JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 1363-1377 (2021)
DOI: 10.6688/JISE.202111_37(6).0010

A Collision Detection Method for Robot Manipulators Modeled by Ellipse

HAOBIN SHI[1], MENG LIANG[1], KAO-SHING HWANG[2] AND WEI-ZHENG WANG[2]
[1]*School of Computer Science*
Northwestern Polytechnical University
Xi'an, 710000 P.R. China
[2]*Department of Electrical Engineering*
National Sun Yat-sen University
Kaohsiung, 841 Taiwan
E-mail: shihaobin@nwpu.edu.cn; lmengzi@outlook.com;
hwang@g-mail.nsysu.edu.tw; carl_hwang@msn.com

The objective of this paper is aimed to estimate the actual shortest distance between links and obstacles which are represented by maximal Lowner-John (L-J) ellipses enclosed by the objects. But due to the elliptical inlay, there will be a case where the link is exposed outside the ellipse. This paper designs a method based on compensation method. Firstly, links are modeled by inner ellipses and related data tables and graphs are established. Then Gaussian function is used to obtain the compensated data graph about the shortest distance. Finally, the shortest distance of links and obstacles is estimated by interpolation method. Three experimental scenarios are designed in this paper, and compares inner ellipse method with circumscribed ellipse method to verify the effect of the proposed collision detection method.

Keywords: L-J ellipse, Gaussian compensation function, internal difference method, collision avoidance, robot manipulators

1. INTRODUCTION

In the field of robotics, acquiring the imminent distance between a robot arm and a restricted environment is the key to the collision-free motion [1]. In general, the link and related objects are represented by primitive geographical models for distance estimation such as with polyhedron modeling, the distance detection between polyhedrons can be hence easily derived. Some methods use simple polyhedron approximations to simplify the modeling procedure [2-4]. But using polyhedral models to describe complex shapes often requires many planes and edges [5-7]. The computational complexity limits the real time control of robot arms. Therefore, it is essential to use a simple model unlike polyhedral one for application [8, 9]. Later, Rimon [10] proposed a method to simplify the shape by using a minimum volume closed ellipsoid and to estimate distance effectively.

An ellipsoid with the smallest volume containing an object is called the L-J ellipsoid, and the corresponding one is L-J ellipse in the two dimensions [11]. For a particular object modeled by an L-J ellipse, its complex geometry is simplified into a flat graph and the calculation is a convex optimization process [12, 15]. In this paper, to simplify the process of generating the elliptic model of a link, the boundary polyhedron of links is defined. The vertex of the polyhedron is used as the input to generate a minimal area ellipse that can

Received October 31, 2019; revised May 7, 2020; accepted September 28, 2020.
Communicated by Jen-Hui Chuang.

cover the link. Suppose a basic L-J ellipse is $(x/a)^2 + (y/b)^2 = 1$, it can be transformed into matrix form $\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} 1/a^2 & 0 \\ 0 & 1/b^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1$. Equivalently, $x^T Cx = 1$. Where central coordinates is (0, 0) and characteristic matrix is C, and the major and minor axes of the ellipse follow the directions of the two eigenvectors of matrix C , and the two corresponding eigenvalues are the reciprocal of the square of the length of the semimajor axis and the semiminor axis respectively. If the radius of both axes is equal, the ellipse is a circle.

Suppose a common L-J ellipse $\varepsilon_y^d(y, Y)$ where d is dimension, y is central coordinates, and Y is characteristic matrix. Computing the shortest distance between two ellipses $\varepsilon_a^2(a, A)$ and $\varepsilon_b^2(b, B)$ is an optimization problem [10]. The circumscribed ellipse is represented by a three-dimensional feature matrix, which is beneficial to reduce the complexity and increase the speed of estimating distance, but the larger the ratio of the width to the length of a link, the larger the volume of an ellipse required, which results in estimated distance being more than actual distance.

This paper improves the accuracy of the distance estimation for the L-J ellipse modeling method, instead of using the circumscribed ellipse, but using the shortest distance of inner ellipses to estimate the actual distance of links. Although it is inevitable that part of a link falls outside the ellipse, it causes an error in the distance estimation. In order to solve the problem, a compensation function in the form of Gaussian function is adopted. Although the inner ellipse method is improved with the compensation mechanism and interpolation method, the stability and reliability of the method are unchanged.

Although it is convenient to use circumscribed ellipses to model links, it will cause overestimation and waste the working space of the manipulator. Using inner ellipse to calculate the distance between a link and an obstacle, combined with compensation mechanism, can not only solve the problem of underestimation, but also the problem of overestimation. Compared with the circumscribed ellipse method, the inner ellipse method makes robot arms closer to other robot arms or obstacles without collision and improves the utilization rate of working space. Using the circumscribed ellipse method to model links, it will cause overestimation and low accuracy of distance estimation. However, estimating distance with the proposed method is more accurate than the circumscribed ellipse method, and the experimental results demonstrate the effectiveness.

