

On-Line Collision-Avoidance Trajectory Planning of Two Planar Robots Based on Geometric Modeling

KAO-SHING HWANG AND MING-DAR TSAI

*Electrical Engineering Department
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.*

Superquadric modeling technology applied to the description of various shapes of robot links is discussed in the paper. Through a series of homogeneous transformations between the robots' links, which have been modeled as a specified superquadric form, ellipse, the complexity of the collision detection problem can be reduced when evaluating the relative location between a transformed ellipse and a point. To cope with the analytic method, a spatial measurement between two objects called the collision probability, instead of the Cartesian distance, is defined by projecting the geometric similarity of an elliptic equation onto the contour of the Gaussian distribution. This probability represents the sensitivity of collision occurrence. The proposed algorithm is implemented in an environment where the on-going paths of the robots are fixed, but their speeds are alternative. Therefore, an objective function is defined to refine the optimal points along the paths for each sampling time, which, hence, forms the collision-free trajectories for the cooperative robots.

Keywords: robotics, collision avoidance, geometric modeling, coordinate transformation, trajectory planning, collision probability

1. INTRODUCTION

In general, multiple robots working together can improve the working capability. By assigning to individual robots their own subtasks to be performed in parallel, the robots can accomplish a task in less time than a single robot can. In this way, the production time can be significantly reduced. However, in arranging several robots to execute a task cooperatively introduces collisions between the links of the robots.

In general, the problems of collision avoidance and path planning can be divided into two sub-problems as proposed by Kant and Zucker [1]. Path planning in a dynamic environment can be decomposed into two sub-problems:

- (1) planning the path to avoid static obstacles, and
- (2) planning velocities along the path to avoid moving obstacles.

This decomposition results in a significant reduction in the complexity of the problem [2]. However, there are still some limitations to this approach. For instance, it does not permit path alterations but does velocity alterations. Therefore, under certain conditions, it cannot give a solution even though one may exist. Analyzing the collision conditions, five collision types are found for any two straight-line moving objects when collision occurs in

Received August 15, 1996; accepted September 25, 1997.
Communicated by Zen Chen.

general. The possible collision types are described below:

Let α be the angle measured from the intersection of any two straight-line paths.

- (a) Acute collision: collision of two objects with $0^\circ < \alpha < 90^\circ$;
- (b) Obtuse collision: collision of two objects with $90^\circ < \alpha < 180^\circ$;
- (c) Perpendicular collision: collision of two objects with $\alpha = 90^\circ$;
- (d) San-Diego collision: collision of two objects when they are moving along the same path and in the same direction, i.e., $\alpha = 0^\circ$, and the speed of the object behind is higher than the speed of the object ahead; and
- (e) Head-On collision: collision of two objects when they are moving along the same path but in opposite directions, i.e., $\alpha = 180^\circ$.

Collision types (a), (b) and (c) can be avoided by simply changing the velocities of the objects. For instance, in an assembly cell, robots are required to slow down or wait to pick up the correct parts from part feeders to ensure correct sequencing; or in a military situation, a battleship can escape from a torpedo attack coming from its side by speeding up or slowing down. Collision types (d) and (e) result when two moving objects have parts of paths which coincide. For instance, one airplane is taxiing onto the runway to prepare for takeoff, and another airplane is trying to land on the same runway at the same time going in the opposite direction. We cannot avoid this collision by only adjusting their speeds; the assigned path (runway) of the landing airplane must be changed or a waiting time for the taxiing airplane should be included in order to avoid collision. The conduct of the two collision types was introduced in our related research [3].

Many methods have been proposed in recent years to solve the problems of collision-avoidance. Chang and his colleagues have proposed a simple time delay method avoid collisions between two robot arms [4]. In their work, links of robots were approximated geometrically using polyhedra. The danger of collisions between two robots is expressed by a distance function associated with the robots in a working space. The collision map scheme in the form of a 2D Traveling Length v.s. Sampling Time (TLVST) graph can describe collisions between two 3D robots effectively. In a similar method in which the complex 3D problems are changed to simple 2D ones, Wu *et. al.* proposed that links of robots in 3D can be simplified to a 2D Space/Time graph [5]. Robots can move with the proper velocity to avoid potential collisions with obstacles or with other robots by constructing an optimal path on the Space/Time graph. Similarly, an optimization approach was proposed to direct the acceleration or deceleration strategy for robot motion using the defined collision probability [2].

A unified approach for robot motion planning with dynamic polyhedral obstacles was presented by Shih and his colleagues [7]. In this method, 2-D and 3-D objects can be represented by a set of polytopes. The optimal path can be obtained by evaluating a family of feasible collision-free trajectories between the initial and the goal polytopes. These trajectories actually are the routes connecting the selected polytopes under speed and time constraints. This approach differs from methods using the fixed path stated above. However, the problem of searching for an optimal collision-free path becomes very complex when the space dimension or the number of obstacles is increased. For the collision-avoidance problem of multiple manipulators in cooperative operation, a notable approach which treats two robot arms as a single redundant system was presented [8]. In addition, a collision map representing the intersection of robots is developed in a two-parameter space in the

proposed method. A trajectory is, thus, planned by searching for a collision-free path on the collision map and minimizing a defined objective function simultaneously.

Since whether or not a the developed method can model the geometric shape of a robot correctly and simply or not is an important factor for the efficiency of a collision detection algorithm, this is still an on-going research topic. In general, most of the approaches model the shapes of robots as a set of geometric primitives or even complex polytopes and polyhedras [4~7]. Superquadric modeling has been applied as primitives for shape representation and recovery in computer graphics and computer vision. A fitting function was introduced by Solina and Bajcsy [9] for recovery of compact volumetric models of single-part objects. The recovery of non-deformed superquadric models, as well as links of robots, can be easily accomplished using these methods. Since an instantiation of a superquadric can be expressed in mathematical form, the spatial relation between robot links can be derived easily in an analytic manner if the links are modeled by superquadrics. Some researchers have applied this the technique to construct 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 [10].

This paper introduces a novel collision-avoidance strategy to direct robots moving along pre-programmed paths in an adaptive speed. By means of the proposed approach, the complexity of the collision detection problem can be reduced when the relative location between a transformed ellipse and a point is evaluated through a series of translation, rotation, and scaling transformations along the robots' links. Since the detection strategy is an analytic method, it can be applied to the on-line collision avoidance problem. This paper is organized as follows. Section 2 discusses the geometrical relationship between a specific superquadric-ellipse and the Gaussian distribution. Section 3 demonstrates the procedures that transform a link model into a point coupled with deformation of the others as a set of geometrical equivalent transformations. Once the collision probability of the spatial point on the motion paths has been evaluated, the control strategy can be applied to drive the robot along a safe path at each sampling time. Section 4 presents the outcome of a simulation of a situation where two robots which would have collided with each other in the middle of motion can be resolved using the proposed strategy.

