Speed Alteration Strategy for Multi-Joint Robots in Co-Working Environment

*Kao-Shing Hwang, Member, IEEE, Ming-Yi Ju, and Yu-Jen Chen

Abstract-- A collision-free trajectory planning method based on speed alternation strategy for multi-joint manipulators in overlapped working envelopes is proposed. Since the shape of a robot's link is usually rectangular or cylindrical approximately, the proposed method models a robot's link mathematically by quadric primitives, such as ellipsoids and spheres. The occurrence of collisions between links can be predicted easily by means of relative coordinate transformations and geometric deformations between those ellipsoids. Furthermore, the collision-trend index which is defined by projecting the ellipsoids geometrically onto the Gaussian distribution plays a significant role in searching the optimal resolution in the proposed collision-avoidance method. Two Motoman robots of YASUKAWAI Co. are conducted to demonstrate the performance of the proposed methods.

Index terms-- trajectory planning, collision detection, collision-trend index, robotics.

I. INTRODUCTION

The use of multiple robots in a common workspace is essential to enhance the utilization of robots, increase productivity, and improve the versatility of potential applications. However, it usually leads to the problems of cooperation since the robot may become obstacles to each other. Therefore, motion planning must take account of collision avoidance between moving robots. One approach to collision-free motion planning for multiple robots is to decompose the problem into two sub-problems: path planning and trajectory planning [1]. Path planning finds the robots’ geometric paths that do not intersect static obstacles, and trajectory planning determines how fast each robot must move along its path to avoid collision with others. The paper focuses on the problem of collision-free trajectory planning for multiple robots.

To solve the problem, plenty of methods have been proposed in recent years. Chang proposed [2] a simple time delay method for avoiding collisions between two general robot arms. In his method, robot links are approximated by polyhedron and the danger of collision is expressed by the function of the distance between two robots. By the collision map scheme, Hwang and his colleagues [3] presented that the links of robots in 3D can be simplified to the different rectangles in 2D Space/Time graphs. Through finding the optimal path between the rectangles on the Space/Time graph, robots can avoid collision easily by means of velocity alteration. Similarly, Griswold [4] proposed an optimization approach which provides acceleration or deceleration strategy for a mobile robot under the constraints of defined collision probability. Besides, a unified approach for robot motion planning with moving polyhedral obstacles was presented by Shih and his colleagues [5]. In their method, 2D and 3D objects can be represented by some proper polytopes. The optimal collision free path can be found by constructing a family of feasible
collision-free trajectories, which simply connect polytopes satisfying the speed and time constraints, between the start and the goal polytope. This kind of approach differs from the methods using the fixed moving path stated above. However, the approach to search an optimal collision-free path becomes much complex. While the space dimension or the number of obstacles is increased, the number of feasible paths increases exponentially accordingly. Superquadric modeling technique has been applied to constructing an artificial potential field which consists of the geometric models of the robot and the obstacles. By calculating the avoidance and approach potentials, respectively, a collision-free path can be derived [6].

Taking mathematical representation and normalization of collision-trend index into consideration, links of robots are suggested to be modeled as simple non-deformed ellipsoids in the proposed approach. By modeling the links of robots as a series of connected ellipsoids and restricting each robot to move along a straight-line path, the proposed method uses the defined collision-trend index as a penalty index for delaying the slave robot to guarantee that the two robots can reach their targets without any collision.

II. MANIPULATOR MODELING

How to model the robot’s shape correctly and simply is very important to an efficient collision detection algorithm. For a modeling method, its criteria can be summarized by

- The model must represent the physical system precisely as possible.
- The model must be simple enough to ensure that the algorithm can be solved fast enough to secure the real-time operation of the manipulator.

In order to increase the accuracy of the model, it is practical to model each link individually by creating a volume representative of each link, rather than to model a robot as a single body. Generally, there are three methods which are usually used to achieve this: polyhedral modeling, cylindrical modeling and spherical modeling.

Although accuracy can be obtained in polyhedral model representation for each link, it is computationally intensive to construct. The description of complex shapes represented by a polyhedral model often requires many planar faces and edges [7-9]. Moreover, high quality rendering of complex or curved shapes requires hundreds of planar faces. Besides, it becomes more difficult when rotations are introduced. As a result, the detection of interactions with other objectives becomes more complex and impairs the real-time operation of the arm. In cylindrical modeling, each link can be effectively ‘inserted’ into a tube. The problem is that, when we try to view a cylindrical model with respect to a Cartesian coordinate system, the mathematical expression becomes complicated that restricts the real-time operation. In sphere modeling, an immediate advantage is simplicity in mathematical formulation. Since a sphere is rotationally invariant, only the calculation of the original position is needed. From the point of view of collision detection, it just needs to compare the distance between an obstacle and the sphere model with the radius of the sphere. However, spherical representation for robot links leads to a large volume of waste.