The remainder of this paper is organized as follows. In Section 2, the algorithm flow of the proposed fast collision detection method for robotic links molded by ellipses inner ellipse are introduced, including using the L-J ellipse to build a three-dimensional data table, constructing the compensation function of Gaussian function from data graph, and using linear interpolation to find the estimated distance. In Section 3, this paper builds an elliptical simulation environment and compares the inner method with the circumscribed ellipse to verify the proposed method. Finally, the conclusion is given in Section 4.

2. COLLISION DETECTION METHOD

This paper refers to the method of wrapping robotic links and obstacles with circumscribed ellipsoids, and estimating the shortest distance between a link and an obstacle by calculating the shortest distance between two ellipsoids [12-14]. The difference is that the distance is estimated and the data table is constructed by using L-J ellipses. Then links or obstacles are modeled by inner ellipses to estimate the shortest distance in two-dimensional space. In order to solve the distance error caused by the overestimation or underestimation about actual distance, a compensation function based on Gaussian function is used to reduce the estimated distance error. Finally, according to the data table and graph generated after compensation, the linear interpolation method is used to estimate the shortest distance in the whole motion.

Underestimation mentioned above means that when links collide, there is still a segment distance between two ellipses. Overestimation means that when two ellipses collide, the distance of the link is still far.

2.1 Create Data Table Based on the L-J Ellipse

The data of Robot Manipulators is shown in Table 1. The L-J ellipse designed according to Table 1 is shown in Fig. 1, where L is the actual distance of the two links, E is the shortest distance of the two ellipses. Fig. 1 shows the underestimation. In order to reduce the distance error, two variables $e1$, $e2$ are designed. $e1$ represents the distance from the vertex of the link A (red wire rectangle box) to its inner ellipse. $e2$ represents the distance from the vertex of the link B (red wire rectangle box) to its inner ellipse. They are expected to satisfy the Eq. (1).

$$E - e1 - e2 \cong L \tag{1}$$

The $E - e1 - e2$ is used to estimate the actual distance, and $L - (E - e1 - e2)$ is used to calculate the distance error. Links's data is shown in Table.1, and it is assumed that distance of the two ellipses are separated by 50 cm. Then the influence of each angle on the estimated distance is calculated and verified, that is, whether it can satisfy Eq. (1), when the angle between two ellipses is different. As is shown in Fig. 2 (a), ellipse A remains stationary, ellipse B revolves around A, while ellipse B rotates by itself. Due to the symmetry of revolution, the maximum revolution angle is 180°. Finally, the result about the influence mentioned above is integrated into a 60 × 60 data graph, as shown in Fig. 2 (b), where 180° of revolution is equally divided into 60 parts, 360° of revolution is equally divided into 60 parts.

Table 1. The links data of the robot manipulators.

Link	Length (cm)	Width (cm)	Distance between the center points of two links (cm)
A	40	20	50
B	30	20	

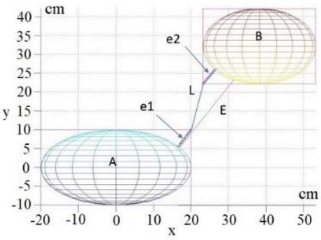


Fig. 1. The diagram of variables E, $e1$, $e2$ and L.

2.2 Compensation Based on Gaussian Function

Fig. 2 (b) shows the underestimation and overestimation about distance estimated by inner ellipse method, where blue regions represent underestimation, *i.e.* $L-(E-e1-e2)$ < 0, the red regions represent overestimation, *i.e.* $L-(E-e1-e2) > 0$. The white block indicates that the distance error is 0 cm. In this paper, two one-dimensional Gaussian function [16] are multiplied to compensate for overestimation and underestimation, and the purpose is to make the color close to white blocks, as shown in Fig. 3.

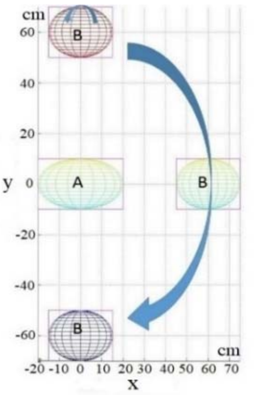


Fig. 2. (a) Revolution and rotation.

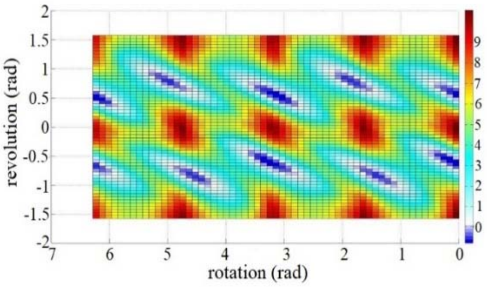


Fig. 2. (b) Distance error.

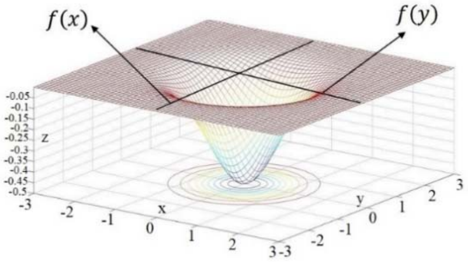


Fig. 3. Multiplication of two one-dimensional Gaussian functions.

