

26

Planning of Collision-Free Paths for a Reconfigurable Dual Manipulator Equipped Mobile Robot

QING XUE

*Department of Computer and Information Science, Purdue University School of Science
at Indianapolis, Indianapolis, IN 46202, U.S.A.*

PHILIP C.-Y. SHEU

*Department of Electrical and Computer Engineering, University of California-Irvine, Irvine,
CA 92717, U.S.A.*

ANTHONY A. MACIEJEWSKI

School of Electrical Engineering, Purdue University, West Lafayette, IN 47907, U.S.A.

STANLEY Y. P. CHIEN

*Department of Electrical Engineering, Purdue University School of Engineering and Technology
at Indianapolis, Indianapolis, IN 46202, U.S.A.*

(Received 7 March 1995; in final form 2 November 1995)

Abstract. In this paper, we study the problem of finding a collision-free path for a mobile robot which possesses manipulators. The task of the robot is to carry a polygonal object from a starting point to a destination point in a possibly cluttered environment. In most of the existing research on robot path planning, a mobile robot is approximated by a fixed shape, i.e., a circle or a polygon. In our task planner, the robot is allowed to change configurations for avoiding collision. This path planner operates using two algorithms: the collision-free feasible configuration finding algorithm and the collision-free path finding algorithm. The collision-free feasible configuration finding algorithm finds all collision-free feasible configurations for the robot when the position of the carried object is given. The collision-free path finding algorithm generates some candidate paths first and then uses a graph search method to find a collision-free path from all the collision-free feasible configurations along the candidate paths. The proposed algorithms can deal with a cluttered environment and is guaranteed to find a solution if one exists.

Key words: reconfigurable robot, path planning, artificial intelligence.

1. Introduction

The problem of planning a collision-free path for a mobile robot has been studied for many years. Many approaches have been suggested to solve the problem. However, almost all existing approaches approximate a mobile robot by an object with a fixed shape. Lozano-Perez [1] modeled objects as polyhedra and developed the configuration space method to solve the collision-free path planning problem. Conceptually, the configuration space of a robot can be obtained by shrinking

the moving robot to a point and enlarging all the obstacles to the shape of the moving robot relative to a reference point. A shortest path of the robot can then be searched in the configuration space. This model has been further extended for different applications [2–5] with the consideration of reducing computational complexity. Schwartz and Sharir [6] represented robots and obstacles by polygons. Their algorithm makes use of a construct called a critical curve, which is a curve traced by a robot which moves in contact with an obstacle. The free space in the robot's work space is partitioned by these critical curves into many regions. A solution is searched in the connectivity graph formed by the regions. Kedem and Sharir [7] improved this result with the assumption that a robot is represented as a convex polygon. Brooks [8] also used polygons to represent robots and obstacles. He used generalized cones to represent free space and solved the problem by finding a path in a set of connected cones. Khatib [9] and Hwang and Ahuja [10] used the concept of artificial potential fields to represent objects. The repulsion force between a robot and obstacles is used to guide the robot to avoid collision with obstacles. However, the potential field method may not guarantee to find a collision free path even there exists one. Carriker et al. [11] proposed to use simulated annealing in robot path planning so that a collision-free path can always be found if it exists. Wong and Fu [12] described a hierarchical approach to plan robot paths in a three-dimensional work space.

In this paper, we study the problem of planning a collision-free path for a mobile robot which carries an object with two arms and maneuvers in a possibly cluttered two-dimensional environment. Since the relative position between the robot body and its arms and between the robot and the carried object can change, the robot and the carried object together are considered as a reconfigurable moving object. Generally speaking, a reconfigurable moving object can be defined as the composition of a set of serially connected rigid bodies whose relative positions can be changed.

Modeling a robot and a carried object together as a reconfigurable object has remarkable advantages over that as a fixed shape object in planning a collision-free path for the robot. During the motion of a reconfigurable object, collisions can be avoided by reconfiguring the shape of the object or by changing the path and orientation of the object. In other words, the reconfigurability of a robot can be utilized to avoid collision. This path planning problem is solved by two algorithms: the collision-free feasible configuration finding algorithm and the collision-free path finding algorithm. The collision-free feasible configuration finding algorithm finds all collision-free feasible configurations for the robot when the position and orientation of the carried object are given. This algorithm is an extension of the work published in [13]. The collision-free path finding algorithm generates some candidate paths first and then uses a graph search method to find a collision-free path from all the collision-free feasible configurations along the candidate paths. For simplicity, the dynamics of the moving object is not considered. Furthermore, it is assumed that all information about the environment

is available before the robot starts moving so that path planning is performed off-line.

The remainder of this paper is organized as follows: In Section 2, the problem is formalized. In Section 3, the collision-free feasible configuration finding algorithm is described. In Section 4, the collision-free path finding algorithm is explained. Section 5 concludes this paper.

