bmatrix} \alpha_{bi} \\ \beta_{bi} \\ \gamma_{bi} \end{bmatrix}, \tag{12} $$

where ${}^{ref}\boldsymbol{p}_{bi}$ is the origin position of $\{O_{bi}X_{bi}Y_{bi}Z_{bi}\}$ with respect to $\{O_{ref}X_{ref}Y_{ref}Z_{ref}\}$ and ${}^{ref}\boldsymbol{\psi}_{bi}$ represents the Euler angles of the orientation of $\{O_{bi}X_{bi}Y_{bi}Z_{bi}\}$. An alternative means to represent the pose is the homogeneous transformation matrix ${}^{ref}\boldsymbol{T}_{bi}$. The motion of each obstacle can be determined by the linear and angular velocities as follows:

$$ {}^{ref}\boldsymbol{v}_{bi} = {}^{ref}\dot{\boldsymbol{p}}_{bi}, \tag{13} $$

$$ {}^{ref}\boldsymbol{\omega}_{bi} = \boldsymbol{N}\left({}^{ref}\boldsymbol{\psi}_{bi}\right){}^{ref}\dot{\boldsymbol{\psi}}_{bi}, \tag{14} $$

where $\boldsymbol{N}\left({}^{ref}\boldsymbol{\psi}_{bi}\right)$ is the matrix mapping the time derivative of the Euler angles to the angular velocity. For the representation of the X-Y-Z Euler angles,

$$ \boldsymbol{N}(\boldsymbol{\Psi}) = \begin{bmatrix} 1 & 0 & s_\beta \\ 0 & c_\alpha & -s_\alpha c_\beta \\ 0 & s_\alpha & c_\alpha c_\beta \end{bmatrix}, \tag{15} $$

s_α, s_β and c_α, c_β are the sinusoidal/cosine of α and β.

4. TRAJIECTORY PLANNING TO AVOID MULTIPLE MOVING OBSTACLES

4.1. Pseudo-distance and normalized pseudo-distance

It is obvious that the Euclidean distance between the closest points, one belonging to the arm and the other to the obstacle, is the most intuitive criterion to evaluate the proximity extent between the manipulator and the obstacle. However, it is very difficult to express this in an analytical form.

4.1.1 Pseudo-distance of a point to an obstacle

For an arbitrary point P, whose coordinates are ${}^{b}\boldsymbol{P} = (x,y,z)$ with respect to the body-fixed frame of an obstacle, we can determine whether it lies on the surface of the obstacle by substituting (x,y,z) in the super-quadric surface function (5). The results are as follows:

$$ \begin{cases} S(x,y,z) < 0, & \text{if point P is inside the body} \\ S(x,y,z) = 0, & \text{if point P is on its surface} \\ S(x,y,z) > 0, & \text{if point P is outside the body.} \end{cases} \tag{16} $$

Taking a cylindrical object as an example, the above relationship is shown in Fig. 4.

It is very clear that the value of $S(x,y,z)$ is directly related to the proximity of point P to the obstacle. The larger the value of $|S(x,y,z)|$, the greater the distance from point P to the surface. From this point, $S(x,y,z)$ can be used to evaluate the proximity, similar to the Euclidean distance. Therefore, $S(x,y,z)$ is called the 'pseudo-distance' of point P. It is denoted by:

$$ \tilde{d}_p(\mathrm{P}) = S(x,y,z). \tag{17} $$

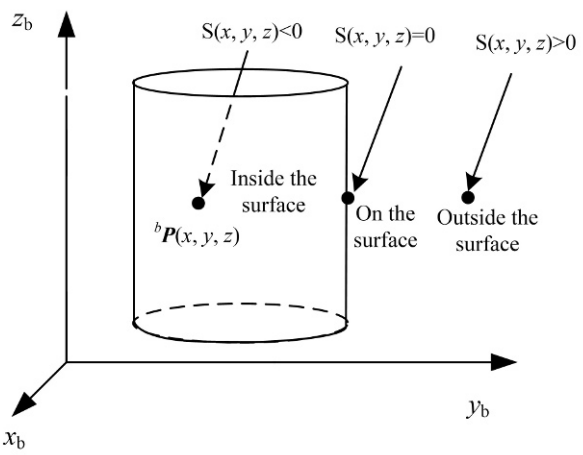


Fig. 4. The pseudo-distance concept.

4.1.2 Practical method to calculate the pseudo-distance of a link

The pseudo-distance from a link to an obstacle is defined as the minimum value of the pseudo-distance from any point on the link to its surface:

$$\tilde{d}_{link}(\mathrm{AB}) = \min_{\mathrm{P}\in\mathrm{AB}} \tilde{d}_p(\mathrm{P}), \tag{18}$$

where $\tilde{d}_{link}(\mathrm{AB})$ is the pseudo-distance from link AB to the given surface. It is generally very complex to analytically calculate the real minimum value by submitting the coordinates of all of the points on the link AB. We can simplify the calculation with the dichotomy method. After considering a sufficient safety margin, it is only required to calculate a few key points, such as the end points and the middle point.

Below is a practical method to calculate the pseudo-distance of a link whose end points and middle point of its central axis are respectively denoted as A, B, and M. If a security margin is used, the pseudo-distance of a link is approximately calculated as:

$$\tilde{d}_{link}(\mathrm{AB}) = \min\left(\tilde{d}_p({}^c\boldsymbol{P}_A),\ \tilde{d}_p({}^c\boldsymbol{P}_B),\ \tilde{d}_p({}^c\boldsymbol{P}_M)\right). \tag{19}$$

This practical method is illustrated in Fig. 5. If more points on line AB are used, the result will be more accurate. For practical applications, the readers can determine which points should be used according to the compromise requirement.

