The method for predicting self-collisions of multi-link manipulators

VA Kramar

1 Sevastopol State University, 33, Universitetskaya st., Sevastopol, 299053, Russia

E-mail: kramary@mail.ru

Abstract. The article proposes a method for solving the problem of predicting the self-collision of multi-link manipulators with their agreed work. The method is based on the analysis of projections of manipulator links on coordinate planes. The proposed approach will make it possible to solve the problem simple and suitable for the online prediction mode of critical positions of manipulators, possible self-collision, with their coordinated work. The developed algorithm was tested when constructing the control of an anthropomorphic robot SAR-400.

1. Introduction

An important role in the control construction for multi-link manipulators of a robotic complex system plays the predict of self-collision of manipulators, i.e. collisions between manipulators.

Most of the work in this direction is aimed at finding restrictions on controls that exclude the movement of two manipulators provided that the paths of the manipulator links intersect by introducing restrictions on the position of the rotation angles of the manipulator links, and the ban on the manipulator's movements in specific directions.

It is known the approaches in which collision avoidance is achieved by planning the trajectory of the manipulator that received the move command and instructions to leave the working area of the first manipulator for the manipulator, which becomes an obstacle to the path. An important task is the construction of the trajectories of manipulators in real-time [1]. An algorithm was proposed in [2], according to which the structure of manipulators is divided into segments so that arbitrary points can be selected and controlled. This algorithm determines the collision points on such segments and generates suitable control for avoiding the collision. For its implementation, sensory information on the position of each link during movement is required in order to plan evasion trajectories to possible collisions.

In [3], it was proposed to use the “slave-master” approach for manipulators and assign priorities to the execution of commands. When the “leading” manipulator is located in the area of the interaction with the “slave” manipulators, the forbidden zone of the “slave” manipulators changes in accordance with the position of the “leading” manipulator. A continuation of this approach is the method of detecting self-collisions based on the discrete division of the working space into many small volumes [4]. Each of the discrete volumes has the status “busy” or “free”. When constructing the control, the situation is monitored when movement along the calculated trajectory will lead the manipulator to occupied discrete volumes of the working space.
These algorithms have a certain redundancy. In our opinion, when building an online manipulator control, it is necessary to carry out the predicting procedure for the possibility of the occurrence of self-collisions.

The article proposes a solution to the problem of predicting the critical positions to prevent collisions between two manipulators, based on an analysis of the projections of the manipulator links on the coordinate planes. The proposed approach is based on the relations from the course of linear algebra [5]. This approach allows making the algorithm simple and suitable for the online critical position prediction mode to prevent collisions of two manipulators during their operation.

2. Theoretical constructions

Let us consider the relative position of two cylinders modelling the corresponding links of the manipulators (Fig. 1).

![Figure 1. The manipulator's links do not intersect](image)

We construct the analytical expressions that determine the intersections of cylinders that in a real situation will mean a collision of the links of the manipulators of the robotic complex system.

Let us consider the arbitrary links of manipulators. In the initial state, we believe that the manipulators are in the Cartesian coordinate system.

To construct a collision prediction algorithm for the cylinders, we rotate the original coordinate system $XYZ$ around the $y$ axis by a certain angle so that the cylinder modelling the link of the first manipulator lies in the plane $OX'Y'$ of the new coordinate system $XYZ'$.

Find $\alpha$ – the angle between the vector $a_1$ describing the axis of the cylinder modeling the link of the manipulator (Fig. 2) and the plane $OXY$. Let the coordinates of the vector $a_1$ in the original basis have the form $(x_1, y_1, z_1)$ – the coordinates of the beginning of the vector $a_1$ and $- (x_2, y_2, z_2)$ the coordinates of the end of the vector $a_1$. For searching the angle $\alpha$, from the beginning of the vector $a_1$, we draw a vector $n_{xy}$ – normal to the plane $OXY$ with the coordinates of the beginning $(x_1, y_1, z_1)$ and the coordinates of the end $(x_1, y_1, z_2)$ (Fig. 2).
Figure 2. a) The schematic position of the vectors; b) The vector $a1$ with a start at point (1,1,1) and end at point (4,4,7) and vector $-\text{standard}$ in Matlab.