2. MEASUREMENT OF COLLISION OCCURRENCE

2.1 Geometric Modeling of Robot Links

A robot can be modeled by a proper superquadric equation if there is a sufficient number of given points on its links. Since an ellipse can be generalized to the superquadric form (n-ellipse), a robot link can be modeled as an ellipse by applying the fitting technique of superquadric modeling. In this way, the representation of a link can be described in a simple mathematical equation.

The general equation (also called the inside-outside function) of an ellipse is

$$F(x, y) : \frac{x^2}{rx^2} + \frac{y^2}{ry^2} = 1. \quad (1)$$

To examine whether a point, (x_o, y_o) , resides in an ellipse or not, $F(x, y)$ should be

calculated at that point. If $F(x_o, y_o) > 1$, the point is located outside the ellipse; otherwise the point is inside the boundary. A best fitting function derived from least-squares minimization is adopted in this work to model a link using a superquadric ellipse. The axes of the ellipse are the controllable parameters when an ellipse is fitted to a link. The best fitting function can be written as

$$\min \sum_{i=1}^N [R(x_i, y_i, r_x, r_y)]^2, \quad (2)$$

$$R(x_i, y_i, r_x, r_y) = \sqrt{r_x \cdot r_y} (1 - F),$$

where $x_i, y_i, i = 1, 2 \dots N$ are the points on the modeled object; r_x and r_y are the axes of the modeling ellipse.

To solve the optimization equation, the parameters r_x and r_y are searched in the limited range:

$$\frac{W}{2} \leq r_x \leq \frac{3W}{4} \quad \Delta r_x = 0.01$$

$$\frac{L}{2} \leq r_y \leq \frac{3L}{4} \quad \Delta r_y = 0.01,$$

where W and L are the width and length of the modeled object.

2.2 Collision Probability

Once the robot links are modeled geometrically as some equivalent ellipses, it is essential to apply a quantity measurement to define the spatial relationship between these links. Therefore, a measurement, called the collision probability, instead of the linear Cartesian distance, is defined by projecting an ellipse equation onto the contour of the Gaussian distribution, based on the geometric similarity between an ellipse and the contour of the distribution. A two dimension general Gaussian distribution can be expressed as

$$p(\underline{x}) = \frac{1}{2\pi} |\Sigma|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\underline{x}-\underline{\mu})^T \Sigma^{-1}(\underline{x}-\underline{\mu})]. \quad (3)$$

It is observed that the exponential portion of a Gaussian distribution, $(\underline{x}-\underline{\mu})^T \Sigma^{-1}(\underline{x}-\underline{\mu})$, actually is in the shape of a rotated ellipse if the covariance matrix is symmetric:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}, \quad \sigma_{12} = \sigma_{21} \neq 0 \quad (a \text{ symmetric matrix}),$$

$$\underset{\text{diagonalize}}{\Sigma} \sim D, \quad \Sigma = P D P^{-1}, \quad D = \begin{bmatrix} r_x^2 & 0 \\ 0 & r_y^2 \end{bmatrix},$$

where Σ is a 2 by 2 covariance matrix; D and P are the 2 by 2 diagonal eigenvalue matrix and eigenvector matrix, respectively. Since matrix D is a diagonal matrix after the transformation, the determinant of Σ is the same as the determinant of D. Thus, $|\Sigma| = |D| = r_x^2 r_y^2$.

Eq. (3) can be rewritten as

$$p(\underline{x}) = \frac{1}{2\pi r_x r_y} \exp\left[-\frac{1}{2} X'^T D^{-1} X'\right] \quad (4)$$

$$X' = P(\underline{x} - \underline{\mu}),$$

where X' is actually the position vector of an ellipse by transformation P. Furthermore, Eq.(3) can be represented in the form of

$$X'^T D^{-1} X' = (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) = -2\ln(2\pi r_x r_y p(\underline{x})) = k. \quad (5)$$

After expansion, the left hand side of Eq.(5) can be rewritten in the polynomial form:

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = k. \quad (6)$$

This equation is, in fact, a transformed elliptic equation. It is reasonable to assume that this ellipse is equivalent to one modeling a robot link. In other words, Eq.(6) projects the distribution onto an ellipse, which models a robot link.

As for the transformations of an ellipse from the world coordinate system to the link's should be further illustrated, Let r_{xi} and r_{yi} be the major and minor axes of the modeled ellipse, (x_i, y_i) be the tip of link $i-1$, and q be the rotational angle of the link i . The sequential transformation can be expressed as follows:

$$\underline{X} = T, R_i T \underline{X} \quad (7)$$

$$\text{where } T_i = \begin{bmatrix} 1 & 0 & x_i \\ 0 & 1 & y_i \\ 0 & 0 & 1 \end{bmatrix}, R_i = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 & r_{xi} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\underline{x} = \begin{bmatrix} r_{xi} \cos \phi \\ r_{yi} \sin \phi \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad 0 < \phi \leq 2\pi.$$

$\underline{X} = [X \ Y \ 1]'$ is the position vector of the rotated ellipse for link i ; \underline{x} is the position vector of an elliptic equation in the parametric form. Fig.1 illustrates the sequence of an ellipse transformed from the base coordinate system into the joint frame to align the modeled link. Eq. (7) can be rewritten in reversed sequence as

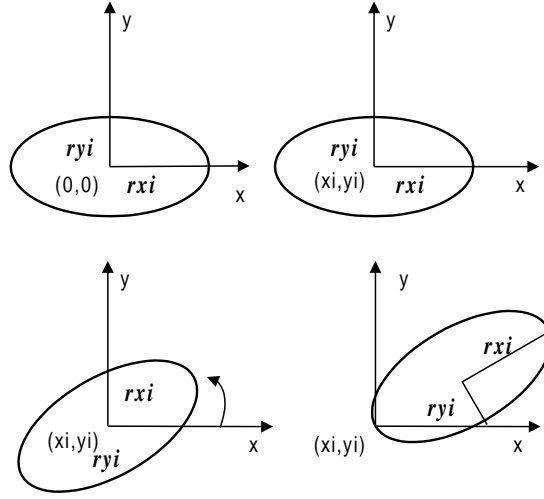


Fig. 1. The sequential transformation of a link modeled with an ellipse.