It can be seen that the polygon method is accurate but computation intensive; the representation of cylinders is elegant but complicated; the sphere modeling is quite inaccurate. Therefore, to seek a method which is able to model the link of a robot simply and estimate the relative spatial distance efficiently is important in the initial stage for collision-detection problems. Since most of robot’s links are bar-like or cylinder-like shapes, proper ellipsoids are able to model these links well. Based on the principles mentioned previously, the minimum-volume enclosing ellipsoids [10] for shape modeling are used in our work.

For a robot, if each link is properly modeled with an ellipsoid, any point on the link can be easily represented as:

\[
\mathbf{x} = A_i^0 \text{Rot}(z, \theta_i) T_i \mathbf{x}'
\]

where \( \mathbf{x} = (x, y, z) \) represents the position vector for a point on the ellipsoid which contains link \( i \) of a robot and has the radii \( r_{ix}, r_{iy}, r_{iz} \), and \( r_{zi} \), with respect to (w. r. t.) the robot base coordinate system; position vector \( \mathbf{x}' \) represents the same point but w. r. t. the center of link \( i \):

\[
A_i^0 = \begin{bmatrix} n_x & 0 & a_x & p_x \\ n_y & 0 & a_y & p_y \\ n_z & 0 & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]
represents the homogeneous transformation from the base frame to link \((i-1)\)'s frame; \(\text{Rot}(z, \theta_i)\) is the rotational transformation with angle \(\theta_i\) w. r. t. the \(z\) axis of link \(i\)'s coordinate system;

\[
T_{ci} = \begin{bmatrix}
1 & 0 & 0 & X_{ci} \\
0 & 1 & 0 & Y_{ci} \\
0 & 0 & 1 & Z_{ci} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

represents the translation from joint \(i\) to the center of link \(i\). Therefore \(\Delta x'_i\) can be evaluated as:

\[
\Delta x'_i = T_{ci}^{-1} \text{Rot}^{-1}(z, \theta_i) A_{i-1}^{-1}
\]

(2)

Eq.(2) can be represented as a standard ellipsoidal equation:

\[
\frac{x'^2}{r_{xi}^2} + \frac{y'^2}{r_{yi}^2} + \frac{z'^2}{r_{zi}^2} = 1,
\]

where

\[
\begin{align*}
\Delta x'_i &= (n_i \cos \theta_i + o_i \sin \theta_i)X_{ci} \\
&+ (n_i \cos \theta_i + o_i \sin \theta_i)y \\
&+ (n_i \cos \theta_i + o_i \sin \theta_i)z \\
&-(n_i \cdot p)\cos \theta_i + (o_i \cdot p)\sin \theta_i + X_{ci}; \\
\Delta y'_i &= (-n_i \sin \theta_i + o_i \cos \theta_i)X_{ci} \\
&+ (-n_i \sin \theta_i + o_i \cos \theta_i)y \\
&+ (-n_i \sin \theta_i + o_i \cos \theta_i)z \\
&+(n_i \cdot p)\sin \theta_i -(o_i \cdot p)\cos \theta_i -Y_{ci}; \\
\Delta z'_i &= ax + ay + az -(a_i \cdot p + Z_{ci}).
\end{align*}
\]

Thus, any link of a robot can be expressed as:

\[
E(x, y, z) = \frac{x'^2}{r_{xi}^2} + \frac{y'^2}{r_{yi}^2} + \frac{z'^2}{r_{zi}^2} = 1
\]

(3)

The relative location between a point, \((x, y, z)\), and an ellipsoid which models a link of a robot can be easily verified by Eq.(3): if \(E(x, y, z) > 1\), then the point must be outside the link, and vice versa. Unfortunately, the problem encountered here is not as simple as the case where a point collides with ellipsoidal objects. Instead, collisions occur between eventually ellipsoids if robot is modeled in this way. Mathematical inspection of the intersection of ellipsoids is not practical due to the computation complexity increasing exponentially along with the number of ellipsoids. Therefore, a method is proposed here to express potential collisions between the links of robots by transforming the problem into the one which only needs to check the geometric relationship between a point and several ellipsoids.