In the environment, the data table of motion state is calculated according to the way that the motion of the rectangle and its inner ellipse. The compensation formula is Eq. (2).

$$f(x)* f(y)*a \tag{2}$$

Where $f(x) = \exp\left(-\frac{(x-x_0)^2}{2\sigma_x{}^2}\right)$ and $f(y) = \exp\left(-\frac{(y-y_0)^2}{2\sigma_y{}^2}\right)$ are Gaussian functions, the corresponding expectation is x_0, y_0, which are position parameters, and determine the position of Gaussian function, and are the positions of the lowest distance error point. The standard deviations σ_x, σ_y are the long and short axis of ellipses. x is the rotation angle, y is the revolution angle. a is a parameter which is used to adjust the distance error of Gaussian function.

The compensation process for blue blocks is similar to that for red blocks. First, blue blocks are processed. After the data graph is cut, three lowest points is found respectively to set the expectations of Gaussian functions in the three blue blocks. Fig. 4 shows how to find the long and short axis is to rotate the ellipse by 25° and use the lowest point as the reference. Then find the distance farthest from the lowest point along the 25° direction in the blue block as the first standard deviation, and then use the same way to find the lowest point in the same block at 115° and the lowest distance is set as the second standard deviation. The ellipse form of each block is obtained by multiplying two Gaussian functions with the same expectation and different standard deviations, and finally multiplying parameter α.

It is essential to design parameter α. There are three blue blocks in the upper right corner in Fig. 4, and three sets of Gaussian functions need to be designed. Since the processes are similar, the middle blue block is used as an example to design the Gaussian function. The Gaussian function is applied to compensate the value of the lowest point of the distance error, but the maximum value is 1, so parameter α is needed to adjust the maximum value to compensate the lowest point of the distance error. In the paper, the distance between the two elliptical centers is 50 cm to150 cm and a set of parameters α are designed, as shown in Table 2.

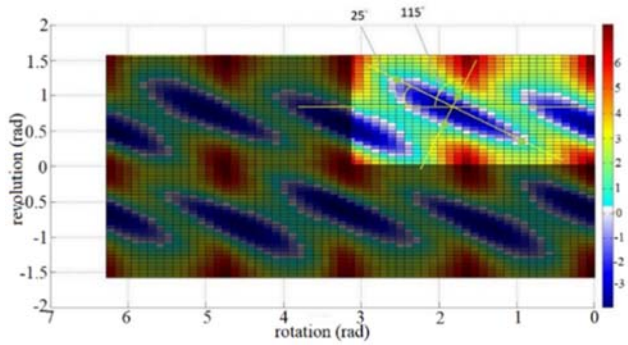


Fig. 4. Cutting blue blocks and the angle to find the longest and shortest axes.

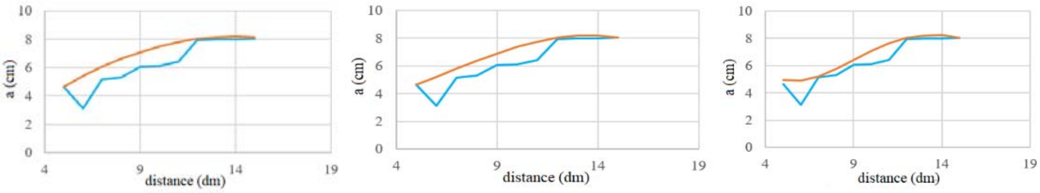
Table 2. The data of parameter α and polynomial approximation.

R	α	Quadratic polynomial	Error	Cubic polynomial	error	Quartic polynomial	error
50	4.6512	4.6600	0.0114	4.66	0.00749	4.97	0.31689
60	3.1151	5.3974	2.2823	5.22424	2.20914	4.89	1.77684
70	5.1785	6.044	0.8655	5.80187	0.62337	5.22	0.03739
80	5.293	6.6024	1.3094	6.368	1.075	5.78	0.48314
90	6.0451	7.0726	1.0275	6.89905	0.85395	6.44	0.38995
100	6.1409	7.4546	1.3137	7.37144	1.23054	7.08	0.94024
110	6.4195	7.7481	1.3289	7.76159	1.34209	7.63	1.20959
120	7.9387	7.954	0.0153	8.04592	0.10722	8.02	0.08104
130	7.9737	8.0714	0.0977	8.20085	0.20715	8.22	0.24639
140	8.0028	8.1006	0.0978	8.2028	0.2	8.22	0.2205
150	8.0273	8.0273	0.0143	8.02819	0.00089	8.05	0.02139

R(cm) is the distance between the center points of two ellipses.