$$\underline{x} = T^{-1} R_i^{-1} T_i^{-1} \underline{X}$$

$$\Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} (X - x_i)\cos\theta + (Y - y_i)\sin\theta - r_{xi} \\ -(X - x_i)\sin\theta + (Y - y_i)\cos\theta \\ 1 \end{bmatrix}.$$

The position vector always obeys the standard elliptic equation as

$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} = 1, \text{ so } x \text{ can be expressed as :}$$

$$\frac{((X - x_i)\cos\theta + (Y - y_i)\sin\theta - r_{xi})^2}{r_{xi}^2} + \frac{(-(X - x_i)\sin\theta + (Y - y_i)\cos\theta)^2}{r_{yi}^2} = 1. \quad (8)$$

Also, the equation can be arranged in polynomial form:

$$AX^2 + BY^2 + CXY + DX + EY + F = 1,$$

$$\text{where } A = \frac{\cos^2\theta}{r_{xi}^2} + \frac{\sin^2\theta}{r_{yi}^2}; B = \frac{\sin^2\theta}{r_{xi}^2} + \frac{\cos^2\theta}{r_{yi}^2}; C = 2\sin\theta\cos\theta\left(\frac{1}{r_{xi}^2} - \frac{1}{r_{yi}^2}\right); D = -2x_iA$$

$$- y_iC - \frac{2r_{xi}\cos\theta}{r_{xi}^2}; E = -2y_iB - x_iC - \frac{2r_{xi}\sin\theta}{r_{xi}^2}; F = x_i^2A + x_iy_iC + y_i^2B +$$

$$\frac{2r_{xi}(x_i\cos\theta + y_i\sin\theta)}{r_{xi}^2}.$$

Eq. (6) is equivalent to the equation above when $k = 1$. In other words, it can be said that the ellipse equation of the modeled link i is equivalent to the exponential term of the Gaussian distribution if $k = 1$. Therefore, the collision probability is defined as follows:

$$p_c(\underline{X}) = \frac{1}{2\pi r_{xi} r_{yi}} \exp\left[-\frac{1}{2}(A X^2 + B Y^2 + CXY + DX + EY + F)\right] \quad (9)$$

$$k = AX^2 + BY^2 + CXY + DX + EY + F.$$

$p_c(\underline{X})$ is defined as the collision probability at point $\underline{X} = (X, Y)$. Since the values of r_{xi} , r_{yi} and the rotational angle q of link i are deterministic, the parameters A~F in Eq.(9) can be evaluated, and the collision probability at \underline{X} can be rewritten as

$$p_c(\underline{X}) = \frac{1}{2\pi r_{xi} r_{yi}} \exp\left[-\frac{k}{2}\right]. \quad (10)$$

A collision probability is redefined as the jeopardizing collision probability P_{jc} when $k = 1$.

Therefore,

$$p_{jc} = \frac{1}{2\pi r_{xi} r_{yi}} \exp\left[-\frac{1}{2}\right] \cong \frac{0.09653}{r_{xi} r_{yi}}. \quad (11)$$

As a result, if a point is given in the workspace, the collision probability associated with the point can be derived directly from Eq. (10). The occurrence of a collision can be easily detected by testing the following condition: if $P_{jc} \leq P_c(\underline{X})$, then a collision occurs.

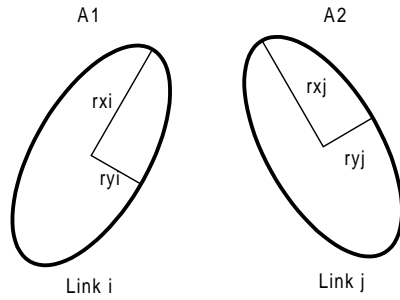
3. FAST COLLISION-DETECTION APPROACH

Collision detection between a point and a link in a workspace is accomplished by examining the value of the collision probability by modeling the link using a simple ellipse. However, if multiple-link robots are involved, the complexity of collision detection increases.

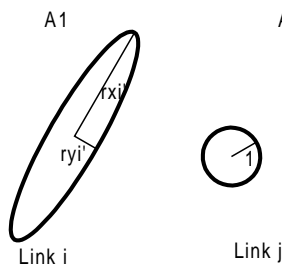
3.1 Relative Transformation Between an Ellipse and a Point

An algorithm has been developed to reduce the checking whether two ellipses intersect with each other to the problem of a point and a transformed ellipse. The basic idea is to deform an ellipse into a unit circle coupled with transformation of the other ellipse. The circle is further scaled down to a small "point". Fig. 2 shows the geometrical effects of these sequential transformations.

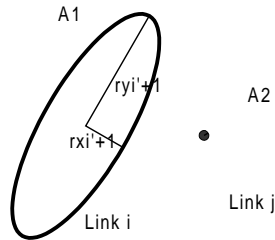
It should be noted that the ellipse of interest is scaled down into a circle along its major and minor axes. In order to maintain topological equivalence, the ellipses representing the other links of the robots should be scaled based on the same magnitude and direction used for of interest. Therefore, the transformations between the links must be derived. The configuration simulated in this paper is depicted in Fig. 3.



(a) Two ellipses represent the links before deformation.



(b) The equivalent topology after deformation of link j into a circle.



(c) The equivalent topology after the circle is scaled down to one unit.

Fig. 2. The transformation steps in the collision detection approach.

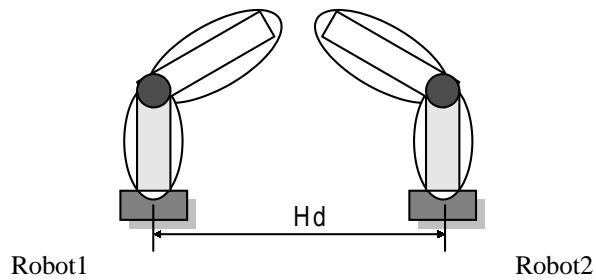


Fig. 3. Two two-joint robots are modeled geometrically using four ellipses.