2. Problem Formulation

In this paper, a mobile robot and its two manipulators are defined to be a composition of five serially connected rigid links a_1, a_2, \dots, a_5 in a two-dimensional plane (see Fig. 1). Each robot link a_i , where $1 \leq i \leq 5$, is represented as a rectangle with length l_i and width w_O if $i = 3$, or w_A if $i = 1, 2, 4, 5$. The centerlines of links a_i and a_{i+1} are assumed to be connected by a revolute joint which is located at point P_{i+1} . The end of a_1 that does not connect to a_2 is a free end named a . The end of a_5 that does not connect to a_4 is another free end named b . An object (see Fig. 1) which is carried by the robot is represented as a polygon with arbitrary vertices.

It is assumed that the robot carries an object with the two free ends a and b , and that links a_1 and a_5 can rotate about points a and b , respectively. Consequently, the grasp points a and b can be considered as two joints (the location of these

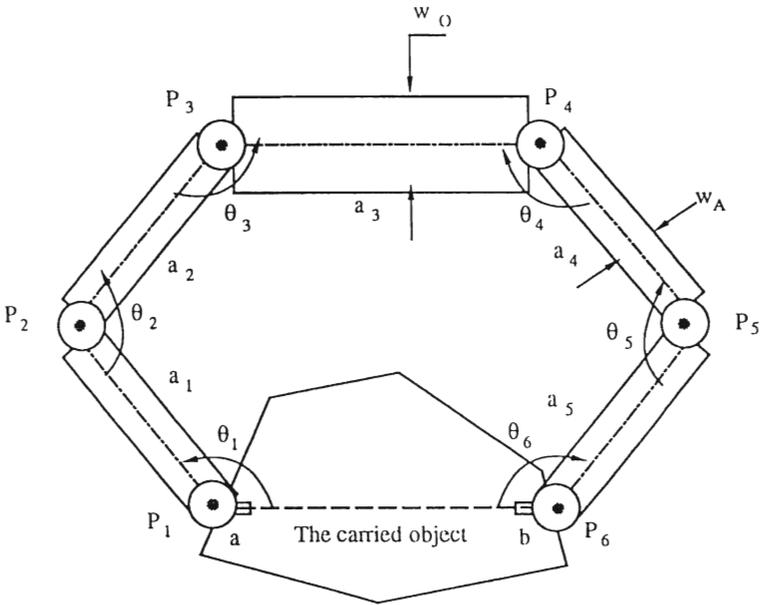


Figure 1. A reconfigurable robot and the carried object together form a 6-joint closed chain or a reconfigurable object.

joints are denoted as P_1 and P_6 , respectively). The angle of the joint located at P_i is denoted as θ_i . The direction of θ_i is defined in Figure 1, where the joint angles of one arm increase in the counter clockwise direction, while the joint angles of the other arm increase in the clockwise direction. For convenience, point a is chosen as the reference position for the carried object with the orientation defined as the angle between line ab and the x -axis of the world coordinate system. It is also assumed that the grasp points on the carried object are chosen arbitrarily and once they are chosen, they will not be changed along a path. Therefore, the robot and the carried object together can be modeled as a closed chain with six joints. This 6-joint closed chain is defined as a reconfigurable object.

In a two-dimensional environment, it is assumed that there are q stationary obstacles which are represented by polygons. Given a robot and an object as described above, our problem is to plan a path for each joint of the reconfigurable object so that the carried object can be moved from a given starting position and orientation to a given final position and orientation without colliding with the obstacles. To find the collision-free path for the reconfigurable object, the following constraints must be satisfied:

1. The Closed Chain Constraint: The closed chain must not be broken.
2. The Link Collision-Free Constraints:
 - (a) If two links are adjacent, one link is not allowed to pass through the other link.
 - (b) The non-adjacent links in the 6-joint closed chain are not allowed to intersect one another.
 - (c) The robot links must not intersect the carried object.
3. The Obstacle Collision-Free Constraint: The reconfigurable object must not intersect any obstacles in its motion.

If these constraints are satisfied, then there is no collision between the robot links, between the robot links and the carried object, and between the 6-joint closed chain and any obstacles in the environment. For convenience, the joint angles in the 6-joint closed chain are represented as a 6-tuple $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. An instance of the 6-tuple which satisfies the closed chain constraint is defined to be a *configuration* of the reconfigurable object. A configuration which satisfies the link collision-free constraints is defined to be a *feasible configuration* which is denoted as FC. Furthermore, a feasible configuration which does not collide with any obstacles is defined to be a *collision-free feasible configuration*, which is denoted as CFFC. It is assumed that the configurations of the reconfigurable object at the given initial and final positions and orientations of the carried object are collision-free and feasible.