On the basis of the pseudo-distances between all of the links of the manipulator to the object, we can define the pseudo-distance between the manipulator and the object as:

$$\tilde{d}_{arm} = \min(\tilde{d}_{link1}, \tilde{d}_{link2}, \cdots, \tilde{d}_{linkn}), \tag{20}$$

where $\tilde{d}_{linki}$ is the pseudo-distance of the ith link.

4.1.3 Normalised pseudo-distance

When there are multiple obstacles, the pseudo-distance between the manipulator and the ith obstacle is denoted as ${}^{obs_i}\tilde{d}_{arm}$. If there are N obstacles, the 'pseudo-distances' between the manipulator and all of the obstacles, ${}^{obs_1}\tilde{d}_{arm}$, ${}^{obs_2}\tilde{d}_{arm}$, ..., ${}^{obs_N}\tilde{d}_{arm}$, can be determined sequentially. However, these pseudo-distances have different scales for different surface functions. To compare them, they must be scaled to the same level. This process is called normalisation (or weighting) and the corresponding pseudo-distance is called the normalised pseudo-distance (or weighted pseudo-distance). For the ith obstacle,

$${}^{obs_i}\hat{d}_{arm} = {}^{obs_i}\tilde{d}_{arm}/w_i, \tag{21}$$

where w_i is the scale factor of the ith obstacle (or so-called weighted coefficient) and ${}^{obs_i}\hat{d}_{arm}$ is the normalised pseudo-distance.

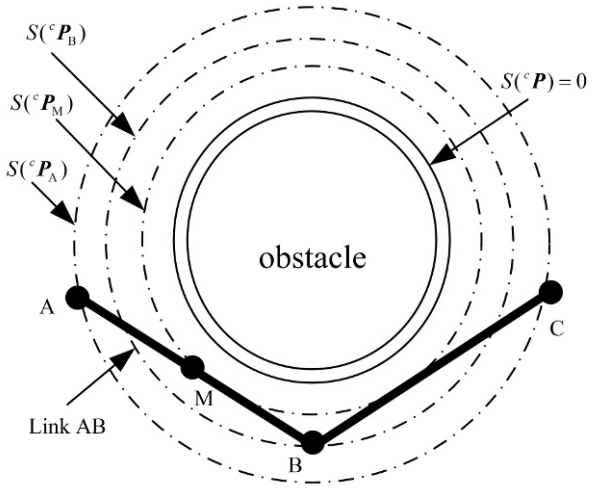


Fig. 5. A practical method to calculate the pseudo-distance of a link.

4.2. Optimized algorithm to avoid multiple moving obstacles

4.2.1 Objective function for avoidance of multiple moving obstacles

Observing the space manipulator studied in this paper, the following key points should be chosen to calculate the pseudo-distance:

(i) Point O: The origin of the base frame of the manipulator, i.e. $\{O_0X_0Y_0Z_0\}$;

(ii) Point S: The centre of the shoulder, i.e., the intersection point of the first joint axes (the first to third joints);

(iii) Points E_1, E_2: The elbow points of the manipulator. E1 is the intersection point of the third and fourth joint axes, and E2 is the intersection point of the fourth and fifth joint axes;

(iv) Point W: The centre of the wrist, i.e., the intersection point of the final joint axes (the fifth to seventh joints);

(v) Point T: The origin of the end-effector frame of the manipulator, $\{O_7X_7Y_7Z_7\}$; and

(vi) Points M_0, M_1, M_2, M_3: The middle points of lines OS, SE_1, E_2W, and WT, respectively.

These key points of the space manipulator are shown in Fig. 6.

The homogeneous coordinates of these key points with respect to the base frame can be determined according to the direct kinematic. The coordinates of the key points

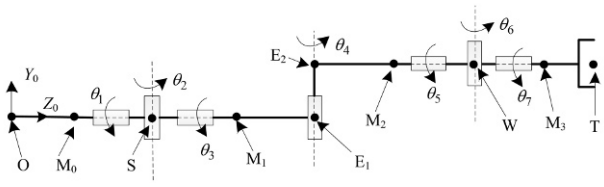


Fig. 6. The key points chosen on the manipulator.

with respect to the corresponding link frames are as follows:

$$ {}^{0}\bar{p}_{\mathrm{O}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, {}^{2}\bar{p}_{\mathrm{S}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, {}^{3}\bar{p}_{\mathrm{E_1}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (22) $$

$$ {}^{3}\bar{p}_{\mathrm{E_2}} = \begin{bmatrix} 0 \\ 0 \\ d_4 \\ 1 \end{bmatrix}, {}^{4}\bar{p}_{\mathrm{W}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, {}^{7}\bar{p}_{\mathrm{T}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (23) $$