The positions of Robot1's link i and Robot2's link j , \underline{X}_1 , and \underline{X}_2 can be expressed in the form of Eq. (7) as

$$\underline{X}_1 = T_{i-1}^0 Rot(\theta_i) T_i \underline{x}_1 \quad (12)$$

$$\underline{X}_2 = T_{12} T_{j-1}^0 Rot(\theta_j) T_j \underline{x}_2, \quad (13)$$

where $\underline{X}_1 = (X_1, Y_1)$ and $\underline{X}_2 = (X_2, Y_2)$ are the position vectors for Robot1's link i and Robot2's link j ; T_{k-1}^0 is the transformation matrix from base to link $k-1$ for Robot1 or Robot2; T_{12} is the transformation matrix between Robot1's base and Robot2's base; $Rot(\theta_k)$ is the rotational matrix of the robot's joint k by an angle of θ_k ; $k = i$ or j , and

$$\underline{x}_k = \begin{bmatrix} r_{xk} \cos \phi_l \\ r_{yk} \sin \phi_l \\ 1 \end{bmatrix} = \begin{bmatrix} r_{xk} & 0 & 0 \\ 0 & r_{yk} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi_l \\ \sin \phi_l \\ 1 \end{bmatrix} = S_k \underline{x}; \quad \begin{matrix} 0 \leq \phi_l \leq 2\pi \\ l = 1 \text{ or } 2, \end{matrix}$$

where r_{xk} , r_{yk} are the radii of the modeled ellipse of link k for Robot1 or Robot2; S_k is the scaling matrix and \underline{x} is the position vector of a unit circle on the origin of the world coordinate. On the other hand, Eq. (13) can be rewritten as

$$\underline{X}_2 = T_{12} T_{j-1}^0 Rot(\theta_j) T_j S_j \underline{x}.$$

In Eq. (12) \underline{x}_1 is the transformed coordinate of \underline{X}_1 obtained by transformation of $T_{i-1}^0 Rot(\theta_i) T_i$. Alternatively, \underline{X}_1 can be expressed as the transformation of $T_{12} T_{j-1}^0 Rot(\theta_j) T_j S_j$ as

$$\underline{X}_1 = T_{12} T_{j-1}^0 Rot(\theta_j) T_j S_j \underline{x}_1', \quad (14)$$

where \underline{x}_1' is the transformed coordinate of \underline{X}_1 through the transformation

$$T_{12} T_{j-1}^0 Rot(\theta_j) T_j S_j.$$

In other words, if the world coordinate frame is transformed by $T_{12} T_{j-1}^0 Rot(\theta_j) T_j S_j$ into a new reference coordinate frame, the coordinates of \underline{x}_1' and \underline{x} , which are based on the reference coordinate system, represent \underline{X}_1 and \underline{X}_2 as shown in Fig. 4.

From Eq. (12) and Eq. (14), \underline{x}_1' can be rewritten as:

$$\underline{x}_1' = S_j^{-1} H \underline{x}_1, \quad (15)$$

where

$$H = T_j^{-1} Rot^{-1}(\theta_j) T_{j-1}^0 T_{12}^{-1} T_{i-1}^0 Rot(\theta_i) T_i$$

$$T_j^{-1} = \begin{bmatrix} 1 & 0 & -r_{xj} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Rot^{-1}(\theta_j) = \begin{bmatrix} \cos \theta_j & -\sin \theta_j & 0 \\ \sin \theta_j & \cos \theta_j & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

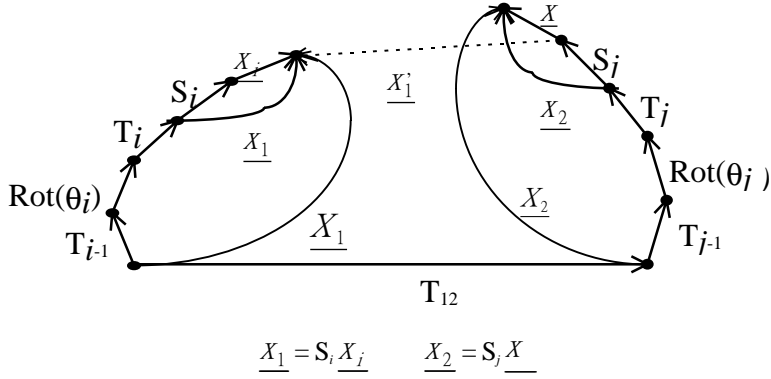


Fig.4. The relative transformations between the links of two cooperative robots.

$$T_{j-1}^0{}^{-1} = \begin{bmatrix} 1 & 0 & a_j \\ 0 & 1 & b_j \\ 0 & 0 & 1 \end{bmatrix}; \begin{cases} j=1 \rightarrow a_j = b_j = 0; \\ j=2 \rightarrow \begin{cases} a_j = L1 \cos \theta_{11}, \\ b_j = L1 \sin \theta_{11}, \end{cases} \end{cases}$$

$$T_{i-1}^0{}^{-1} = \begin{bmatrix} 1 & 0 & a_i \\ 0 & 1 & b_i \\ 0 & 0 & 1 \end{bmatrix}; \begin{cases} i=1 \rightarrow a_i = b_i = 0; \\ i=2 \rightarrow \begin{cases} a_i = L1 \cos \theta_{21}, \\ b_i = L1 \sin \theta_{21}, \end{cases} \end{cases}$$

where θ_{11} and θ_{21} are the rotational angles of link1 for Robot1 and Robot2; L1 is the length of link1:

$$T_i = \begin{bmatrix} 1 & 0 & r_{xi} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{Rot}(\theta_i) = \begin{bmatrix} \cos \theta & -\sin \theta_i & 0 \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{12}^{-1} = \begin{bmatrix} 1 & 0 & -Ad \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ (Ad is the distance between the two robots)}$$

$$\text{let } H = \begin{bmatrix} n_x & o_x & p_x \\ n_y & o_y & p_y \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the components of the transformation matrix H can be calculated as follows:

$$\begin{aligned} n_x &= \cos(\theta_i - \theta_j), & o_x &= -\sin(\theta_i - \theta_j) \\ n_y &= \sin(\theta_i - \theta_j), & o_y &= \cos(\theta_i - \theta_j) \\ p_x &= r_{xi}n_x + (a_i - Ad) \cos \theta_j + b_i \sin \theta_j + a_j \cos \theta_j + b_j \sin \theta_j - r_{xj} \\ p_y &= r_{xi}n_y - (a_i - Ad) \sin \theta_j + b_i \cos \theta_j - a_j \sin \theta_j + b_j \cos \theta_j. \end{aligned}$$