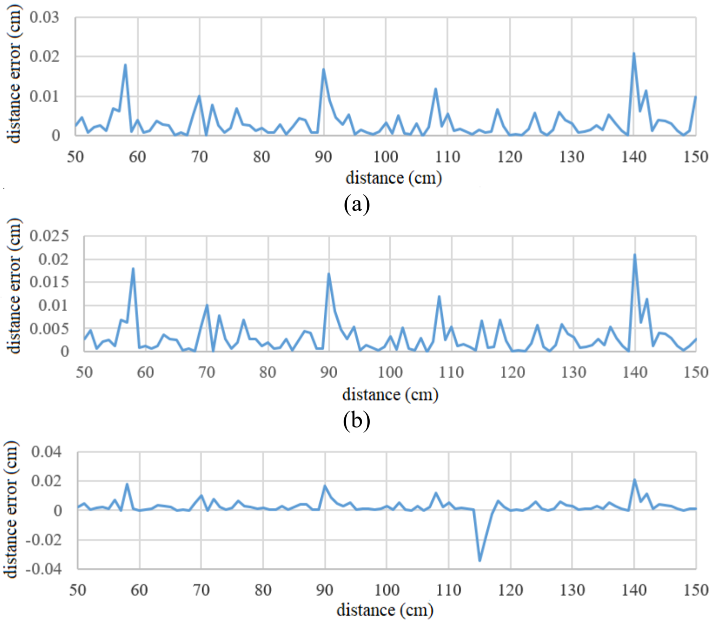
Polynomial functions with different powers are used to approximate the values of α in Table 2 and the results are in Fig. 5. In the whole motion, the distance errors between several functions and the real curve is shown in Fig. 6. The value of original α is adjusted every 10 cm starting from the distance (50 cm) between the center points of the two ellipses. It is determined that the distance error in Fig. 5 will be greater than zero with the designed variables. However, the α values are connected in a linear way, and the more approximate this curve is, the overfitting situation where the distance error of the quartic polynomial function is less than 0 may occur in Fig. 6. The total error of the cubic polynomial is less than twice polynomial, so the blue block is designed by using a cubic polynomial, as shown in Eq. (3). Other blue blocks are designed in the same way to find the most appropriate polynomial function.

$$\alpha = -0.00393 * \left(1 + \frac{(R-50)}{10}\right)^3 + 0.02962 * \left(1 + \frac{(R-50)}{10}\right)^2 + 0.2042 * \left(1 + \frac{(R-50)}{10}\right) + 4.1288 \quad (3)$$



(a) Quadratic polynomial function. (b) Cubic polynomial function. (c) Quadric polynomial function.

Fig. 5. The curve of parameter α and polynomial approximation curve. Where the red line is a curve of parameter α, and the blue line is a curve of polynomial function.



(c)

Fig. 6. The distance error between polynomial function and real curve; (a) Quadratic polynomial function; (b) Cubic polynomial function; (c) Quadric polynomial function.

Next, dealing with the red blocks. According to the red blocks in Fig. 2. Like the compensation way of blue blocks, two one-dimensional Gaussian functions are multiplied to compensate the distance error and make the red color close to the white area in the data graph. The environment of red block is the same as that of blue block, the expectation is the parameter that determines the position of the Gaussian function, but unlike the blue block previously processed, here is the position of the highest distance error point, and the standard deviation is the long and short axis of the ellipse. As shown in Fig. 7, in the 6 red blocks after the data graph is cut. The upper 3 red blocks are treated in the same way as the 3 blocks below, and the three highest points are to be found respectively as the expectation of the Gaussian function. The standard deviation is the position farthest from the highest point in the direction of 180° degree and 90°.

For the design of parameter α, the red blocks are also divided into three groups. Similar to the previous operation of blue blocks, the Gaussian function is also used to compensate the value of distance error in the highest point, and then the parameter α is designed to adjust the maximum of Gaussian functions so that it can subtract the value of the distance error in the highest point. Then polynomial functions are used to approximate the curve of parameter α, and finally choosing the polynomial function with the smallest error.

In summary, after compensating the blue blocks and suppressing the red blocks. As shown in Fig. 8, the horizontal axis is the distance between two ellipses and the vertical axis is the distance error. It reveals that the distance errors from 50 cm to 150 cm are calculated according to the distances between the two elliptical center points. Fig. 8 shows that in this motion space, when the distance between two ellipse centers is fixed, the estimated distance of each state falls within the maximum and minimum error intervals, which neither underestimates nor restrains the overestimated area.

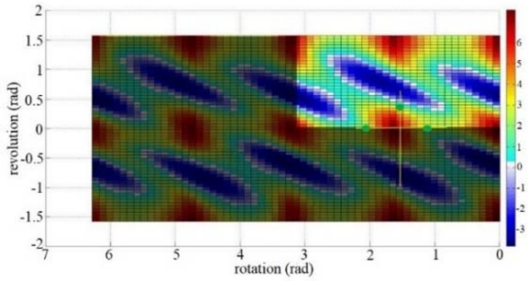


Fig. 7. The way to cut red blocks.

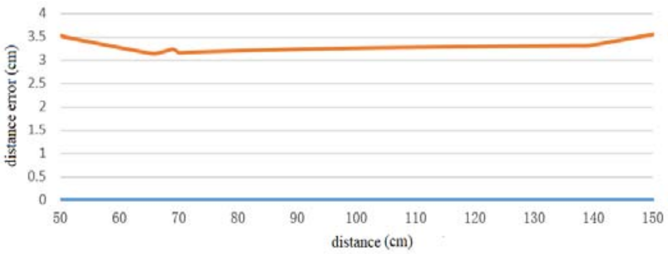


Fig. 8. Maximum and minimum error. The blue line represents the minimum distance error, and the orange line is the maximum distance error.