The first stage in the approach is to scale down the active ellipsoid, which presents the link that needs to detect if there is any potential collision with the other links. The scaling operation, which should be executed at the origin of the link's own coordinate frame, transforms that ellipsoid into a spherical shape or a large point. Therefore, the reference...
coordinate system should be moved from the base frame to the link’s coordinate frame. In this manner, the other ellipsoids in the working space can maintain relative geometric equivalence while they are also transformed by the same scaling operation simultaneously. The consequences of these transformations are shown in Fig. 2. The detail mathematical derivation for relative transformations between multi-link manipulators is given in Appendix.

![Diagram](a) (b) (c)

Fig. 2(a)-(c). The sequential transformations of the collision detection algorithm.

According to the results listed in Appendix, collision detection between two links can be simplified to test whether the point, \((0, 0, 0)\), which represents the shrunk active link, resides inside or outside the function of \(E_{ij}(x'', y'', z'')\) representing the ellipsoid equation of the other link:

\[
E(x'', y'', z'') |_{x''=0, y''=0, z''=0} \leq 1. 
\] (4)

If the condition holds, this means that the active link has collided with the other links. Thus, if each of \(m\) robots has \(n\) links modeled using ellipsoids, then the intersection check between the robots is simplified to testing whether the point falls inside or outside \(n^{-1}m\) transformed ellipsoids. The collision detection algorithm between two links of robots can be summarized as:

\[
\text{if } \quad \begin{align*}
E_{ij}(x'', y'', z'') |_{x''=0, y''=0, z''=0} &\leq 1 \\
\Rightarrow \quad \text{collision}
\end{align*}
\] (5)

where \(1 \leq i \leq n; 1 \leq j \leq n\).

\(E_{ij}(x'', y'', z'')\) is the transformed ellipsoid of link \(i\) of the other robot w. r. t. the coordinate frame of active link \(j\) of the other robot.

A. Collision-trend Index

The procedure above can be used to scale down the active ellipsoid into a point while the other ellipsoids are dilated accordingly. Therefore, computational complexity of checking if two ellipsoids overlap can be reduced. However, there is still no suitable measurement to indicate occurrence of potential collision. Unlike in the case of spheres, the Euclidean distance between a point and the center of an ellipsoid cannot express how close two modeled objects to each other are. Evaluating the orthogonal distance between a point and the surface of an ellipsoid may lead to complicated derivation. Therefore, a measurement, the collision-trend index, is defined to describe the degree of imminent jeopardy while an active link is approaching some other links. The collision-trend index is derived by conceptual projecting the scaled ellipsoidal equation onto a Gaussian distribution function.

A 3D Gaussian distribution function can be written as:

\[
P(x) = (2\pi)^{-3/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2} k\right) 
\] (6)

The symmetric matrix \(\Sigma\) can be diagonalized; if scale value \(k\) can be regarded as the value of \(E(x'', y'', z'')\), i.e., \(E(x'', y'', z'') = k\), then the collision-trend index between two links is defined as:

\[
P_{ij}(x'') |_{x''=0, y''=0, z''=0} = \frac{(2\pi)^{-3/2} e^{-\frac{1}{2} E_{ij}(x'', y'', z'')}}{(r_{ij}^0 + r_i)(r_{ij}^0 + r_j)(r_{ij}^0 + r_i + r_j)} 
\] (7)
where $r_{i'}$, $r_{j'}$, and $r_{a'}$ are radii of the ellipsoid containing link $i$; $r_a$ is the smallest radius among the three axes of the ellipsoid. The collision-trend index varies with robot motion. For example, the collision-trend index between link $i$ of robot 1 and link $j$ of robot 2 in Fig.1 is associated with the consecutive angular changes from joint 1 to joint $i-1$ of robot 1 and from joint 1 to joint $j-1$ of robot 2. Therefore, when $E(x'^a, y'^a, z'^a)$ is set to one (i.e., $k=1$), the probability, called the jeopardizing collision-trend index, which represents occurrence of collision between the links is equal to:

$$
P_c = \frac{0.038509}{(r_{a'} + r_a)(r_{a'} + r_a)(r_{a'} + r_a)} \exp\left[-\frac{1}{2} \right]
$$

(8)