On the other hand, Eq. (15) can be rewritten as

$$\begin{aligned} \underline{x}_1 &= H^{-1} S_j \underline{x}_1' \text{ let } \underline{x}_1' = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} & (16) \\ \Rightarrow \underline{x}_1 &= \begin{bmatrix} r_{xi} \cos \phi \\ r_{yi} \sin \phi \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & n_y & -n \cdot p \\ o_x & o_y & -o \cdot p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{xj} & 0 & 0 \\ 0 & r_{yj} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \Rightarrow \underline{x}_1 &= \begin{bmatrix} r_{xi} \cos \phi \\ r_{yi} \sin \phi \\ 1 \end{bmatrix} = \begin{bmatrix} n_x r_{xj} x + n_y r_{yj} y - n^T \cdot p \\ o_x r_{xj} x + o_y r_{yj} y - o^T \cdot p \\ 1 \end{bmatrix}. \end{aligned}$$

Because the position vector \underline{x}_1 obeys the standard elliptic equation, it can be represented as

$$\frac{(n_x r_{xj} x + n_y r_{yj} y - (n_x p_x + n_y p_y))^2}{r_{xi}^2} + \frac{(o_x r_{xj} x + o_y r_{yj} y - (o_x p_x + o_y p_y))^2}{r_{yi}^2} = 1. \quad (17)$$

Eq. (17) is the new scaled ellipse of Robot1's link i as shown in Fig. 2(b).

From the derivation, the collision problem between two ellipses can be simplified as one between a unit circle and a scaled ellipse. However, it is still not easy to check whether they will collide with each other or not. A straightforward approach to the problem is to expand the radius of the new ellipse by a unit length and shrink the unit circle to a "point" relatively as Fig. 2(c) shown. In this way, the problem of collision detection can be reduced to that of checking whether a point falls inside or outside an enlarged ellipse. It is worth pointing out that there is topological difference between an ellipse expanded by one unit length along the radii and the one expanded by the same length radially as shown in Fig. 5.

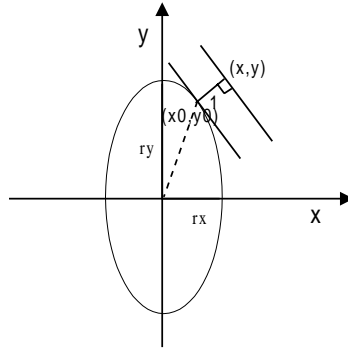


Fig. 5. The contour of an ellipse enlarged by one unit radially.

The parametric representation of an ellipse which is expanded by one unit radially is

$$\begin{aligned} x &= x_0 + r_y \cos \theta / \sqrt{r_x^2 \sin^2 \theta + r_y^2 \cos^2 \theta}, \\ y &= y_0 + r_x \sin \theta / \sqrt{r_x^2 \sin^2 \theta + r_y^2 \cos^2 \theta}, \end{aligned} \quad 0 < \theta \leq 2\pi;$$

$$x_0 = r_x \cos \theta,$$

$$y_0 = r_y \sin \theta. \quad (18)$$

However, the equation of an ellipse expanded radially by one unit in the parametric form is as follows:

$$x = (r_x + 1) \cos \theta,$$

$$y = (r_y + 1) \sin \theta. \quad (19)$$

Eq. (19) is still the standard form of an ellipse while Eq. (18) is not. Some researchers have used polygons to approximate the expanded ellipse radially and have shown that the error resulting from the topological difference is so small that it can be ignored, especially when the expanded magnitude is just one unit. Therefore, Eq. (19) is adopted in this work, instead of Eq. (18), to keep the characteristic of the standard ellipse after expansion. Furthermore, the parametric form of an ellipse can be manipulated more easily in the process of modeling the link, so Eq. (19) is a simpler and more convenient representation.

Furthermore, Eq. (17), which represents the transformed ellipse coupled with circle reduction deformation, is not the standard form of an elliptic equation. It can be observed that the axes $r_x i$ and $r_y i$ are not actually the new radii after scaling. In fact, the standard form of a transformed elliptic equation is essential when the equation representing an ellipse expanded by one unit is written by just increasing the new radii one unit. Therefore, the next step is to arrange Eq. (17) into the standard form of an elliptic equation as follows:

$$f(x, y) = Ax^2 + By^2 + Cxy + Dx + Ey + F = 0, \quad (20)$$

where

$$A = \frac{n_x^2 r_{xj}^2}{r_{xi}^2} + \frac{o_x^2 r_{xj}^2}{r_{yi}^2}; B = \frac{n_y^2 r_{yj}^2}{r_{xi}^2} + \frac{o_y^2 r_{yj}^2}{r_{yi}^2}; C = \frac{2n_x n_y r_{xj} r_{yj}}{r_{xi}^2} + \frac{2o_x o_y r_{xj} r_{yj}}{r_{yi}^2},$$

$$D = \frac{-2(n_x p_x + n_y p_y) n_x r_{xj}}{r_{xi}^2} + \frac{-2(o_x p_x + o_y p_y) o_x r_{xj}}{r_{yi}^2},$$

$$E = \frac{-2(n_x p_x + n_y p_y) n_y r_{yj}}{r_{xi}^2} + \frac{-2(o_x p_x + o_y p_y) o_y r_{yj}}{r_{yi}^2},$$

$$F = \frac{(n_x p_x + n_y p_y)^2}{r_{xi}^2} + \frac{(o_x p_x + o_y p_y)^2}{r_{yi}^2} - 1.$$

The function can be transformed into the standard form in two steps: in the first step, let $x = x' + h$ and $y = y' + k$, where (h, k) is the solution of the two equations as

$$\begin{cases} \frac{df(x, y)}{dx} = 2Ax + Cy + D, \\ \frac{df(x, y)}{dy} = 2By + Cx + E. \end{cases} \quad (21)$$

Eq. (21) can be derived from the differential of $f(x, y)$ in x and y individually. Then, $f(x, y)$ changes to the following form:

$$f(x, y) = Ax'^2 + By'^2 + Cx'y' + K = 0, \quad (22)$$

where

$$K = f(h, k) = Ah^2 + Bk^2 + Chk + Dh + Ek + F.$$

Let the coordinates of x' and y' change to x'' and y'' by rotating an angle of θ :

$$x' = \cos \theta x'' - \sin \theta y'',$$

$$y' = \sin \theta x'' + \cos \theta y''.$$

Then, the equation can be rewritten as follows by replacing x' and y' by x'' and y'' :

$$f(x, y) = A'x''^2 + B'y''^2 + C'x''y'' + K = 0, \quad (23)$$

where

$$A' = A \cos^2 \theta + B \sin^2 \theta + C \sin \theta \cos \theta,$$

$$B' = A \sin^2 \theta + B \cos^2 \theta - C \sin \theta \cos \theta,$$

$$C' = -2A \sin \theta \cos \theta + 2B \sin \theta \cos \theta + C(\cos^2 \theta - \sin^2 \theta).$$