To calculate the desired angle, we will consider a vector $\hat{a}1$ with the following coordinates: $(x_1,0,z_1)$ – the coordinates of the beginning of the vector and $(x_2,0,z_2)$ - the coordinates of the end of the vector. The coordinates of the beginning of the vector – the “new” normal $\hat{n}_{xy}$ will have the form $(x_1,0,z_1)$, and the coordinates of the end of the vector $\hat{n}_{xy} - (x_1,0,z_2)$.

The desired angle $\alpha$ will be found from the formula

$$\alpha = 90^\circ - \gamma,$$
$$tg \gamma = \frac{\|\hat{a}1 \times \hat{n}_{xy}\|}{\langle \hat{a}1, \hat{n}_{xy} \rangle}. \tag{1}$$

In (1), the numerator denotes the norm operation of the vector product, and the denominator indicates the scalar product.

Next, we rotate the coordinate system $XYZ$, for example, around an axis $OY$ by an angle $\alpha$. At the same time, we move on to a new coordinate system, for example, $XYZ'$ where the axis of the manipulator lies on the plane $OXY'$. For this, we will use the rotation matrix [1]

$$M_Y = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}. \tag{2}$$

Similarly, this procedure can be formulated for rotation around an axis $OX$.

Now we can represent any vector on the basis of a new coordinate system. For example, for the vector $a1$, we have

$$a1' = a1 \cdot M_Y. \tag{3}$$
Now we will take into account that the cylinder modelling the link of the manipulator has a radius $R$ (Fig. 4).

![Figure 3](image)

**Figure 3.** a) The offset coordinate system; b) The offset coordinate system in Matlab.

Find the coordinates of the vectors, which are the projections of the tangents of the cylinder on the plane of the coordinate system $XYZ'$. Consider the projection vectors on the plane $OXY'$ (Fig. 5). The subscripts in the designation of the vectors indicate on what plane they are the projection.

![Figure 4](image)

**Figure 4.** Cylinder as part of the manipulator.

Applying the trigonometric relations for a right-angled triangle, we obtain the coordinates of the vectors. For a vector $a_{XY-R}'$ – the coordinates of the beginning of the vector will be equal $(x_1' - R \cdot \sin(\alpha-90^\circ), y_1' - R \cdot \cos(\alpha-90^\circ), 0)$, and the end of the vector – $(x_2' - R \cdot \sin(\alpha-90^\circ), y_2' - R \cdot \cos(\alpha-90^\circ), 0)$. For a vector $a_{XY+R}'$ – the coordinates of the beginning of the vector will be equal $(x_1' + R \cdot \sin(\alpha-90^\circ), y_1' + R \cdot \cos(\alpha-90^\circ), 0)$, and the end of the vector – $(x_2' + R \cdot \sin(\alpha-90^\circ), y_2' + R \cdot \cos(\alpha-90^\circ), 0)$. For the plane $YZ'$, the coordinates of the

![Figure 5](image)

**Figure 5.** a) Vectors on the plane $OXY'$; b) Vectors on the plane $OXY''$ in Matlab.
projection vectors will look like: \((0, y_1' - R \cdot \cos(\alpha - 90^\circ), R)\)- coordinates of the beginning of the vector \(a_1'_{YZ+R}\), 
\((0, y_2' + R \cdot \cos(\alpha - 90^\circ), R)\) – coordinates of the end of the vector \(a_1'_{YZ+R}\), 
\((0, y_1' - R \cdot \cos(\alpha - 90^\circ), -R)\) – coordinates of the beginning of the vector \(a_1'_{YZ-R}\), 
\((0, y_2' + R \cdot \cos(\alpha - 90^\circ), -R)\) – coordinates of the end of the vector \(a_1'_{YZ-R}\). For the plane \(XZ'\), the coordinates of the projection vectors will look like: \((x_1' + R \cdot \sin(\alpha - 90^\circ), 0, -R)\) – coordinates of the beginning of the vector \(a_1'_{XZ-R}\), 
\((x_2' - R \cdot \sin(\alpha - 90^\circ), 0, R)\) – coordinates of the end of the vector \(a_1'_{XZ-R}\), 
\((x_1' + R \cdot \sin(\alpha - 90^\circ), 0, R)\) – coordinates of the beginning of the vector \(a_1'_{XZ+R}\), 
\((x_2' - R \cdot \sin(\alpha - 90^\circ), 0, 0)\) – coordinates of the end of the vector \(a_1'_{XZ+R}\).