The core of the path planner is composed of two algorithms. The first algorithm (called collision-free feasible configuration finding algorithm) finds all the collision-free feasible configurations of a reconfigurable object once the position and the orientation of the carried object are specified and fixed. This algo-

rithm is an extension of the collision-free feasible configuration finding algorithm described in [13]. In [13] the bases of two coordinate robot arms are fixed to the ground and the object to be handled by robot arms must be a rectangle. The second algorithm (called collision-free path finding algorithm) first selects the best candidate path based on an objective function. Subsequently, all the collision-free feasible configurations along the best candidate path are found with the first algorithm. Finally, a collision-free path is searched in these collision-free feasible configurations.

3. Collision-Free Feasible Configuration Finding Algorithm

Finding a collision-free feasible configuration of the reconfigurable object is meaningful only when a portion of the object is fixed relative to a set of reference coordinates. The fixed portion of the reconfigurable object is then used as a reference to the other moving portion of the object. For convenience, it is assumed that the position and the orientation of the carried object are fixed in the reference coordinate frame. The input to the collision-free feasible configuration finding algorithm includes the position and the orientation of the carried object and the specifications of the robot and the obstacles; the output of the algorithm includes all collision-free feasible configurations of the reconfigurable object or a message indicating that no collision-free feasible configuration exists.

3.1. STRATEGY

In order to find the collision-free feasible configurations (CFFC's) for a reconfigurable object, the following proposition which describes the relationship between different configurations will be employed:

PROPOSITION 1. *In order to move a reconfigurable object (as a closed chain described in Section 2) from one configuration to another while the position and the orientation of the carried object are fixed, the positions of at least three consecutive links need to be changed.*

It is easy to show that the above is true since any three consecutive links $a_i, a_{i\pm 1}$ and $a_{i\pm 2}$ in a closed chain have one internal degree of freedom if one end of a_i and one end of $a_{i\pm 2}$ which is not connected to $a_{i\pm 1}$ are fixed. For convenience, any link in which the position of one or both ends can be changed is called a *changeable link*. Since the position and the orientation of the carried object are the reference, by Proposition 1, any position change of a 6-joint closed chain is caused by the position change of a set of at least three consecutive changeable links. Therefore, each set of three consecutive changeable links a_i, a_{i+1}, a_{i+2} , where $1 \leq i \leq 3$, in a 6-joint closed chain is defined to be a *basic changeable unit*, $BCU(a_i, a_{i+1}, a_{i+2})$. Consequently, there are three basic changeable units $BCU(a_1, a_2, a_3)$, $BCU(a_2, a_3, a_4)$ and $BCU(a_3, a_4, a_5)$ in a 6-joint closed

chain. Since there is one degree of freedom of motion in a BCU, there are three degrees of freedom of motion in a 6-joint closed chain or a reconfigurable object.

Our strategy to find all the CFFC's for a reconfigurable object is to find all the CFFC's generated by a BCU for each quantized motion of the other two changeable links which are not in the BCU. For illustration, in the following discussion $BCU(a_2, a_3, a_4)$ is arbitrarily chosen and the motion of links a_1 and a_5 of the closed chain is quantized. Since in $BCU(a_2, a_3, a_4)$ there is only one degree of freedom, the parameter θ_5 is chosen to represent $BCU(a_2, a_3, a_4)$. We call the ranges of θ_5 in which the closed chain constraint is satisfied the motion range $MR(\theta_5)$ and call the subregions of the motion range $MR(\theta_5)$ in which the link collision-free constraint is satisfied the feasible region $FR(\theta_5)$. Furthermore, the subregions of the feasible region $FR(\theta_5)$ in which the obstacle collision-free constraint is satisfied is called the collision-free feasible regions or CFFR's. Consequently, our strategy to find all of the CFFC's in $BCU(a_2, a_3, a_4)$ is to one by one add the constraints given in Section 2 to the range of θ_5 .

In authors' previous paper [13], a 6-link closed chain with ground as one link was studied. If the carried polygonal object in Figure 1 is replaced by a line segment ab and P_1 and P_6 are fixed to the workspace, the reconfigurable object can be represented by a 6-link closed chain as described. When the carried object is fixed and only the robot arms are allowed to move, the difference between the mechanical structure of the reconfigurable object and the a 6-link closed chain described in [13] is the representation of link ab . This difference affect the moving ranges of different joints of the closed chain. Here we modify the method used in [13] to find the feasible region for a 6-joint closed chain. The required modifications is discussed in the following subsections.