If $C' = 0$, then $\theta = \frac{1}{2} \tan^{-1}\left(\frac{C}{B-A}\right)$, and $f(x, y)$ becomes

$$f(x, y) = A'x'^2 + B'y'^2 + K = 0. \quad (24)$$

Then, the standard equation of a transformed ellipse can be obtained from Eq. (24) as

$$\frac{x''^2}{r_{xi}'^2} + \frac{y''^2}{r_{yi}'^2} = 1, \quad (25)$$

where r_{xi}' and r_{yi}' are the real radii of the new ellipse as follows:

$$r_{xi}' = \sqrt{\frac{-K}{A'}} \quad x'' = \cos \theta (x-h) + \sin \theta (y-k)$$

$$r_{yi}' = \sqrt{\frac{-K}{B'}} \quad y'' = -\sin \theta (x-h) + \cos \theta (y-k)$$

Hence, the equation which expands the radii of an ellipse by one unit as mentioned previously can be represented as

$$\frac{x''^2}{(r_{xi}'+1)^2} + \frac{y''^2}{(r_{yi}'+1)^2} = 1. \quad (26)$$

In this way, a very complex problem of collision detection for multiple robots can be reduced to a simpler one in which we only need to check if a point resides inside or outside a deformed ellipse. In summary, the proposed fast collision detection algorithm is represented in the following functions:

$$E(x, y) = \frac{(\cos \theta (x-h) + \sin \theta (y-k))^2}{(r_{xi}'+1)^2} + \frac{(-\sin \theta (x-h) + \cos \theta (y-k))^2}{(r_{yi}'+1)^2}.$$

The detection of collision between the links on two different robots can be summarized as

$$E_{ij}(x, y) \Big|_{x=0, y=0} \leq 1 \Rightarrow \text{collision}, \quad (27)$$

where $E_{ij}(x, y)$ means the transformed elliptic function between the i th link of Robot1 and the j th link of Robot2.

Since $E_{ij}(x, y)$ is just an elliptic equation, k in Eq. (10) can be replaced by Eq. (27). In this way the collision probability between the i th link of Robot1 and link j th of Robot2 can be evaluated by

$$p_c(x, y) = \frac{1}{2\pi(r_{xi}' + 1)(r_{yi}' + 1)} \exp\left[-\frac{E_{ij}(x, y)}{2}\right] \quad (28)$$

and

$$P_{jc} \leq P_c(x, y) \Big|_{x=0, y=0} \Rightarrow \text{collision},$$

$$\text{where } P_{jc} \cong \frac{0.09653}{(r_{xi}' + 1)(r_{yi}' + 1)}.$$

3.2. Control Strategy

The hands of two robots were assumed to move repeatedly along two intersecting paths, as shown in Fig. 6, in the simulation. A strategy based on the defined collision probability controls the moving speed of one robot to prevent collision. This strategy tries to keep the master robot moving at a speed which is as constant as possible and the slave robot moving in proper alternative speed.

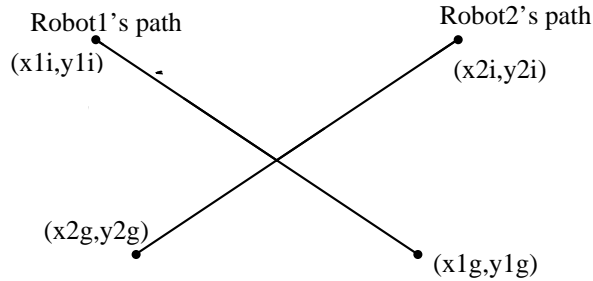


Fig. 6. The moving paths of two two-joint robots, where Robot1 moves from (x1i, y1i) to (x1g, y1g), and Robot2 moves from (x2i, y2i) to (x2g, y2g).

Robot2 (the master) is considered to be more important than Robot1 (the slave), so the former moves at a constant speed of $V2$, which is derived from the equations given below:

$$T2 = \frac{\sqrt{(x2g - x2i)^2 + (y2g - y2i)^2}}{V2}, \quad Ns = \frac{T2}{st}$$

$$\Delta S2 = \frac{\sqrt{(x2g - x2i)^2 + (y2g - y2i)^2}}{Ns}, \quad (29)$$

where $T2$ is the total elapsed time; Ns is number of sampling times; st is the unit of sampling time; $\Delta S2$ is the average moving distance at a sampling time.

If the maximum speed of Robot1 moving along the path is $Vmax$, the optimal moving distance at each sampling time can be determined as

$$optimal \ \Delta S1 = \frac{\sqrt{(x1g-x1i)^2 + (y1g-y1i)^2}}{Ns},$$

$$optimal \ V1 = \frac{\Delta S1}{st},$$

$$\Delta S \ max = V \ max \cdot st, \tag{30}$$

where $\Delta S1$ and $V1$ are the optimal average moving distance and velocity, respectively; $\Delta S \ max$ is the possible maximal moving distance at a sampling time.

If the robots are moving on fixed paths, Robot1 should change speed by searching for the optimal point in the region of possible positions under the constraint of the maximum speed $Vmax$ while Robot2 moves at a constant speed. The objective function for searching for the next optimal position to which Robot 1 should move is defined as

$$f_{obj} = k1 \frac{|V(t) - V1op|}{V1op} + k2 P1(t) + k3 \frac{dP1(t)}{dt}, \tag{31}$$

where $V(t)$ and $V1op$ are the selected and optimal velocity of Robot1 at time t ; $P1(t)$ is the collision probability of Robot1's link 2 and Robot2's link2 at time t ; $\frac{dP1(t)}{dt} = P1(t) - P1(t - 1)$; $k1$, $k2$ and $k3$ are weighting factors of the function.

The search algorithm utilizes the objective function, $\min(fobj)$, to evaluate which points which are collision-free after fast collision detection on the moving path of Robot1 at the sampling time. Fig. 7 shows the next possible position of the two robots.