The above vectors can be expressed in the base frame (used as the reference frame) by coordinate transformation:

$$ \begin{cases} {}^{0}\bar{\boldsymbol{p}}_{\mathrm{S}} = {}^{0}\boldsymbol{T}_{2}^{2}\bar{\boldsymbol{p}}_{\mathrm{S}}, \ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{E_1}} = {}^{0}\boldsymbol{T}_{3}^{3}\bar{\boldsymbol{p}}_{\mathrm{E_1}}, \ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{E_2}} = {}^{0}\boldsymbol{T}_{3}^{3}\bar{\boldsymbol{p}}_{\mathrm{E_2}} \\ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{W}} = {}^{0}\boldsymbol{T}_{5}^{4}\bar{\boldsymbol{p}}_{\mathrm{W}}, \ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{T}} = {}^{0}\boldsymbol{T}_{7}^{7}\bar{\boldsymbol{p}}_{\mathrm{T}}. \end{cases} \quad (24) $$

Correspondingly, the homogeneous coordinates of the middle points can be calculated as:

$$ \begin{cases} {}^{0}\bar{\boldsymbol{p}}_{\mathrm{M_0}} = \dfrac{{}^{0}\bar{\boldsymbol{p}}_{\mathrm{O}} + {}^{0}\bar{\boldsymbol{p}}_{\mathrm{S}}}{2}, \ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{M_1}} = \dfrac{{}^{0}\bar{\boldsymbol{p}}_{\mathrm{S}} + {}^{0}\bar{\boldsymbol{p}}_{\mathrm{E_1}}}{2} \\ {}^{0}\bar{\boldsymbol{p}}_{\mathrm{M_2}} = \dfrac{{}^{0}\bar{\boldsymbol{p}}_{\mathrm{E_2}} + {}^{0}\bar{\boldsymbol{p}}_{\mathrm{W}}}{2}, {}^{0}\bar{\boldsymbol{p}}_{\mathrm{M_3}} = \dfrac{{}^{0}\bar{\boldsymbol{p}}_{\mathrm{W}} + {}^{0}\bar{\boldsymbol{p}}_{\mathrm{T}}}{2}. \end{cases} \quad (25) $$

To calculate the pseudo-distance from each key point to the *i*th obstacle, all of the key points should be described in the body-fixed frame of the obstacle, $\{\mathrm{O}_{bi}\mathrm{X}_{bi}\mathrm{Y}_{bi}\mathrm{Z}_{bi}\}$. As the obstacle moves, we should predict the pose of the obstacle at the next sample time with respect to the base frame, according to the current pose ${}^{0}\boldsymbol{T}_{bi}(t)$, linear velocity ${}^{0}\boldsymbol{v}_{bi}$(t), and angular velocity ${}^{0}\boldsymbol{w}_{bi}$(t).

The position and attitude of the obstacle at the next sample time can be estimated as:

$$ {}^{0}\boldsymbol{p}_{bi}(t+\Delta t) = {}^{0}\boldsymbol{p}_{bi}(t) + {}^{0}\boldsymbol{v}_{bi}(t)\Delta t, \quad (26) $$

$$ {}^{0}\boldsymbol{\psi}_{bi}(t+\Delta t) = {}^{0}\boldsymbol{\psi}_{bi}(t) + {}^{0}\dot{\boldsymbol{\psi}}_{bi}(t)\Delta t, \quad (27) $$

where Δt is the sample period of the controller, and can be calculated using the following equation (see (14)):

$$ {}^{0}\dot{\boldsymbol{\psi}}_{bi}(t) = \boldsymbol{N}\left({}^{0}\boldsymbol{\psi}_{bi}(t)\right)^{-1}\left[{}^{0}\boldsymbol{\omega}_{bi}(t)\right]. \quad (28) $$

${}^{0}\boldsymbol{T}_{bi}(t+\Delta t)$ can be constructed according to ${}^{0}\boldsymbol{p}_{bi}(t+\Delta t)$and ${}^{0}\boldsymbol{\psi}_{bi}(t+\Delta t)$.

The coordinates of the key points with respect to the obstacle frame at the next sample time are then calculated as follows:

$$ {}^{bi}\bar{\boldsymbol{p}}_{x}(t+\Delta t) = \left({}^{0}\boldsymbol{T}_{bi}(t+\Delta t)\right)^{-1}\left[{}^{0}\bar{\boldsymbol{p}}_{x}\right]. \quad (29) $$

The subscript 'x' denotes the symbol of the key points, 'O', 'S', 'E_1', 'E_2', 'W', 'T', 'M_0', 'M_1', 'M_2', and 'M_3'.

Correspondingly, the pseudo-distance of each key point with respect to the *i*th obstacle is as follows:

$$ {}^{obs_i}\tilde{d}_x = S\left({}^{bi}\boldsymbol{p}_x(t+\Delta t)\right), \quad (30) $$

where $\boldsymbol{p}_x$ is the position vector obtained by eliminating the fourth element ('1') of the homogeneous coordinates $\bar{\boldsymbol{p}}_x$. The minimum value of these pseudo-distances is then defined as the pseudo-distance between the manipulator and the *i*th obstacle:

$$ {}^{obs_i}\tilde{d}_{arm} = \min({}^{obs_i}\tilde{d}_{\mathrm{O}}, {}^{obs_i}\tilde{d}_{\mathrm{S}}, \cdots, {}^{obs_i}\tilde{d}_{\mathrm{T}}) - {}^{obs_i}\tilde{d}_s. \quad (31) $$