3.2. FINDING ALL THE COLLISION-FREE FEASIBLE CONFIGURATIONS FOR $BCU(a_2, a_3, a_4)$

In this section, we shall find the feasible region of the 6-joint closed chain, i.e., a set of subregions of the motion range $MR(\theta_5)$ in which the link and obstacle collision-free constraints are satisfied. Since there are three changeable links in $BCU(a_2, a_3, a_4)$, to find the feasible regions of θ_5 , the following three steps can be followed:

1. Find the subregion of $MR(\theta_5)$ in which the corresponding positions of a_4 satisfy the link collision-free constraints.
2. Find the subregion of $MR(\theta_2)$ in which the corresponding positions of a_2 satisfy the link collision-free constraints; subsequently map this subregion of θ_2 to the ranges of θ_5 ; finally, intersect these mapped ranges and the ranges generated in Step 1.
3. Find the subregions of the ranges generated in Step 2 in which the corresponding positions of a_3 satisfy the link collision-free constraints.

Here we only need to study the case where $w_O = 0$ and $w_A = 0$. If $w_O \neq 0$ or $w_A \neq 0$, the results can be modified by shrinking the width of the robot links to zero and enlarging the size of the carried object [1].

3.2.1. *Finding the Subregion of $MR(\theta_5)$ in Which the Corresponding Positions of a_4 Satisfy the Link Collision-Free Constraints*

Based on the link collision-free constraints, link a_4 must not pass through the adjacent links, must not intersect non-adjacent links and must not intersect the carried object. The subregion of $MR(\theta_5)$ in which the corresponding positions of a_4 satisfy the link collision-free constraints is denoted as $R_{a_4}(\theta_5)$. To find $R_{a_4}(\theta_5)$, the following definitions need to be introduced first.

DEFINITION 1. If one end of the centerline of a changeable link a_i is fixed at a point P , then all the possible positions for the other end of the centerline of a_i form a circle with radius l_i and center P . This circle is called the *link position circle* of a_i and is denoted as $LPC_{a_i}(P)$ or simply LPC_{a_i} .

DEFINITION 2. If there exist two intersections between link position circle LPC_{a_4} of a_4 and the carried object, then they are called the *critical points* of a_4 . For fixed values of θ_1 and θ_6 , the one which corresponds to the smaller value of θ_5 is denoted as $cp_{a_4}^1$ and the other one which corresponds to the larger value of θ_5 is denoted as $cp_{a_4}^2$ (see Fig. 2). If LPC_{a_4} intersects one of the fixed links, this intersection is still called a critical point and is denoted as both $cp_{a_4}^1$ and $cp_{a_4}^2$. The critical points of a_2 are similarly defined.

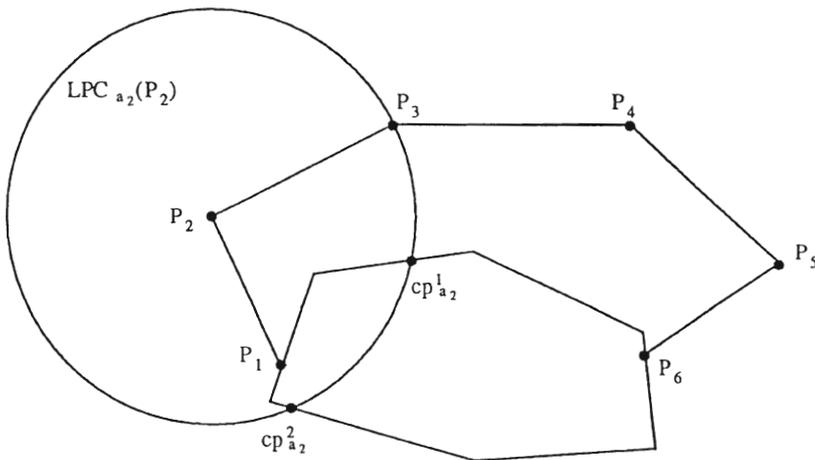
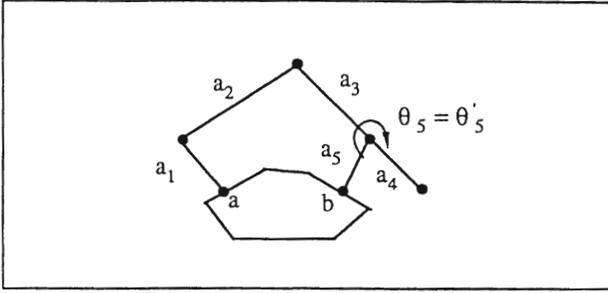
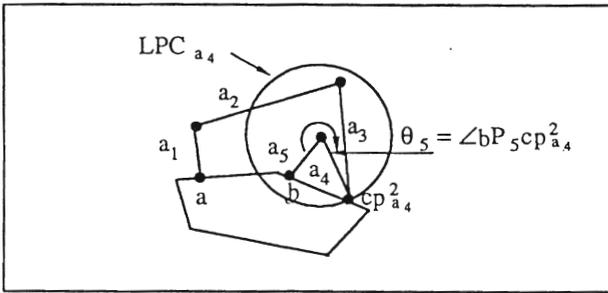


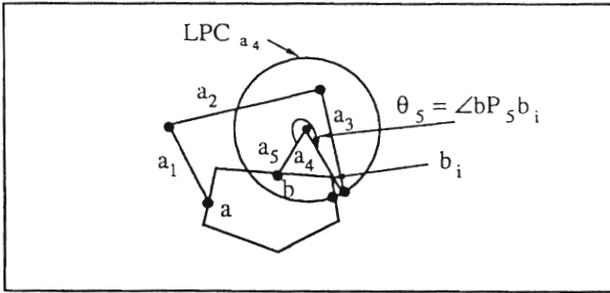
Figure 2. A graphical interpretation of the critical points of a_2 .