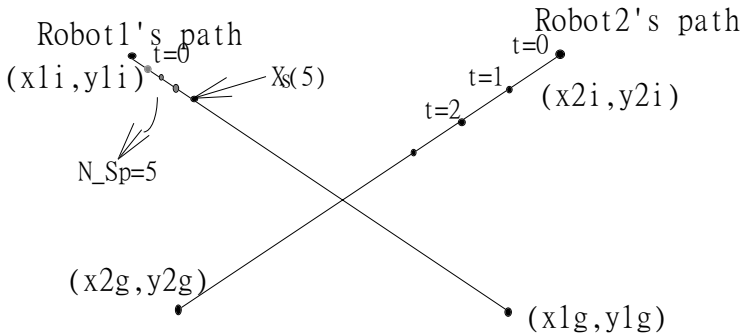


Fig. 7. The moving paths of the robots at the some sampling time, where N_Sp is the sampling index during the sampling time, and $Xs(5)$ is the maximum moving position which Robot1 can reach at the sampling time.

The next possible position of Robot1 can be represented as follows:

$$Xs(i) = x(t-1) + \frac{\Delta S \max_x}{N_Sp} \cdot i,$$

$$Ys(i) = y(t-1) + \left(\frac{y1g - y1i}{x1g - x1i}\right) \cdot (Xs(i) - x(t-1)),$$

$$i = 0, 1, \dots, N_Sp, \quad (32)$$

where $\Delta S \max_x$ is the maximum moving distance of Robot1 in the x-axis, and $(x(t-1), y(t-1))$ is previous coordinate of Robot1. N_Sp is the sample number during the sampling time.

4. SIMULATION

To examine the proposed algorithm, two two-joint robots were used in the experiments. Table 1 shows the kinematic data of the robots. The path was arranged such that the joint angular profiles of one robot could resemble those of its counterpart with an angular offset in the original programmed speeds. The simulative parameters for the two robots are illustrated as Table 2.

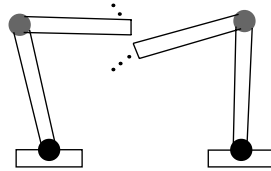
Table 1. The kinematic data of the robots.

Length of robot1 and robot2's link1	L1 = 80 cm
Length of robot1 and robot2's link2	L2 = 70 cm
Width of robot1 and robot2's link1	W1 = 10 cm
Width of robot1 and robot2's link2	W2 = 10 cm
Distance between robot1 and robot2	Ad = 120 cm
the major axis of ellipses modeled from link1 of robot1 and robot2	rx11 = 45.001 cm rx21 = 45.001 cm
the major axis of ellipses modeled from link2 of robot1 and robot2	rx12 = 40.005 cm rx22 = 40.005 cm
the minor axis of ellipses modeled from link1 of robot1 and robot2	ry11 = 40.005 cm ry21 = 40.005 cm
the minor axis of ellipses modeled from link2 of robot1 and robot2	ry12 = 6.635 cm ry22 = 6.635 cm

Figs. 8 and 9 show the results of the collision-avoidance simulation. Originally, the robots could collide with each other if they proceeded at the pre-programmed speeds, as shown in Fig. 8(a). However, due to the resolution of the proposed strategy, the slave robot changed speed to prevent collision. The resolved result is shown in Fig. 8(b~c). Fig. 9(a-b) shows the motion profiles of joint 1 and joint 2 of the two robots. As noted above, each joint

Table 2. The simulative parameters for the robots.

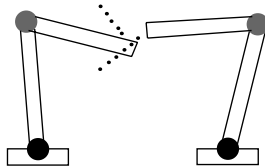
initial position of robot 1's path	$(x1i, y1i) = (40, 90)$
initial position of robot 2's path	$(x2i, y2i) = (40, 90)$
goal position of robot 1's path	$(x1g, y1g) = (80, 40)$
goal position of robot 2's path	$(x2g, y2g) = (90, 90)$
weighting factor	$k1 = k2 = 1 \quad k3 = 10$
maximum velocity of the robot	$V_{max} = 1.5$
optimal velocity of the robot	$V_{op} = 1.0$
sampling time	$st = 0.05$
sample number during the sampling time	$N_Sp = 10$
Total number of samples	$Ns = 20$



the tip position of robot 1 $(51.200000, 76.000000)$
the tip position of robot 2 $(55.000000, 62.000000)$

rotational angle of robot1 : $(103.5752257, -1.444831)$
rotational angle of robot2 : $(88.096087, 194.863193)$

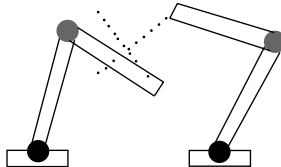
(a) The robots will collide with each other if they continue moving at the programmed speed.



the tip position of robot 1 $(61.600000, 63.000000)$
the tip position of robot 2 $(70.000000, 74.000000)$

rotational angle of robot1 : $(94.565129, -13.841183)$
rotational angle of robot2 : $(75.586576, 182.851224)$

(b) The robots resolve the collision under the strategy control.

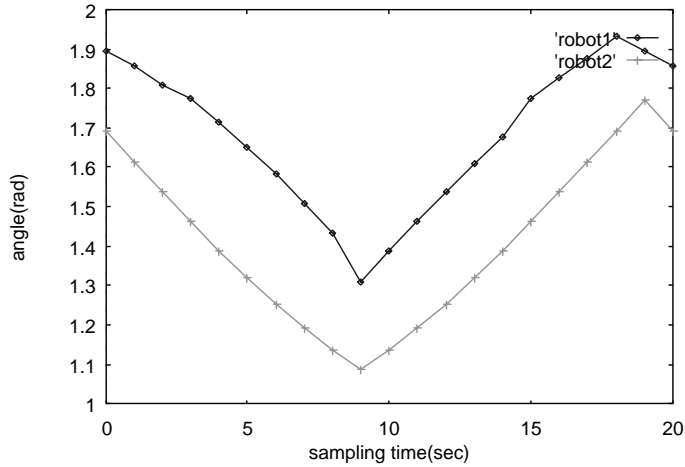


the tip position of robot 1 $(80.000000, 40.000000)$
the tip position of robot 2 $(90.000000, 90.000000)$

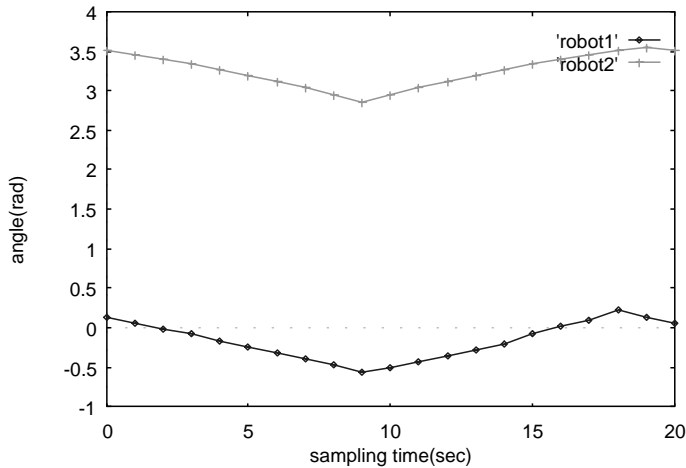
rotational angle of robot1 : $(74.972217, -32.163906)$
rotational angle of robot2 : $(62.203606, 164.054012)$

(c) The robots can reach the target points as schedule.