An example of projections for a cylinder that describes a manipulator link with an axis in the form of a vector with the beginning at the point \((1,1,1)\) and with the end at the point \((4,4,7)\) is shown in Figure 6.

**Figure 6.** Projections of the cylinder in space \(X'Y'Z'\)

Further, to analyze possible collisions of manipulators, we consider a cylinder modelling a link of the second manipulator. We represent a vector describing the axis of the cylinder in the coordinate system \(X'Y'Z'\) using the rotation matrix indicated in (3). Next, we find the projection of the vector modelling the axis of the second cylinder on the plane of the coordinate system \(X'Y'Z'\). Using the above approach, we find projections of vectors which are describing the boundaries of the cylinder describing the link of the second manipulator on the plane of the coordinate system \(X'Y'Z'\).

**3. The Self-collision algorithm.**

As noted earlier, the considered approach for predicting the self-collision of robot manipulators can be based on an analysis of the projections of manipulator links on coordinate planes. Based on the formulas in section 3, we formulate a rule for predicting self-collisions.

In order for the links of the manipulators not to intersect, it is necessary that at least on one plane, the projections of the links of the manipulator do not intersect (Fig. 1). For simplify the calculations, it is assumed that the projections of the ends of the links – ellipses – are approximated by rectangles.

Figure 7 shows an example of projection analysis for the vector of the first link with the coordinates of the beginning at the point \((1,1,1)\) and the end at the point \((4,4,7)\) and the second link with the coordinates of the beginning at the point \((5,6,4)\) and the end at point \((5,12,12)\).
In order for the links of the manipulators to collide, it is necessary that the projections of the links of the manipulators intersect on all planes. This procedure must be performed to check the collision of all links of the first manipulator with all links of the second manipulator.

In conclusion, we present a method for checking the intersection of segments on a plane [5]. The first segment has the coordinates \((x_1, y_1)\) and \((x_2, y_2)\), and the second segment \(- (x_3, y_3)\) and \((x_4, y_4)\).

Next, we determine the angular coefficients in the equations of lines:

\[
\begin{align*}
  k_1 &= \frac{y_2 - y_1}{x_2 - x_1}, \\
  k_2 &= \frac{y_4 - y_3}{x_4 - x_3}.
\end{align*}
\]

If \(k_1 = k_2\), then the lines are parallel and the segments cannot intersect. Otherwise, we determine the free terms in the equations of lines: \(b_1 = y_1 - k_1x_1\), \(b_2 = y_3 - k_2x_3\). If the lines have a point of intersection, then \(k_1x + b_1 = k_2x + b_2\). From the last formula, we calculate the intersection point of the lines. If it lies in a given interval, then the segments intersect.

4. Conclusion

The article presents a method for predicting the intersection of manipulator links when they perform the agreed work. The proposed method allows us to conclude whether the links of the manipulators intersect when they set a certain trajectory. Despite the apparent complexity of the above method, we note that all operations are simple and, therefore, the proposed method is convenient for using it as an online analysis of the manipulator’s self-collision when constructing a control. The proposed method for predicting the self-collision of manipulators will be effectively used in constructing control of the manipulator’s trajectories to prevent their collisions. The use of the modern approach to constructing control «Programming by demonstration» [6] will also require the application of the self-collision predict method developed in the article.

5. Acknowledgements

The reported study was funded by RFBR and the Government of Sevastopol, project number 20-41-925002.

References

[1]. Siciliano B, Khatib O 2016 Springer-Verlag Berlin Heidelberg - 2227
[2]. Santis A, Albu-Schäffer A, Ott C, Siciliano B Hirzinger G 2007 IEEE/ASME international conference on advanced intelligent mechatronics 9871732
[3]. JefferiesM, YeapW, 2008 NY: Springer-Verlag 202
[4]. Kabanov A, Tokarev D 2020 IOP Conference Series: Materials Science and Engineering, 709 044021.
[5]. Steven R 2005 Springer-Verlag New York 486
[6]. AlchakovV, KramarV, LarionenkoA, 2020 IOP Conf. Series: Materials Science and Engineering709 022092