In (31), ${}^{obs_i}\tilde{d}_{arm}$represents the pseudo-distance between the manipulator and the *i*th obstacle, and ${}^{obs_i}\tilde{d}_s$ is the security margin for the *i*th obstacle. If there are *N* obstacles, the pseudo-distances between the manipulator and all of the obstacles, ${}^{obs_1}\tilde{d}_{arm}$, ${}^{obs_2}\tilde{d}_{arm}$, ..., ${}^{obs_N}\tilde{d}_{arm}$, can be determined sequentially. Then, the objective function for avoiding all of the obstacles can be defined as the minimum value of all of the normalised pseudo-distances, i.e.:

$$ \hat{d}_{\min}(\boldsymbol{\Theta}) = \min\left({}^{obs_1}\hat{d}_{arm}, {}^{obs_2}\hat{d}_{arm}, \cdots, {}^{obs_N}\hat{d}_{arm}\right), \quad (32) $$

where $\hat{d}_{\min}(\boldsymbol{\Theta})$ is the minimum normalised pseudo-distance. $\hat{d}_{\min}(\boldsymbol{\Theta}) > 0$ shows that the manipulator does not collide with any obstacle. Therefore, the collision-free trajectory can be generated by maximising $\hat{d}_{\min}(\boldsymbol{\Theta})$ to ensure that $\hat{d}_{\min}(\boldsymbol{\Theta})$ is always greater than 0, using the manipulator redundancy.

4.2.2 Obstacle avoidance based on adaptive redundancy resolution

The primary task of the manipulator is to move its end-effector along the desired trajectories. Thanks to its redundancy, the arm can simultaneously accomplish a secondary task, which is taken here to be obstacle avoidance. According to the determination of the objective function (32) (the normalised pseudo-distance of the manipulator with respect to all of the obstacles), the following performance criterion function can be used to handle the collision avoidance problem:

$$ H(\boldsymbol{\Theta}) = \hat{d}_{\min}(\boldsymbol{\Theta}). \quad (33) $$

The performance criterion $H(\boldsymbol{\Theta})$ represents the function of all seven of the joints. The self-motion can be used to optimise the given performance criterion $H(\boldsymbol{\Theta})$. This is the so-called 'redundancy resolution'. For the gradient projection method, the redundancy is resolved as follows:

$$ \dot{\boldsymbol{\Theta}} = \boldsymbol{J}^{\#}\dot{\boldsymbol{x}}_e + k\left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right)\nabla H(\boldsymbol{\Theta}), \quad (34) $$

where $\boldsymbol{J}$ is the Jacobian matrix of the kinematic equations $\boldsymbol{J} = {}^{0}\boldsymbol{J}_m$ for (1) and $\boldsymbol{J} = \boldsymbol{J}_g$ for (3), and $\boldsymbol{J}^{\#}$ is the Moore-Penrose generalised inverse or pseudo-inverse of $\boldsymbol{J}$. Matrix $\boldsymbol{I}$ is an $n \times n$ identity matrix and $\left(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J}\right)$ is the null

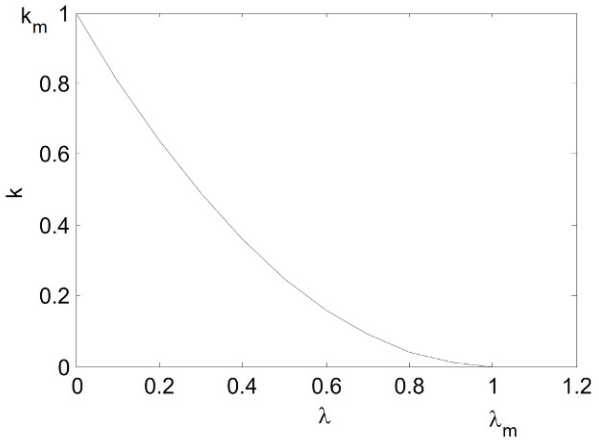


Fig. 7. The coefficient k vs the threshold value λ.

space projection. $\nabla H(\theta)$ is the gradient vector of $H(\boldsymbol{\Theta})$ and is defined as:

$$\nabla H(\boldsymbol{\Theta}) = \left[\frac{\partial H}{\partial \theta_1}, \frac{\partial H}{\partial \theta_2}, ..., \frac{\partial H}{\partial \theta_n}\right]^{\mathrm{T}}. \tag{35}$$

The gain k in (34) plays a great role in the gradient projection method. The selection of an unsuitable value will result in the obtained joint rates not achieving either the primary or the second task. Too large or too small values of k will excessively strengthen or weaken, respectively, the redundant motion (second task). If the distances between the manipulator and the obstacles are far enough, we think that it is not necessary to use self-motion to avoid the obstacles. However, the smaller the distance, the greater the risk of collision. Therefore, a threshold value $\lambda > 0$ is chosen to determine the safety domain and the value of k is adjusted adaptively, using the following equation:

$$k = \begin{cases} 0, & \hat{d}_{\min}(\boldsymbol{\Theta}) \geq \lambda \\ \dfrac{\left(\hat{d}_{\min}(\boldsymbol{\Theta}) - \lambda\right)^2}{\lambda^2} k_m, & 0 < \hat{d}_{\min}(\boldsymbol{\Theta}) < \lambda, \end{cases} \tag{36}$$

where $k_m > 0$ is the maximum value of k. According to (36), if $\hat{d}_{\min}(\boldsymbol{\Theta}) \geq \lambda$, then the obstacles are outside the safety domain and $k = 0$. When $\hat{d}_{\min}(\boldsymbol{\Theta}) < \lambda$, a collision will occur. The self-motion of the redundant manipulator should be used to avoid the collision, i.e., k should be greater than 0. The smaller the $\hat{d}_{\min}(\boldsymbol{\Theta})$, the greater is k. The relationship between k and λ is shown in Fig. 7. The two constants, k_m and λ, largely affect the avoidance performance. The adaptive redundancy resolution is realised by applying a gain value according to (36).