2.4 Linear Interpolation Method

Linear interpolation [17] is an approximate method for calculating the values of known or unknown functions by using the ratio relationship based on the values of a set of known independent variables of unknown functions and their corresponding function values.

In the simulated environment, the cutting angle is not enough to simulate all angles and distances such as rotation angles, revolution angles and center points, but the linear interpolation method is used to solve this problem. The interpolation formula is Eq. (4).

$$Y = \mathit{inserttf}(x, y, X, \mathit{method}) \tag{4}$$

Where *insertf* is interpolation function, the vector *x* is the coordinate (input value) of the data point, the vector *y* is the coordinate of the data point (output value), *X* is the interpolation point, *Y* is the fitting function of the output after interpolation, and the method represents the way of linear interpolation.

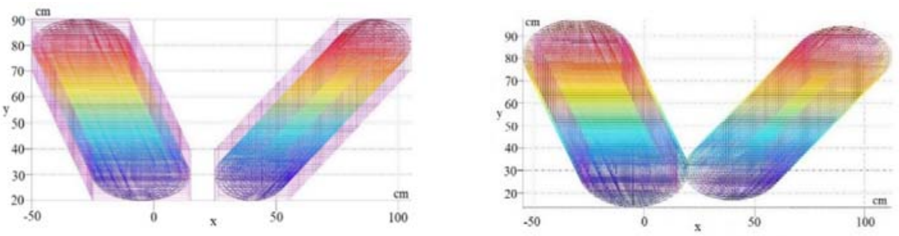
3. SIMULATION AND EXPERIMENT

The experiment aims to verify the effectiveness of the collision detection method with inner ellipses proposed in estimating the actual distance between links and obstacles. At the same time, compared with the circumscribed ellipse method, the accuracy of distance estimation is verified. The circumscribed ellipse method has the problem of overestimation, which can also lead to the problem of underestimation when the shape of obstacles or links is special. The same problems exist when inner ellipses without compensation are used to model links or obstacles. Therefore, in each scenario, the minimal distance between two ellipses is calculated by the inner ellipse method and the circumscribed ellipse method respectively. Then, the result is used to calculate the distance error and to compare the accuracy of the distance estimated by the two methods.

According to the working environment of manipulators, three scenarios are designed. The scenario 1 is simple, it simulates the situation that two robot arms close to each other. On the basis of scenario 1, in the scenario 2, it simulates the situation that the robot link B rotates around link A and moves to link A. In the scenario 3, the motion between a two-axis manipulator and an obstacle is simulated. In the three experiments, the data of the two links are shown in the Table 1.

3.1 Scenario 1

In this experiment, as shown in Fig. 9, link A and link B move towards to each other by translational motion. In the Fig. 9 (a), each red rectangle represents link A on the left, each red rectangle represents link A on the right, and each rectangle contains its inner ellipse. At the beginning, link A and link B are 10 cm apart. As two links close to each other, the distance between them gradually decreases. Similarly, in the Fig. 9 (b), each red rectangle represents link A on the left, each red rectangle rep-resents link A on the right, and each rectangle is contained by its circumscribed ellipse.



(a) Inner ellipses method. (b) Circumscribed ellipses method.

Fig. 9. The motion of links.

Furthermore, the changes of distances between two links calculated by the inner ellipse method and the circumscribed ellipse method are shown in Fig. 10. It can be seen the distance calculated by the inner compensation ellipse is 10 cm more than that estimated by the circumscribed ellipse, and the distance error is 2 cm between the distance of the inner compensation ellipse and the actual distance during the whole motion. In Step 1, the distance calculated by the inner ellipse method is about 5 cm smaller than the actual distance. As the actual distance drops to 10 cm, the distance error of the inner ellipse method decreases. After 30 steps, the distance error is close to 0 cm. However, the distance between the distance estimated by circumscribed ellipse method and the actual distance is always about 15 cm. After 49 steps, the distance of the circumscribed ellipses is less than 0 cm, indicating that the links have collided, but there is still 10 cm available in practice. As whole the inner ellipse method is better than the circumscribed ellipse method in scenario 1.

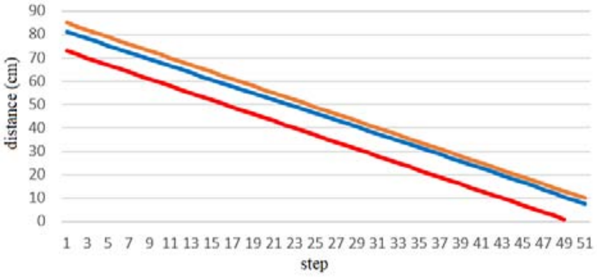
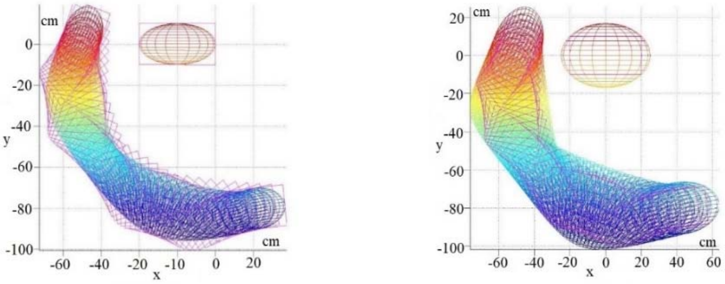


Fig. 10. The distance calculated by the inner ellipse and the circumscribed ellipse methods. The orange line represents the actual distance of the two links, the blue line represents the distance calculated by the inner ellipse method, and the red line represents the distance calculated by the circumscribed ellipse.