Fig. 8. The configuration of the robots at the different sampling times.



(a) angle of joint1



(b) angle of joint2

Fig. 9. (a)-(b) Joint angle profiles of the robots at each sampling time.

of the robots was arranged to have the same motion profile for the purpose of comparison. Due to the on-line control of the strategy, some portions of the joint motion profiles had certain deviations, for instant at time 7-9 sec. for joint 1 and time 17- 19 for joint2.

5. DISCUSSION AND CONCLUSION

In this paper, collision-free trajectory planning between two moving manipulators has been studied. The proposed method is based on the technique of modeling the links of robots geometrically by means of ellipses. The complexity of the collision detection problem is thereby reduced since the possibility of collision occurrence between the objects

is expressed by a relative measurement, called the collision probability, between a transformed ellipse and a point through the coordinate transformations, and by geometric deformation. Based on the defined collision probability, basic strategies for dealing with collision-avoidance trajectory planning have been investigated. Meanwhile, an objective function based on the probability has been introduced to search for the feasible trajectory points along the programmed paths at each sampling interval.

Unlike the polyhedral modeling approaches, which evaluate the spatial configuration by measuring a family of polytypes for an object, the proposed approach provides an analytic method, instead of a numerical method, to evaluate the collision sensitivity between the object of interest and obstacles. Therefore, this approach is more suitable for on-line applications. Meanwhile, in comparison with the superquadric artificial potential function approach, the collision probability approach allows a larger number of objects in the working space. However, the proposed approach is implemented in an environment where the ongoing paths of the robots are fixed but the speed is adaptable. The other problem that still needs to be studied is that of collision situations which require path modification, such as the Head-On and San-Diego types. These types of collisions cannot be avoided by only adjusting the speeds because of the overlapping navigation path segment. An alternative path has to be considered to prevent collision. In some cases, a waiting time should be put in between the motions of multiple objects to avoid collision. Selecting feasible paths and inserting waiting time are the main factors involved in constructing cost functions for performance evaluation. Therefore, strategies for selecting paths under different circumstances are another research topic in the study of general collision-free motion among moving objects or machines.

Although the main theme of this paper has been to develop collision-free motion among two moving robots, the motion planning strategies for avoiding collisions are necessary to deal with problems of sequencing and scheduling. In other words, the proposed approach can also be applied to these problem domains, such as in aviation flight traffic control and assembly sequence planning. Aircraft Sequencing is an excellent demonstration of the applicability of the proposed method to plan motions among moving objects. Another application is planning of an efficient assembly sequence for an automated assembly center. In manufacturing, robots are often used to assemble many parts from part feeders or on production plates, such as x-y tables. In other words, the locations of different parts are predetermined based on product requirements. The problem is how to plan a collision-free trajectory for robots to use in assembling these parts. Furthermore, these locations of parts are the intermediate positions of a global path to be determined. The problem is how to visit these desired points efficiently under the collision-free requirement. Different arrangements for assembling a product result in a variety of assembling sequences. Finding the most efficient one under certain measures, such as collision probability, from all possible sequences is also an optimization problem. When there exists a large variety of part types, an efficient sequence can really increase productivity. This problem will be the focus of one of our future studies.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful suggestions for revising this paper. This work was supported by the National Science Council, Taiwan, under grant NSC 84-2212-E-194-003.

REFERENCES

1. K. Kant and S. W. Zucker, "Planning collision-free trajectories in time-varying environments. A two-level hierarchy," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1988, pp. 1644-1649.
2. N. C. Griswold and J. Eem, "Control for mobile robots in the presence of moving objects," *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 2, 1990, pp. 263-268.
3. K. S. Hwang, M. D. Tsai and M. Y. Ju, "A constant speed trajectory planning of two-link serpentine robot", *Journal of Automatic Control*, Vol. 4-1, 1996, pp. 13-21.
4. C. Chang, M. J. Chung and B. H. Lee, "Collision avoidance of two general robot manipulators by minimum delay time," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 3, 1994, pp. 674-699.
5. C. H. Wu, D. T. Lee, K. Freter and K. S. Hwang, "An automated collision avoidance motion planner among moving objects or machines," in *Proceedings of IEEE Conference on Decision and Control*, 1991, pp. 367-372.
6. B. J. Oommen and I. Reichstein, "On translating ellipses amidst elliptic obstacles," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1986, pp. 1755-1760.
7. C. L. Shih, T. Lee and W. A. Gruver, "An unified approach for robot motion planning with moving polyhedral obstacles," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 4, 1990, pp. 903-915.
8. C. L. Shih, J. P. Sadler and W. A. Gruver, "Collision avoidance for two SCARA robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1991, pp. 674-678.
9. F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, 1990, pp. 131-146.
10. R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, 1990, pp. 1423-1436.



Kao-Shing Hwang (黃國勝) is an associate professor in the Electrical Engineering Dept. at National Chung Cheng University, Taiwan. He received the B.S. Degree in Industrial Design from National Cheng Kung University, Taiwan, in 1981, and the M.M.E. and Ph.D. degrees in EECS from Northwestern University, Evanston, I.L., in 1989 and 1993, respectively. He was a design engineer at the SANYO Electric Co., Taipei, Taiwan, during 1983-1985. Between 1987 and 1988, he was a system programmer with the C & D Microsystem Co., Plastow, N. H. His job involved designing PC drivers and graphic animation. Since August 1993, he has been at National Chung Cheng University, Taiwan. He is also the director of the Information Management Division at the university computer center. His areas of interest are neural networks and learning control, robotic compliance, and collision avoidance.



Ming-Dar Tsai (蔡明達) was born in Taiwan in 1971. He received the B.S. degree in 1994 from Chinese Culture University, Taipei, and the M.S. degree in 1996 from National Chung-Cheng University, Chia-Yi, both in electrical engineering. In addition to the modeling and simulation of robotic collision avoidance, his research interests include control theories, neural fuzzy systems, genetic algorithms and machine learning.