5. CASE STUDY

5.1. Avoidance of multiple moving obstacles

To verify the proposed approach, a 3-D simulation system is developed in the Microsoft Visual Studio C++ environment embedded OSG (Open Scene Graph, an open-source, high-performance 3-D graphics toolkit). To illustrate the applicability of the proposed obstacle model and trajectory planning method.

The manipulator is commanded to perform an ORU replacement task. Initially, the joint angles of the space manipulator are respectively as follows:

$$\boldsymbol{\Theta}_0 = [0^\circ,\ 66^\circ,\ 0^\circ,\ -84^\circ,\ 0^\circ,\ -50^\circ,\ 90^\circ]^{\mathrm{T}}. \tag{37}$$

Correspondingly, the position and attitude of the end-effector of the space manipulator are respectively as follows:

$$\boldsymbol{p}_{e0} = [-0.3887\ \mathrm{m},\ -0.2328\ \mathrm{m},\ 4.1072\ \mathrm{m}]^{\mathrm{T}}, \tag{38}$$

$$\boldsymbol{\Psi}_{e0} = [90.00^\circ,\ 0.00^\circ,\ 68.00^\circ]^{\mathrm{T}}, \tag{39}$$

where $\boldsymbol{p}_e$ is the position of the end-effector and $\boldsymbol{\Psi}_e$ represents the Z-Y-X Euler angles of the end-effector's attitude.

The initial pose of the target is

$$\boldsymbol{p}_{t0} = [3.94\ \mathrm{m},\ 1.62\ \mathrm{m},\ 5.96\ \mathrm{m}]^{\mathrm{T}}, \tag{40}$$

$$\boldsymbol{\Psi}_{t0} = [-10.00^\circ,\ 2.00^\circ,\ -15.00^\circ]^{\mathrm{T}}. \tag{41}$$

During the capturing mission, the target is assumed to be moving at the following velocities:

$$\begin{aligned} \boldsymbol{v}_t &= [-10.0,\ -2.0,\ -9.0]^{\mathrm{T}}\ \mathrm{mm/s}, \\ \boldsymbol{\omega}_t &= [0.01,\ 0.4,\ 0.3]^{\mathrm{T}}\ ^\circ/s. \end{aligned} \tag{42}$$

There are two moving obstacles in the workspace of the space manipulator. The geometry shapes and moving velocities of the two obstacles are show in Fig. 8. The

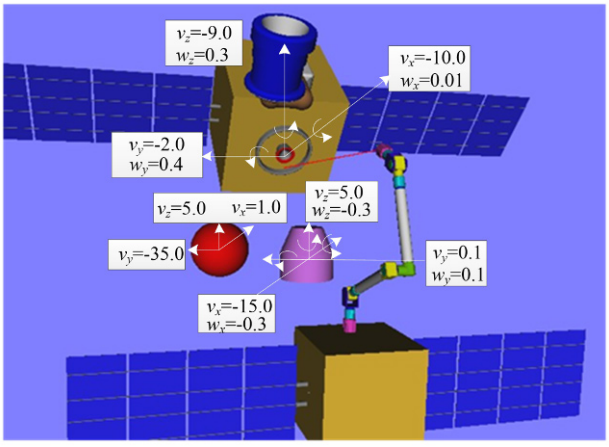


Fig. 8. The initial condition for multiple obstacle avoidance.

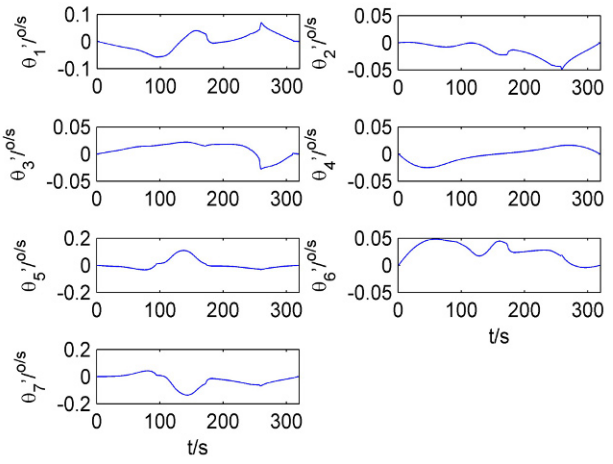


Fig. 9. Joint rates planned by the collision-free planning method.

workspace diameter is $r_0 = 1.0$ m. The initial position of its centre is $(0, 4.1, 2.6)$ m with respect to the coordinate system of link0 of the manipulator.