3.2 Scenario 2

Based on scenario 1, in this experiment, as shown in Fig. 11, link A remains stationary, link B rotates around link A, and move toward link A at the same time. In the Fig. 11 (a), the link B is located on the left and the left half represents its trajectory and posture of rotation and translation. The link A is located in the upper right corner, and they are 60 cm apart at the beginning. Then, the link B rotates and translates simultaneously close to A, and the distance between two links decreases, until it drops to about 14 cm. Similarly, in the Fig. 11 (b), each red rectangle represents link B on the left, the red rectangle rep-resents link A on the right, but each rectangle is contained by its circumscribed ellipse.



(a) Inner ellipse method. (b) Circumscribed ellipse method.

Fig. 11. The motion of links.

Further, the changes of distances calculated by two methods are shown in Fig. 12. Because of the rotational movement, the distance is reduced in a wave-like manner. From Steps 1-20, it shows that the blue line represented by the inner ellipse method is closer to the orange line represented by the actual distance. The distance error calculated by the inner ellipse is about 1 cm during the whole motion. The distance error is 0 cm when the step is 5, 6 or 17, and the error is obviously less than that calculated by the circumscribed ellipse method. From Steps 23-26, and Steps 30-41, the result is similar to the situation in the Fig. 10, the distance error calculated by inner ellipse is less than 1 cm, but the distance error of the circumscribed ellipses is about 3 cm. Although from Steps 20-22, Steps 27-29, the estimated distance of the circumscribed ellipses is closer to the actual distance. The reason is that when making the elliptical area minimize in the circumscribed ellipse method, the tangent point of the ellipse will fall on each vertex, so that the shortest distance of the ellipses is almost equal to the shortest distance of the links, and the distance error is 0 cm. However, in terms of overall performance, the inner compensation ellipse method is superior to the circumscribed ellipse method.

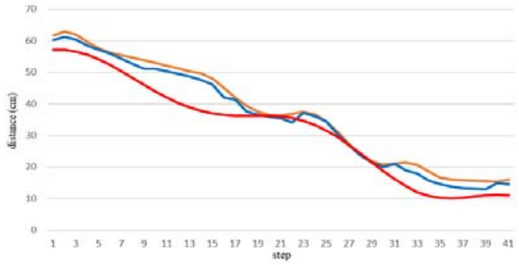


Fig. 12. The calculated distances.

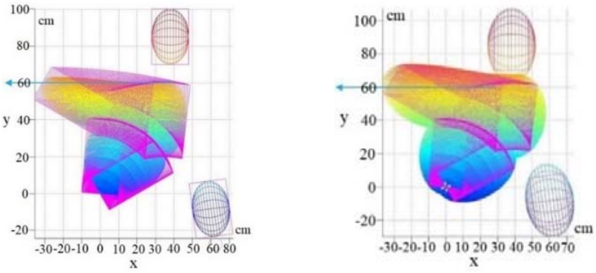
3.3 Scenario 3

In scenario 3, the changes of the distance between the two-axis robotic arm and the obstacle is analyzed. The motion posture of robotic arms and obstacles is shown in Fig. 13. The two rectangles in the middle represent the two link A and act as the two-axis manipulator, the rectangle in the upper and the rectangle in the bottom represent link B, act as obstacles. In Fig. 13 (a), each rectangle contains its inner ellipse. In Fig. 13 (b), each rectangle is covered by its circumscribed ellipse.

In Fig. 13. The two obstacles remain stationary, and the coordinate point is established with the end of the robotic arm, that is, the end of the second link A, and the robotic arms move horizontally to the left. At the same time, the two links A can rotate, just like a real two-axis robot arm. Note that the center coordinates of two links A are obtained by using the forward kinematics [18], and the angle of two rotational axes and the posture information of the arms are obtained by using the inverse kinematics [19].

Now analyzing the influence of different obstacles on the distance estimation of two links. Firstly, the influence of the bottom obstacles on the distance estimation of the first link is analyzed. The actual distance changes between the obstacle and the first link A, the changes of distances calculated by the inner ellipse method and the circumscribed ellipse method are shown in Fig. 14 (a). From Steps 1-15, it can be seen that the actual distance is about 10 cm, and the distance error calculated by two methods is 0 cm. But after Step 15, as the actual distance increases, until it is 37 cm, the blue line is located between the orange line and the red line. Specially, the distance error calculated by the inner ellipse method is about 3 cm, and the distance error calculated by the circumscribed ellipse method is about 5 cm. During the whole motion process, the inner compensation ellipse method is superior to the circumscribed ellipse method.