(a)



(b)



(c)

Figure 3. A graphical interpretation for $\theta_{5u} = \min(\angle bP_5cp_{a_4}^2, \angle bP_5b_i, \theta'_5)$.

In order to satisfy the link collision-free constraints, a_4 cannot pass through adjacent links a_3 and a_5 , and cannot intersect the carried object. To map these two requirements onto the range of θ_5 , the upper and lower bound, i.e., θ_{5u} and θ_{5l} , of the range of θ_5 in which a_4 satisfies the above two requirements must be found. For fixed values of θ_1 and θ_6 , θ_{5u} and θ_{5l} can be determined as follows:

1. Since a_4 cannot pass through a_3 , the upper bound of θ_5 must be less than the angle which makes a_4 and a_3 overlap and align, i.e., $\theta_4 = 0$ (see Fig. 3(a)). This angle of θ_5 is denoted as θ'_5 . Furthermore, a_4 cannot intersect the carried object so that θ_{5u} must also be less than $\angle bP_5CP_{a_5}^2$ (see Fig. 3(b)). θ_{5u} must be less than $\angle bP_5b_i$ where b_i is the corner of the carried object which is outside the polygon formed by the robot links and the line segment that connects grasp points a and b (see Fig. 3(c)). Consequently, the upper bound θ_{5u} is given by

$$\theta_{5u} = \min \left\{ \angle bP_5b_i, \theta'_5, \angle bP_5CP_{a_5}^2 \right\}. \quad (1)$$

2. Since a_4 cannot intersect the carried object, the lower bound of θ_5 is given by

$$\theta_{5l} = \max \left\{ \angle bP_5b_i, \angle bP_5CP_{a_5}^1 \right\}. \quad (2)$$

where b_i is the corner of the carried object which is inside the polygon formed by the robot links and the line segment that connects grasp points a and b .

Consequently, $R_{a_4}(\theta_5)$ should be equal to $MR(\theta_5) \cap [\theta_{5l}, \theta_{5u}]$.

3.2.2. Finding the Subregion of $MR(\theta_5)$ in Which the Corresponding Positions of a_4 and a_2 Satisfy the Link Collision-Free Constraints

The valid range of θ_2 in which a_2 satisfies the link collision-free constraints can be found in a manner similar to finding $R_{a_4}(\theta_5)$. This range of θ_2 is denoted as $R_{a_2}(\theta_2)$. Since θ_5 is chosen as a parameter to represent $BCU(a_2, a_3, a_4)$, $R_{a_2}(\theta_2)$ must be mapped to the ranges of θ_5 . These mapped ranges of θ_5 are denoted as $R_{a_2}(\theta_5)$. Since the procedure for finding the mapping of $R_{a_2}(\theta_2)$ to $R_{a_2}(\theta_5)$ is the same as described in [13], it is omitted here.

3.2.3. Finding the Valid Ranges of θ_5 in Which Three Changeable Links, a_2 , a_3 , a_4 Satisfy the Link Collision-Free Constraints

The subregions of $MR(\theta_5)$ in which all three changeable links (a_2, a_3, a_4) satisfy the link collision-free constraints are called the feasible range denoted as $FR(\theta_5)$. It is easy to show that if the carried object is a straight line, then $FR(\theta_5) = R_{a_2}(\theta_5) \cap R_{a_4}(\theta_5)$. If the carried object is a polygon and a corner of the polygon is inside the closed chain, link a_3 may collide with the carried object when $\theta_5 \in R_{a_2}(\theta_5) \cap R_{a_4}(\theta_5)$. Therefore, the subregions of $R_{a_2}(\theta_5) \cap R_{a_4}(\theta_5)$ in which the corresponding positions of a_3 do not intersect the carried object need to be found.