The linear velocities of the spherical obstacle are

$$\boldsymbol{v}_{obs1} = [1,\ -35,\ 5]^{\mathrm{T}}\ \mathrm{mm/s},$$
$$\boldsymbol{\omega}_{obs1} = [0.5,\ 0.5,\ 0.5]^{\mathrm{T}}\ {}^{\circ}/s. \tag{43}$$

The other obstacle, called the hybrid-shaped obstacle, is an obstacle comprising two geometric primitives, a cylinder and a truncated cone. Initially, the geometry centre of the cylinder is $(0.5, 1.3, 2.4)$ m with respect to the coordinate system of link0. The linear velocities of this obstacle are as follows:

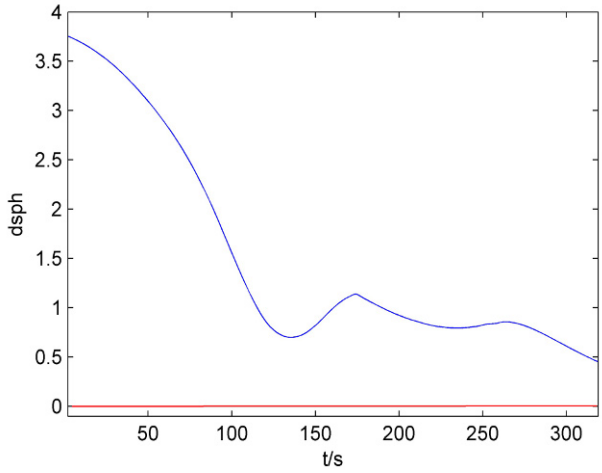
$$\boldsymbol{v}_{obs2} = [-15,\ 0.1,\ 5]^{\mathrm{T}}\ \mathrm{mm/s},$$
$$\boldsymbol{\omega}_{obs2} = [-0.3,\ 0.1,\ -0.3]^{\mathrm{T}}\ {}^{\circ}/s. \tag{44}$$

The collision-free methods are used to plan the trajectories of the space manipulator. The simulation studies is detailed in the following comments.

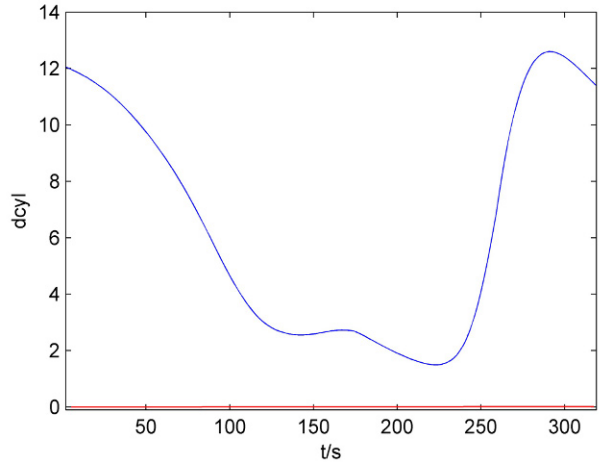
5.2. Collision-free trajectory planning with normal moving obstacles

The simulation conditions are the same as those used above, but the collision-free trajectory planning method is used to calculate the joint rates.

The simulation results are shown in Figs. 9-11. The planned joint rates are shown in Fig. 9 and the pseudo-distances between the manipulator and the two obstacles are shown in Fig. 10. The pseudo-distances are always greater than zero during the simulation, indicating that collision with the obstacles is effectively avoided. Comparing them with the corresponding states of the traditional planning method, we can see that the use of the collision-free trajectory planning method proposed here can avoid multiple moving obstacles with different shapes.



(a) Pseudo-distance to the spherical obstacle.



(b) Pseudo-distance to the hybrid-shaped obstacle.

Fig. 10. The pseudo-distance between the manipulator and two obstacles.

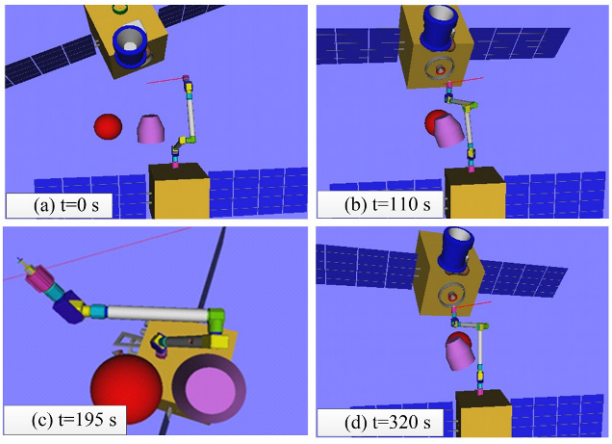


Fig. 11. Typical states during the simulation.

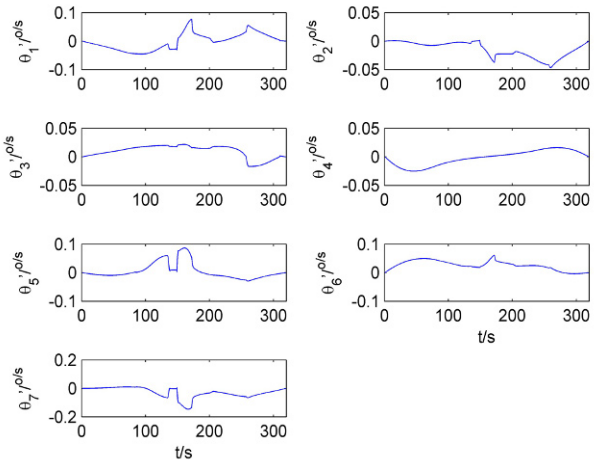


Fig. 12. Joint rates planned by the collision-free method.