Then, the influence of obstacles at the bottom on the distance of the second link A is analyzed. The changes of distances are shown in Fig. 14 (b). During the whole motion, from Steps 1-69, the distance calculated by the inner ellipse is closer to the actual distance, the average distance error is 1 cm, while the distance error calculated by the circumscribed ellipse method is about 5 cm. Obviously, the inner ellipse method is better than the circumscribed ellipse method.



(a) Inner ellipse. (b) Circumscribed ellipse.

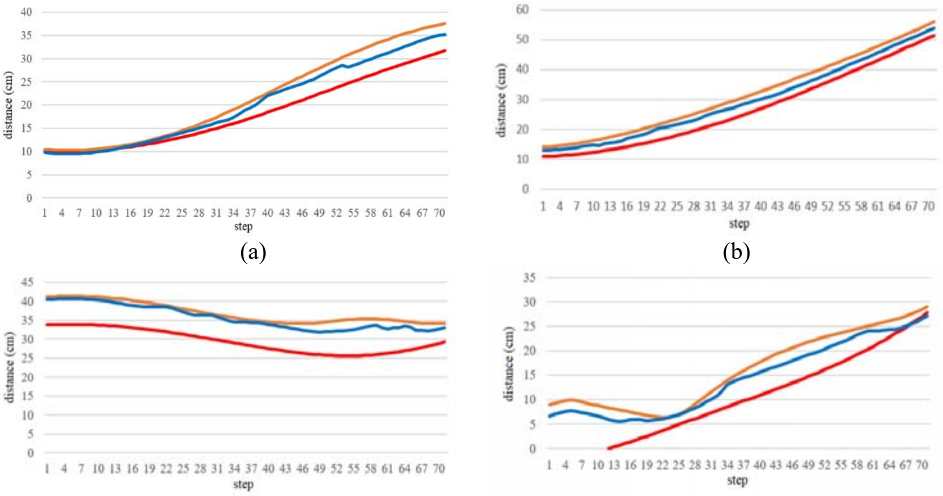
Fig. 13. motion posture of obstacles and robotic arms.

Then, the influence of obstacles at the top on the distance of the first link A is analyzed. The changes of distances calculated by two method are shown in Fig. 14 (c). Similar to the previous results, from Steps 1-40, the distance error is less than 1 cm. After Step 40, the maximal distance error calculated by the inner ellipse method is about 3 cm. However, the distance error estimated by the circumscribed ellipse method is about 6 cm, and the distance error is about 10 cm from Steps 50-59. On the whole, the inner ellipse method is better than the circumscribed ellipse method.

Finally, the influence of obstacles at the top on the distance of the second link A is analyzed. The changes of distances calculated by two methods are shown in Fig. 14 (d).

From Steps 1-13, the distance calculated by the circumscribed ellipse method is less than 0 cm, indicating that the top obstacle collides with the second link A, but the actual distance between them is greater than 8 cm. From Steps 13-26, the distance error calculated by the circumscribed ellipse method is reduced from 8 cm to 2 cm. From Steps 26-67, the distance error is 7 cm on average. However, the distance error between the distance of inner ellipses and the actual distance is 0 cm to 3 cm, and the accuracy of distance estimated is higher. Although from Steps 68 and 69, the distance error calculated by the circumscribed ellipse method is smaller. In terms of overall performance, the inner ellipse method is superior to the circumscribed ellipse method in distance estimation.

In conclusion, the experiment shows that the best distance estimated by the circumscribed ellipse method falls near the corners of each other, while the worst situation is where the long axis is opposite to the long axis. Under these two postures, when the circumscribed ellipses have collided, the inner compensation ellipse has 5-7 cm to be left, and when the distance estimated by circumscribed ellipse estimates method is closest to actual distance, the error between two methods is about 1 cm. It can be proved that the proposed inner compensation ellipse method can estimate the distance accurately at the worst estimation of the circumscribed ellipse. And under the best estimation of the circumscribed ellipse, the distance estimated by the method proposed in this paper is not bad.



(a) (b)

(c) (d)

Fig. 14. The distance changes between the obstacle and the link; (a) The influence of the bottom obstacle on the first link; (b) The influence of the bottom obstacle on the second link; (c) The influence of the top obstacle on the first link; (d) The influence of the top obstacle on the second link.

4. CONCLUSIONS

In this paper, a compensation method is proposed to estimate the actual shortest distance between links and obstacles modeled by inner ellipses. Firstly, Lowner-John ellipse is used to build three-dimensional data table about the motion of links. Then the method introduces a way to compensate the data graph by Gaussian function to solve the problem of overestimation and underestimation. Three experimental scenarios demonstrate the effectiveness of the proposed method in estimating the actual distance between links and obstacles. However, the method used in this paper adopts a fixed structure, where the compensation method needs to adjust the parameters accordingly. A highly prospective development direction in the future is to extend to a higher degree-of-freedom, and hopefully applied machine learning to generating the data tables to achieve more effective compensation and estimate distance more accurately.