Consider the case where $[\theta_{s_1}, \theta_{s_2}]$ belongs to $R_{a_2}(\theta_5) \cap R_{a_4}(\theta_5)$. There should be some feasible configurations (FC's) that divide $[\theta_{s_1}, \theta_{s_2}]$ into several subregions such that a subregion at one side of the FC is a range of feasible configurations (or an FR), and the other side of the FC is not an FR. This kind of FC in a $[\theta_{s_1}, \theta_{s_2}]$ is called a *critical* FC. If all critical FC's can be obtained, then all

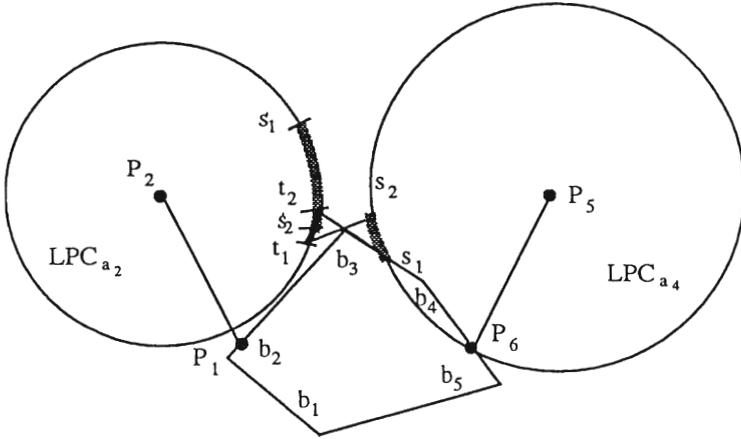


Figure 4. An illustration of the technique used for finding critical FC's which determine the boundaries of feasible regions.

FR's can be obtained as well. Subsequently, all FC's can be found. To find the critical FC's, the following proposition is introduced.

PROPOSITION 2. *If a 6-joint closed chain is in a critical FC, then a_3 must pass through one corner or an edge of the carried object which is inside the closed chain.*

It is easy to show the proposition is true since link a_3 is a line segment and the carried object is a convex polygon. By Proposition 2, the critical FC's can be found by finding the FC's in which the location of a_3 touches a corner or an edge of the carried object which is inside the closed chain. For convenience, in the following discussion, since each position of a_4 (or a_2) corresponds to a distinct value of θ_5 (or θ_2), it is assumed that if the changeable end of the centerline of a_4 (or a_2) is at a point P on LPC_{a_4} (or LPC_{a_2}), the corresponding θ_5 (or θ_2) is denoted as θ_{5P} (or θ_{2P}). In Figure 4, it is assumed that b_3 is a corner of the carried object and $[\theta_{5s_1}, \theta_{5s_2}] \subseteq R_{a_2}(\theta_5)$ is the mapping of a subregion of $[\theta_{2s'_1}, \theta_{2s'_2}] \subseteq R_{a_2}(\theta_2)$ on LPC_{a_2} . In order to avoid an exhaustive search, an algorithm to find the critical FC's with respect to vertex b_3 in $[\theta_{5s_1}, \theta_{5s_2}]$ is given below:

Algorithm for finding critical FC's

1. Calculate the possible intersection of line s_1b_3 with $LPC_{a_2}(P_2)$, denoted t_1 and the intersection of line b_3s_2 with $LPC_{a_2}(P_2)$, denoted t_2 .
 - (a) If both intersections do not exist or the intersection of the two arcs t_1t_2 and $s'_1s'_2$ is empty, then there is no critical FC with respect to the corner b_3 .

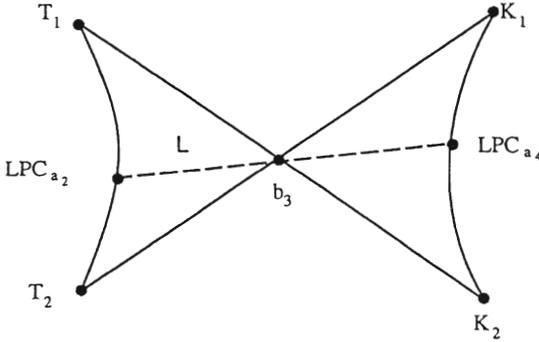


Figure 5. The region in which a critical FC may exist.

- (b) If both t_1 and t_2 exist and the intersection between arcs $s'_1s'_2$ and t_1t_2 is not empty, set $T_1T_2 = t_1t_2 \cap s'_1s'_2$.
- (c) If there exists only one t_i , $i = 1$ or 2 , the line $t_i s_i$ would separate the entire plane into two subplanes and separate LPC_{a_2} into two arcs.
 - (i) If there is no intersection between $s'_1s'_2$ and one arc of LPC_{a_2} which is in the same subplane with line $s_j b_3$, where $i = 1, 2$, $j = 1, 2$, $i \neq j$, then there is no critical FC with respect to the corner.
 - (ii) Otherwise, T_1T_2 is equal to $s'_1s'_2$.

2. Calculate the possible intersection of line T_1b_3 with $LPC_{a_4}(P_5)$, denoted K_1 and the intersection of line T_2b_3 with $LPC_{a_4}(P_5)$, denoted K_2 (see Fig. 5). If there is a line segment L with length l_3 which has one end on $[T_1, T_2]$ of LPC_{a_2} , one end on $[K_1, K_2]$ of LPC_{a_4} , and passes through b_3 , a feasible position of a_3 in the area shown in Figure 5 can be found (the line segment L can be obtained by solving a set of equations defined from the geometric properties of the area shown in Figure 5) [13].