Therefore, if the collision-trend index $P_c(\theta_{11}, ..., \theta_{i-1}, \theta_{11}, ..., \theta_{j-1}) > P_c$, then it assures that collision occurs; $\theta_{i1} \sim \theta_{i-1}$ represents the angles from joint 1 to joint $i-1$ for robot 1, and $\theta_{21} \sim \theta_{2j-1}$ represents the corresponding joint angles of robot 2. To execute transformations and collision detection for an ellipsoid requires 4 trigonometric calculations, $13+45*n$ multipliers, and $7+33*n$ additions to evaluated the collision-trend index, where $n$ is the number of the other robot’s links. The derivation for the proposed approach seems to be complex, but computation complexity is relaxed in real applications. Only the parameters of ellipsoids which are used to model objects should be instantized into the derived transformation matrix. It is convenient and efficient for the jobs which need to execute collision detection repeatedly. Although the collision-trend index method and other distance function methods [11, 12] have the same complexity $O(nm)$, where $n$ and $m$ are the numbers of primitives used to represent the two objects, the proposed method adopts a mathematical strategy to model a robot's link as an ellipsoid instead of using a case-dependent strategy to model a link as an union of spheres. Consequently, the use of collision-trend index should be more efficient for collision detection as the number of robot's link is increased. Besides, the collision-trend index can also be used as an evaluation of the degree of jeopardy, based on which the change of a robot dynamic can be arranged accordingly in planning an optimal velocity profile. Therefore, the use of collision-trend index can resolve the drawback of the hierarchical time-space strategies [13] where rotation of a robot is not allowed, and it finds a non-optimal collision free path only.

### IV. Speed Alteration Strategy

The proposed method is to tackle the problem of speed alteration, where the robots should move at their programmed speeds along a defined path. However, when a possible collision is predicted, the slave robot with less priority should change its speed to yield to the dominant one (master) unless any alteration of the slave's speed will fail. The slave robot searches an optimal position of the end-effector within the range where the maximal speed of the tip can reach in each servoing time. An optimal speed profile for the slave robot arm can be generated by minimizing an objective function which is defined as

$$
f_{obj} = \sum_j \sum_i P_j(t_i) + \sum_j (P_j(t_i) - P_j(t_i-1)) + \left| \frac{v - v_{preprog}}{v_{preprog}} \right|
$$

(9)

where $P_j(t_i)$ is the collision-trend index between links of the counterparts at sample time $t_i$; $v$ is the selected speed; $v_{preprog}$ is the preprogrammed speed. The second term at the right side of Eq. (9) is designed to force the tip of the slave robot to move at the velocity which can reduce the sum of collision-trend index mostly. The third item, which is normalized to the range of $[0, 1]$, is the difference between the selected speed and the preprogrammed speed. This factor is applied to generating a speed profile where the accumulative variation between the selected speed and the preprogrammed speed is minimized as possible while the slave robot arm moves along its trajectory. The objective function for speed profile generation is actually a higher order function of spatial distances between the links since it is a function of the collision-trend index. Consequently, the kinematics effect has been
taken into account in sense while the speed profile is being generated. Since a higher order interpolation function can generate a smooth trajectory, it should be able to avoid an abrupt change in dynamics. The algorithm is illustrated in Chart 1.

Chart 1. Flowchart of the algorithm, where $v_l$ is the velocity of robot 1, which is the slave robot, and should yield to robot 2, which is the master robot, if potential collision is predicted; $V_{max}$ is the maximum velocity of the robot.

Instead of applying the configuration space method to transferring the Cartesian coordinate system into the joint coordinate system, the proposed method implements collision detection directly in the spatial domain. Through introducing jeopardizing collision–trend index, the regions where induce collisions are clearly specified. Although an ellipsoid may not represent the shape of a robot’s link precisely, it provides a much restrict safe envelope to grant a collision-free trajectory. The proposed method differs from the configuration space method which needs to define all forbidden regions and plan a trajectory in the free space. It examines only two robots’ configurations while the end-effectors are moving along trajectories. Therefore, the proposed method can save much time and is more practicable for on-line applications than the configuration space method.