REFERENCES

1. J. Bimbo, L. D. Seneviratne, K. Althoefer, *et al.*, "Combining touch and vision for the estimation of an object's pose during manipulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4021-4026.
2. J. M. Bravo, T. Alamo, M. Fiacchini, *et al.*, "A convex approximation of the feasible solution set for nonlinear bounded-error identification problems," in *Proceedings of IEEE Conference on Decision and Control*, 2007, pp. 5743-5748.
3. X. Guo, L. Xie, and Y. Gao, "Optimal accurate Minkowski sum approximation of polyhedral models," in *Proceedings of International Conference on Intelligent Computing*, 2008, pp. 179-188.
4. L. A. Piegl, P. Kulkarni, and K. Rajab, "Algorithm for the removal of rectangle containment for rectangle spline generation," *Computer-Aided Design and Applications*, Vol. 12, 2015, pp. 1-8.
5. J. S. Liu and W. H. Pan, "Automatic computation of range of motion for coherent contact maintenance," in *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2008, pp. 682-691.
6. X. Cheng and M. Zhao, "Analysis on the trajectory planning and simulation of six degrees of freedom manipulator," in *Proceedings of International Conference on Mechanical*, *Control and Computer Engineering*, Vol. 1, 2018, pp. 385-387.
7. C. Fragkopoulos and A. Gräser, "Dynamic efficient collision checking method of robot arm paths in configuration space," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2011, pp. 784-789.
8. C. J. Ong and E. G. Gilbert, "Fast versions of the Gilbert-Johnson-Keerthi distance algorithm: Additional results and comparisons," *IEEE Transactions on Robotics and Automation*, Vol. 17, 2011, pp. 531-539.
9. Z. Wei and L. Ying, "A fast collision detection algorithm based on distance calculations between NURBS surfaces," in *Proceedings of International Conference on Computer Science and Electronics Engineering*, 2012, pp. 534-537.
10. E. Rimon and S. P. Boyd, "Obstacle collision detection using best ellipsoid fit," *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 18, 1997, pp. 105-126.
11. R. A. Sasongko and S. S. Rawikara, "3D obstacle avoidance system using ellipsoid geometry," in *Proceedings of IEEE International Conference on Unmanned Aircraft Systems*, 2016, pp. 562-571.

12. A. Y. Uteshev and M. V. Yashina, "Distance computation from an ellipsoid to a linear or a quadric surface in IR," *Computer Algebra in Scientific Computing*, 2007, pp. pp. 392-401.
13. C. de Michele, "Simulating hard rigid bodies," *Journal of Computation Physics*, Vol. 4, 1999, pp. 7-25.
14. J. O. Chong and E. G. Gilbert, "Fast versions of the GilbertJohnson-Keerthi distance algorithm: additional results and comparisons," *IEEE Transactions on Robotics and Automation*, Vol. 17, 2001, pp. 531-539.
15. X. Jia, Y. K. Choi, B. Mourrain, *et al.*, "An algebraic approach to continuous collision detection for ellipsoids," *Computer Aided Geometric Design*, Vol. 28, 2011, pp. 164-176.
16. H. W. Guo, "A simple algorithm for fitting a Gaussian function," *IEEE Signal Processing Magazine*, Vol. 28, 2011, pp. 134-137.
17. T. Blu, P. Thévenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Transactions on Image Processing*, Vol. 13, 2004, pp. 710-719.
18. X. Zheng, Y. Zheng, Y. Shuai, *et al.*, "Kinematics analysis and trajectory planning of 6-DOF robot," in *Proceedings of IEEE 3rd Information Technology*, *Networking*, *Electronic and Automation Control Conference*, 2019, pp. 1749-1754.
19. F. Wang and Z. Zhao, "Research on inverse kinematics of robot based on motion controller," in *Proceedings of IEEE International Conference of Intelligent Robotic and Control Engineering*, 2018, pp. 34-37.

Haobin Shi (史豪斌) received the Ph.D. degrees in Computer Science and Technology from Northwestern Polytechnical University, Shaanxi, China. He is currently a Professor at Northwestern Polytechnical University. His research interests include wireless sensor networks and communication networks.

Meng Liang (梁猛) received the B.S. and M.S. degrees in Computer Science and Technology from Northwestern Polytechnical University, Shaanxi, China. His research interests include intelligent robot systems, intelligent decision, machine learning.

Kao-Shing Hwang (黃國勝) received the M.M.E. and Ph.D. degrees in Electrical and Computer Engineering from Northwestern University, Evanston, IL, U.S.A., respectively. Since August 1993, he has been with National Sun Yat-sen University in Taiwan. He was the chairman of the Electrical Engineering Department (2003-2006). His research interests include methodologies and analysis for various intelligent robot systems, machine learning, embedded system design, and ASIC design for robotic applications.

Chin-Wei Fan (王威幀) received the B.S. and M.S. degrees in Electrical Engineering from National Sun Yat-sen University, Taiwan. His research interests include electrical engineering robotics, machine learning.