After checking each corner of the carried object which is inside the closed chain, the edges of the carried object which are inside of the closed chain should also be checked by overlapping link a_3 with these edges. If there is a feasible configuration in which link a_3 and an edge of the carried object overlap, it is a critical FC. After finding all the critical FC's in $R_{a_2}(\theta_5) \cap R_{a_4}(\theta_5)$, all the FR's and hence all the FC's can be determined.

So far we have found the collision-free motion of the robot manipulators without considering the actual obstacles. If there are obstacles in the environment, then any collision with these obstacles should also be avoided. Our goal is to find the subregions of $FR(\theta_5)$ in which the obstacle collision-free constraint is satisfied, i.e., CFFR's. In fact, in the determination of collision-free motion range of θ_5 , the carried object is treated as the obstacle with respect to the

motion of robot manipulators. We can apply to the same procedure to find the collision-free motion range of θ_5 with respect to each actual obstacles. Then the superposition principle can be applied to determine the collision-free motion of the robot manipulators with respect to all obstacles.

4. Collision-Free Path Finding Algorithm

In the last section, an algorithm to find all the possible collision-free feasible configurations for a reconfigurable object at a given position and orientation was described. In this section, we shall use this algorithm to find a collision-free path which connects the starting point and the destination point in the free space for the object. In our approach, two levels of planning i.e., global path planning and local path planning, are used. In the global path planning, a construct called “free space valleys” is employed to represent the free space. Then based on a cost function which is assigned to each free space valley, the best candidate global path is generated. The local path planning focuses on planning the local path and motion for the reconfigurable object on the most cluttered parts of the global candidate path. First the local planner finds all the CFFR’s and the transitions among them in these hard regions along the global candidate path. Subsequently, a connection graph whose vertices are the CFFR’s of the reconfigurable object and whose edges are the transitions among these CFFR’s is built. Finally, the local path planner searches the connection graph for a collision-free path in the hard regions.

4.1. GLOBAL PATH PLANNING

Since there exist an infinite number of candidate paths in a free space, it is impossible to enumerate all of them. In our approach, we first classify all possible paths into a finite group of topologically distinct paths called free space valleys. Each group of free space valleys is examined for the possibility of yielding a collision-free path. The groups of free space valleys are sorted and stored in a queue according to a cost function that consists of the path length and the difficulty of motion. Subsequently, starting with the head of the queue, the candidate paths are classified into three regions: open regions, hard regions and dead regions. In an open region, the object can move freely without colliding with any obstacles. In a dead region, the object always collides with some obstacles. Our goal is to find a path in the hard regions in which the robot has to maneuver and/or reconfigure without colliding with any obstacles.

4.1.1. Finding Free Space Valleys in the Free Space

Let the robot’s work space contain q polygonal obstacles and let P be a point in the free space. If d_i^P denotes the minimum distance from point P to obstacle i , then the *minimum distance from P to all the obstacles* d_{\min}^P is given by,

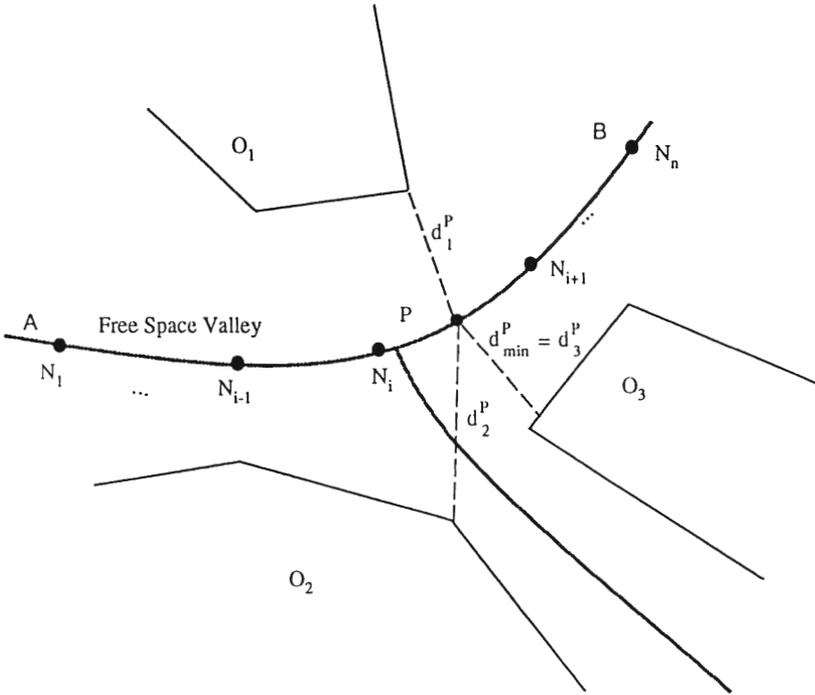


Figure 6. The definition of free space valleys and d_{\min}^P .

$$d_{\min}^P = \min_i d_i^P \tag{3}$$

where $i = 1, \dots, q$ (see Fig. 6). A circle centered at P with radius d_{\min}^P is called the reach circle of point P .