V. EXPERIMENTS

In order to examine the performance of the proposed speed alternation strategy for collision-free trajectory planning, two Motoman robots made by YASUKAWAI Co., namely Robot 1 and Robot 2, working in an overlapped working envelope are set up. As shown in Fig. 3, it is assumed that the two robots stand away from each other so that their link 1 and link 2 may not collide with their counterparts. Therefore, only link 3 and link 4 are involved in collision detection; i.e., only four ($2 \times 2$) ellipsoids should be transformed and tested in the collision detection process. In other words, four collision-trend indices associated with links 3 and 4 of the two robots should be evaluated at each sampling time. The maximal speed of the end-effector of each Motoman robot is set as 40 cm/sec. Initially, both robots stand in the same pose, then move to the other pose simultaneously along a preprogrammed trajectory, and go back to the original pose finally. Since the end-effectors of the two robots are moving at the same constant speed, 20 cm/sec, therefore, the joint angle profiles of the two robots for the round-way case are all the same during the motion and shows in Fig. 4. In this case, collisions between the links will occur if both robots move along the preprogrammed trajectory. The collision-trend index and the collision message, generated by the proposed collision detection algorithm, between links of the two robots at each sampling time are illustrated in Fig. 5 and Fig. 6 respectively. The $p_{ij}$ in Fig. 5 represents the evaluated collision-trend index between the link $i$ of the master robot and the link $j$ of the slave one. Larger $p_{ij}$ implies that links are closer and may induce collision. The $k_{ij}$ in Fig. 6 is the value of $E_{ij}(x^*,y^*,z^*)$ related to the master robot’s link $i$ and the slave one’s link $j$. If collision message is smaller than 1, collision occurs. Let’s have a view at the profiles of collision-trend index or collision message, both them show that collisions occur at sampling times 33, 34, 35, 65, 66 and 67 while the two robots move along their own preplanned path at the preprogrammed speed. Therefore, it needs to apply some trajectory planning algorithm to avoid this undesired situation.
In the second experiment, the slave robot is controlled to alter the speed by means of the proposed algorithm in order to avoid collisions. The master robot’s end-effector moves at the same preprogrammed speed as in the previous experiment unless the alteration of the slave robot’s speed fails to avoid the occurrence of collision. To decide an alternative speed for the tip of the slave robot, an objective function Eq. (9) is used and the maximal collision-trend index among the four links is selected as the first term of the objective function. Meanwhile, if the change of the collision-trend index in two consecutive times remains small as possible, the profile can be generated more smoothly. The resultant collision message and the collision-trend index under the control algorithm are shown in Fig. 7 and 8 respectively. Fig. 7 shows that the collision messages at the sampling time 33, 34, 35, 65, 66 and 67 are larger than 1.0. This situation means that collisions between the two robots are successfully removed by the proposed control algorithm. The speed profiles of the two robots’ tips for the round-way case are shown in Fig. 9. It is observable that the slave robot decelerates and accelerates to avoid colliding with the master one.

VI. CONCLUSION
The paper presents a systematic method to solve the collision-free trajectory planning problem for multi-robot systems. Based on the requirements of waste volume minimization, simple mathematical description, and efficient geometric representation, the links of a robot are modeled as ellipsoids. It is instructive in dealing with spatial distance using the collision-trend index. Through coordinate transformations and scaling operations, the problem of collision detection between two ellipsoids can be reduced to determine whether a point falls inside or outside a transformed ellipsoid. Unlike the polyhedral modeling approaches, which evaluate the spatial configuration by measuring a family of polytopes for an object, the proposed approach provides an analytic method, instead of a numerical method, to evaluate collision sensitivity between the object of interest and obstacles. Therefore, the proposed approach is more suitable for on-line applications. Meanwhile, in comparison with the superquadric artificial potential function approach where only one obstacle is allowed to be located in the workspace [6], the proposed method based on the use of collision-trend index can allow more number of objects in the working space.

The proposed approach is implemented in an environment where the on-going paths of robots are pre-defined but the speed is adaptable. Under the control of the proposed algorithm, the master robot moves at a constant speed while the slave robot moves at the optimal speed based on the defined objective function unless any alternation of the slave robot's speed is in vain. The collision-trend index is defined based on the similar geometric representation between the ellipsoids and the Gaussian distribution. It is not only a significant measure index for the objective function in searching for the optimal resolution but also able to improve the computation efficiency in collision-avoidance problems. Since the function of collision-trend index is actually a higher order function of spatial distances between the links, therefore, the kinematics effects has been taken into account accordingly while the trajectory is being generated. Since a higher order interpolation function can generate a smoother trajectory, it should be able to avoid an abrupt change in dynamics based on collision-trend index.