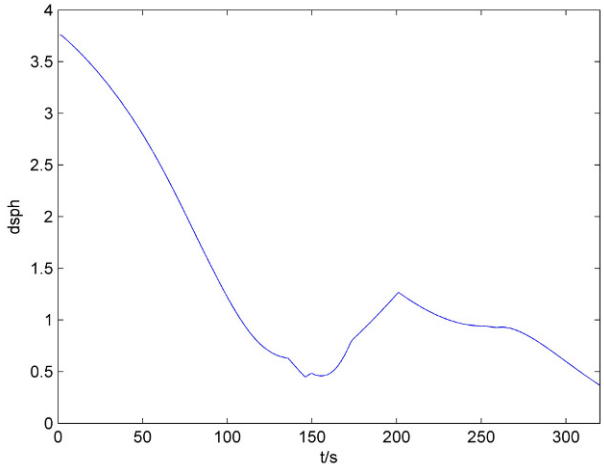
5.3. Collision-free trajectory planning with a oscillating obstacle

The simulation conditions are the same as those used above, but the sphere is supposed to be a oscillating obstacle. During 0 s to 110 s the oscillating obstacle moves to the manipulator; during 110s to 195s the oscillating obstacle moves away from the manipulator; during 190 s to 320 s the oscillating obstacle moves to the manipulator again. The collision-free trajectory planning method is used to calculate the joint rates.

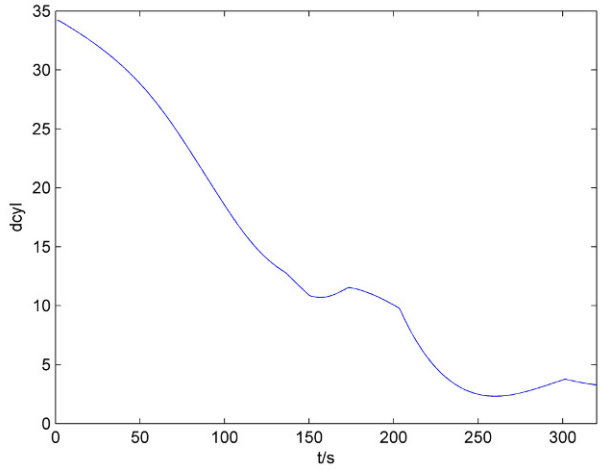
The simulation results are shown in Figs. 12-14. The planned joint rates are shown in Fig. 12 and the pseudo-distances between the manipulator and the two obstacles are shown in Fig. 13. The pseudo-distances are always greater than zero during the simulation, indicating that collision with the obstacles is effectively avoided. We can see that the use of the collision-free trajectory planning method proposed here can avoid multiple moving obstacles with a oscillating obstacle.

6. CONCLUSION

In this paper, we proposed a unified framework to model a general obstacle and a trajectory planning method for a redundant space manipulator to avoid multiple moving obstacles within its workspace. The geometries, pose, and velocities are included in the model. Because a spacecraft is generally composed of both modular and standard devices, the exterior contour of an obstacle can be approximately represented by one or several regular shapes, such as a cylinder, sphere, cone, or cube. A super-quadratic function is used to define these geometric primitives, and the pseudo-distance concept is extended to evaluate the proximity of the manipulator to an obstacle. For different types of obstacles, the pseudo-distances have different scales. Therefore, the pseudo-distances should be nor-



(a) Pseudo-distance to the oscillating obstacle.



(b) Pseudo-distance to the hybrid-shaped obstacle.

Fig. 13. The pseudo-distance between the manipulator and two obstacles.

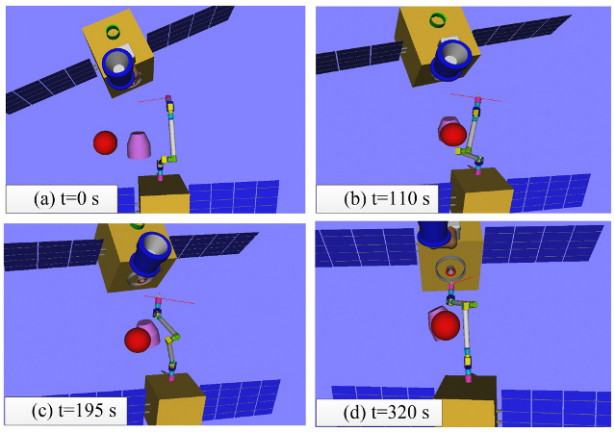


Fig. 14. Typical states during the simulation.

malised to the same level. Then, the objective function for avoiding multiple obstacles can be defined as the minimum value of all of the normalised pseudo-distances. A method based on gradient projection was then presented to plan the 3-D collision-free trajectory.

The proposed model describes an obstacle in a unified framework, which is easily realised as a structured type in certain programming languages. The trajectory planning method has high computational efficiency and good real-time performance because the complex computation of Euclidean distances is avoided. The work described here can be easily applied to other research fields such as mobile or industrial robots and spacecraft formation flying.

REFERENCES

[1] A. E. White and H. G. Lewis, "An adaptive strategy for active debris removal," *Advances in Space Research*, vol. 53, no. 8, pp.1195-1206, 2014. [click]

[2] Y.-K. Ma and H.-B. Ji, "Robust control for spacecraft rendezvous with disturbances and input saturation," *International Journal of Control, Automation, and Systems*, vol. 13, no. 2, pp.353-360, 2015. [click]

[3] D.-K. Wang, P.-F. Huang, J. Cai, and Z.-J. Meng, "Coordinated control of tethered space robot using mobile tether attachment point in approaching phase," *Advances in Space Research*, vol. 54, no. 6, pp.1077-1091, 2014.