However, to deal with the problem of Head-on or San-Diego type collision [14], the dimension of the problem domain should be expanded from the speed alteration method to
path alteration. Searching for the feasible solutions for path alteration also needs an objective function as an index. The proposed collision-trend index still may play an instructive role, analogous to the potential magnitude does in the potential field method, in searching an optimal trajectory. Therefore, the future work should be conducted to study of path alternation problem for cooperative robots.

VII. APPENDIX

Let's consider the ellipsoid containing the link $i$ of robot 1 in Fig. 1, it can be represented as:

$$X_i = A_i^0, \text{Rot}(z, \theta_i)T_{cl}X_i$$

(A.1)

If the base coordinate system of robot 1 is the reference (world) coordinate system, then link $j$ of robot 2 can be expressed as:

$$X_j = T_{12}A_j^0, \text{Rot}(z, \theta_j)T_{12}X_j$$

(A.2)

where $x_i'$ and $x_j'$ can be written in the following polar vector form:

$$X_i' = S_iX_i = \begin{bmatrix} r_{xi} \cos \phi_i \cos \psi_i \\ r_{yi} \cos \phi_i \sin \psi_i \\ r_{zi} \sin \phi_i \\ 1 \end{bmatrix},$$

$$-\pi / 2 \leq \phi_i \leq \pi / 2$$

$$-\pi \leq \psi_i \leq \pi$$

$$X_j' = S_jX_j = \begin{bmatrix} r_{xj} \cos \phi_j \cos \psi_j \\ r_{yj} \cos \phi_j \sin \psi_j \\ r_{zj} \sin \phi_j \\ 1 \end{bmatrix},$$

$$-\pi / 2 \leq \phi_j \leq \pi / 2$$

$$-\pi \leq \psi_j \leq \pi$$

Here, $S_i$ and $S_j$ are the scaling matrices which transform a sphere into an ellipsoid containing a link with the radii, $r_x, r_y,$ and $r_z$. $T_{12}$ in Eq. (A.2) is the matrix of transformation from robot 1's to robot 2's base frame:

$$T_{12} = \text{Trans}(A_{12}, 0, 0) \text{Rot}(z, \theta_{12})$$

where $A_{12}$ is the distance between robot 1 and robot 2 and $\theta_{12}$ is the change of the rotational angle from robot 1 to robot 2 in the $z$ axis.

Thus, the ellipsoid containing link $i$ can be represented as in the following equations:

$$X_i = A_i^0, \text{Rot}(z, \theta_i)T_{cl}X_i = T_{12}A_{12}^0, \text{Rot}(z, \theta_j)T_{12}S_jX_j,$$

$$\Rightarrow x_i^* = S_{ij}^{-1}T_{ij}^{-1} \text{Rot}^{-1}(z, \theta_i)A_{12}^0, \text{Rot}(z, \theta_j)T_{12}X_j,$$

$$\Rightarrow x_i^* = S_jHx_j^*, \quad (A.3)$$

where $x_i^* = (x_i^*, y_i^*, z_i^*)$ is the coordinate of link $i$ of robot 1 w. r. t. the coordinate frame of robot 2's link $j$. $H$ is the transformation as shown below:
\[ H = T_{i}^{-1} \text{Rot}(z, \theta_{j})A_{i}^{-1}T_{i}^{-1}A_{i}\text{Rot}(z, \theta_{j})T_{i} \]

\[
\begin{bmatrix}
 n'' & o'' & a'' & p'' \\
 0 & 0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
 n'' & o'' & a'' & p'' \\
 0 & 0 & 0 & 1
\end{bmatrix}
\]

Therefore, Eq. (A.3) can be rewritten as:

\[
\begin{bmatrix}
 x_{i} \\
y_{i} \\
z_{i}
\end{bmatrix} = H^{-1}S_{j}X''_{i} \tag{A.4}
\]

Since \( X''_{i} \) is the position vector of an ellipsoid with radii of \( r_{xi}, r_{yi}, \) and \( r_{zi} \) w. r. t. the center of the link \( i \), it can be represented as:

\[
E(x'', y'', z'') = \frac{x_{i}^{2}}{r_{xi}^{2}} + \frac{y_{i}^{2}}{r_{yi}^{2}} + \frac{z_{i}^{2}}{r_{zi}^{2}} = 1 \tag{A.5}
\]

After the above transformations, the collision detection problem between robot 1’s link \( i \) and robot 2’s link \( j \) is simplified to one between an ellipsoid and an unit sphere. Meanwhile, to prevent the ellipsoids from coming to contact the transformed ellipsoid of Eq. (A.5) should be expanded slightly. Furthermore, the unit sphere of link \( j \) of robot 2 should be reduced to a point so that the detection problem can be further simplified. A straightforward way is to expand the radii of the other ellipsoids by magnitude of the radius of this sphere. It should be pointed out that there is topological difference between the ellipsoid expanded one unit length along the radiuses and the one expanded the same length radially. The former is used in this work since it is still a standard form of ellipsoids. It is worth noting that in shrinking the ellipsoids, the scaling operation may cause one or more of the radii of the smaller ellipsoid to approach zero (degenerate in dimension) if the larger ellipsoid is scaled to a unit sphere. This situation always occurs when the sizes of these two links differ very much. To eliminate this error, the larger ellipsoid should be scaled to a sphere with a radius of \( r_{d} \) which is its smallest radius, instead of a unit sphere. For instant, if \( r_{ij} \) is the smallest radius of the ellipsoid, then the scaling matrix \( S_{j} \) should be modified to:

\[
S_{j} = \begin{bmatrix}
 d_{ij} & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & d_{ij} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

This means that the three scaling factors of the initial \( S_{j} \) are normalized to \( r_{d}/j, 1 \) and \( r_{d}/j \). Therefore, the equation of the ellipsoid after expanding by a scale of \( r_{d} \) along the radii is:

\[
E(x'', y'', z'') = \frac{x_{i}^{2}}{r_{xi}^{2}} + \frac{y_{i}^{2}}{r_{yi}^{2}} + \frac{z_{i}^{2}}{r_{zi}^{2}} = 1 \tag{A.6}
\]

or

\[
E(x'', y'', z'') = A'x_{i}^{2} + B'y_{i}^{2} + C'z_{i}^{2} + K = 0 ,
\]

where

\[
A' = \frac{-K}{A}, \quad B' = \frac{-K}{B}, \quad C' = \frac{-K}{C'};
\]

\[
x_{i} = u_{x}(x''_{i} - x_{c}) + u_{y}(y''_{i} - y_{c}) + u_{z}(z''_{i} - z_{c});
\]

\[
y_{i} = v_{x}(y''_{i} - y_{c}) + v_{y}(y''_{i} - y_{c}) + v_{z}(z''_{i} - z_{c});
\]

\[
z_{i} = w_{x}(z''_{i} - z_{c}) + w_{y}(y''_{i} - y_{c}) + w_{z}(z''_{i} - z_{c}).
\]

Here, \( A', B', \) and \( C' \) are the eigenvalues of matrix \( M \); \( u=(u_{x}, u_{y}, u_{z}), v=(v_{x}, v_{y}, v_{z}), \) and \( w=(w_{x}, w_{y}, w_{z}) \) are their corresponding normalized eigenvectors. \( M \) and \( K \) are defined as follows:

\[
M = \begin{bmatrix}
\frac{(n_{x}''r_{xij})^{2}}{r_{xi}^{2}} + \frac{(o_{x}''r_{xij})^{2}}{r_{yi}^{2}} + \frac{(a_{x}''r_{xij})^{2}}{r_{zi}^{2}} \\
\frac{(n_{y}''r_{xij})^{2}}{r_{xi}^{2}} + \frac{(o_{y}''r_{yij})^{2}}{r_{yi}^{2}} + \frac{(a_{y}''r_{yij})^{2}}{r_{zi}^{2}} \\
\frac{(n_{z}''r_{xij})^{2}}{r_{xi}^{2}} + \frac{(o_{z}''r_{yij})^{2}}{r_{yi}^{2}} + \frac{(a_{z}''r_{yij})^{2}}{r_{zi}^{2}}
\end{bmatrix}
\]
where \((r_{xi}, r_{yi}, r_{zi})\) is the radii of the ellipsoid containing the link \(i\). \(K = E(x_c, y_c, z_c)\), where \((x_c, y_c, z_c)\) is the solution of the following three equations:

\[
\frac{\partial}{\partial x} E(x', y', z') = 0,
\frac{\partial}{\partial y} E(x', y', z') = 0, \text{ and }
\frac{\partial}{\partial z} E(x', y', z') = 0.
\]

REFERENCES