[4] X. Xu and P. Huang, "Coordinated control method of space-tethered robot system for tracking optimal trajectory," *International Journal of Control, Automation, and Systems*, vol. 13, no. 1, pp.182-193, 2015. [click]

[5] J.-G. Wang, Y.-M. Li, and X.-H. Zhao, "Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm," *International Journal of Advanced Robotic Systems*, vol. 7, no. 4, pp.1-9, 2010.

[6] S. B. Nokleby, "Singularity analysis of the canadarm2," *Mechanism and Machine Theory*, vol. 42, no. 4, pp.442-454, 2007. [click]

[7] R. V. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: theory and experiments," *Journal of Robotic Systems*, vol. 22, no. 12, pp.737-757, 2005. [click]

[8] Y. Dai and S.-G. Lee, "Formation control of mobile robots with obstacle avoidance based on GOACM using onboard sensors," *International Journal of Control, Automation, and Systems*, vol. 12, no. 5, pp.1077-1089, 2014. [click]

[9] K. Glass, R. Colbaugh, D. Lim, and H. Seraji, "Real-time collision avoidance for redundant manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 7, pp.448-457, 1995.

[10] E. Yoshida, C. Esteves, I. Belousov, J.-P. Laumond, T. Sakaguchi, and K. Yokoi, "Planning 3D collision-free dynamic robotic motion through iterative reshaping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp.1186-1198, 2008.

[11] V. Perdereau, C. Passi, and M. Drouin, "Real-time control of redundant robotic manipulators for mobile obstacle avoidance," *Robotics and Autonomous Systems*, vol. 41, no. 1, pp.41-59, 2002.

[12] R. Colbaugh, H. Seraji and K. Glass, "Obstacle avoidance of redundant robots using configuration control," *Journal of Robotic Systems*, vol. 6, no. 6, pp.721-744, 1989. [click]

[13] N. Rahmanian-Shahria and I. Trocha, "Collision-avoidance control for redundant articulated robots," *Robotica*, vol. 13, no. 2, pp.159-168, 1995. [click]

[14] S. Bonner and R.-B. Kelley, "A novel representation for planning 3-D collision-free paths," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 6, pp.1337-1351, 1990.

[15] F. Shadpey, C. Tessier, R.-V. Patel, B. Langlois, and A. Robins, "A trajectory planning and object avoidance system for kinematically redundant manipulators: an experimental evaluation," *Proc. of AAS/AIAA American Astrodynamics Conference*, Halifax, NS, Canada, pp.979-997, 1995.

[16] S. I. Choi and B. K. Kim, "Obstacle avoidance control for redundant manipulators using collidability measure," *Robotica*, vol. 18, no. 2, pp.143-151, 2000. [click]

[17] K. Hwang and M.-Y. Ju, "3D collision-free motion based on collision index," *Journal of Intelligent and Robotic Systems*, vol. 33, no. 1, pp.45-60, 2002.

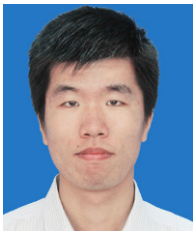
[18] S. R. F.-P. Saramago and V. S. E. Junior, "Optimal trajectory planning of robot manipulators in the presence of moving obstacles," *Mechanism and Machine Theory*, vol. 35, no. 8, pp.1079-1094, 2000.

[19] E. Papadopoulos and S. Dubowsky, "On the nature of control algorithms for free-floating space manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp.750-758, 1991.

[20] K. Yoshida, "Engineering test satellite VII flight experiments for space robot dynamics and control: theories on laboratory test beds ten years ago, now in orbit," *International Journal of Robotics Research*, vol. 22, no. 5, pp.321-335, 2003.

[21] R. V. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: theory and experiments," *Journal of Robotic Systems*, vol. 22, no. 12, pp.737-757, 2005.

[22] Z. Mu, W. Xu, X. Gao, L. Xue, and C. Li, "Obstacles modeling and collision detection of space robots for performing on-orbit services," *Proc. of 2014 4th IEEE International Conference on Information Science and Technology (ICIST),* Shenzhen, China, pp.461-466, 2014.



Zonggao Mu received the B.E. degree in Mechatronics engineering, from Shandong University of Science and Technology, Qingdao, Shandong, China in 2011, and the M.E. degree in Mechatronics engineering, from Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China in 2013. He is currently pursuing a Ph.D. degree in Mechanical Engineering. His research interests include space robotics, redundant manipulator.

Wenfu Xu received the B.E. degree in 2001, and the M.E. degree in 2003, both in control engineering, from Hefei University of Technology, Hefei, China, and the Ph.D. degrees in the control science and engineering from Harbin Institute of Technology, Harbin, China, in 2007. He was a research associate (as a visiting scholar, from Feb. 22, 2013 to Feb.21, 2014) with the Dept. Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China. He is currently a Professor with the Department of Mechanical and Automation Engineering, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. His research interests include space robotics, multi-body system dynamic, redundant manipulator.

Bin Liang received the B.S. and the M.S. degrees in control engineering from Northwestern Polytechnical University, Xi-An, China, in 1991 and 1994 respectively, and the Ph.D. degree in precision instrument and mechanology from Tsinghua University, Beijing, China, in 1994. He is currently a professor with the Department of Automation, Tsinghua University, Beijing, China. His research interests include space robotics, manipulators and intelligent control.