From the path planning point of view, the robot has a lower possibility of colliding with any obstacle if its path is along the centerline of the free space. Consequently the centerline of the free space is called a *free space valley* (see Fig. 6) which is found by the following procedure:

1. A reach circle is computed at the starting point of the reconfigurable object.
2. By definition, the points on the reach circle which have a local maximum for d_{\min}^P on the circumference of the reach circle should be the points on the free space valley. These points are found by following two steps:
 - (a) The set of points x_{ij} on the reach circle which have the same distance (denoted as d_{ij}) to a pair of obstacles O_i and O_j (see Fig. 7) are found.
 - (b) The distance between x_{ij} and every other obstacle in the workspace (see Fig. 7) is checked. If d_{ij} is smaller than one of those distances, then x_{ij} does not represent a local maximum for d_{\min}^P . If d_{ij} is larger than all of

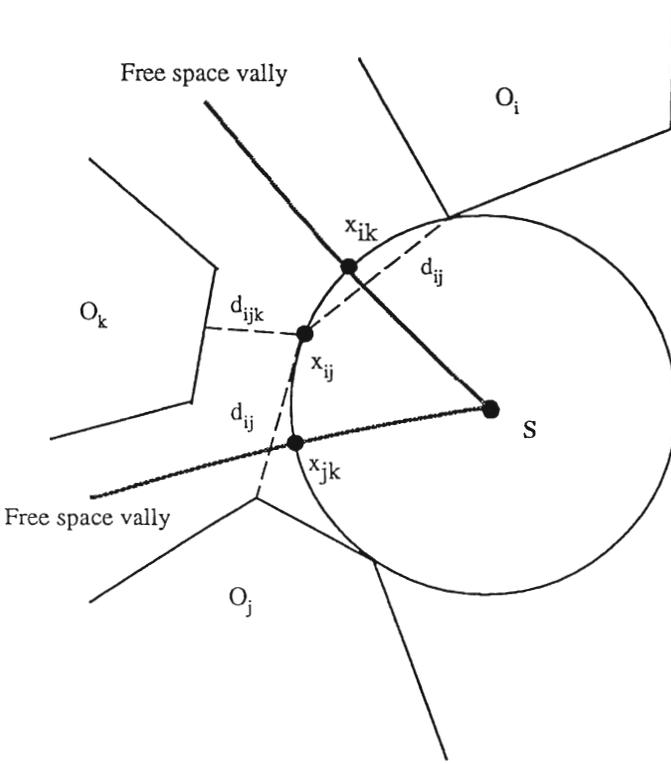


Figure 7. An example of calculating the free valleys. Since $d_{ijk} < d_{ij}$, where d_{ijk} is the distance between x_{ij} to O_k , x_{ij} is not on the free space valley. But x_{ik} and x_{jk} are on the free space valley.

those distances, then x_{ij} is a local maximum point. Consequently, this x_{ij} is labeled as a neighbor point of the starting point. Each neighbor point is then used as the center of a new reach circle.

3. Step 1 to Step 3 are repeated from the destination position of the reconfigurable object.
4. This process is repeated until all of the free space is covered by reach circles. The straight line segments that connect centers of the circles define the free space valleys. A junction of free space valleys is defined to be a place where two or more free space valleys meet.

4.1.2. Selecting the Best Candidate Path

After generating the set of free space valleys, a candidate path from the starting point to the destination point is selected. The selected path is optimal in the sense that it represents the best tradeoff between path length and maneuverability.

In order to measure the difficulty of traversing along each free space valley, a *containing circle* of the reconfigurable object is defined to be the smallest circle that circumscribes the reconfigurable object. Since the shape of the reconfigurable object can be changed, the radius of the containing circle is defined as

$$r_c = \min_{\theta_1, \dots, \theta_6} r'_c \quad (4)$$

where $\theta_1, \dots, \theta_6$ are the joint angles of the reconfigurable robot and r'_c is the radius of a circle which can circumscribe the reconfigurable object. Furthermore, W_{\min} is defined to be the minimum width of the reconfigurable object.

With the above definitions, a cost function which is a measure of the difficulty of traveling in a free space valley can be defined. This cost function C is designed to be proportional to the length of the branch and inversely proportional to the value of d_{\min}^P along the branch:

$$C = \int W(P) |dP| \quad (5)$$

where $W(P)$ is a weighting factor which depends on d_{\min}^P at point P . If d_{\min}^P is greater than r_c , a uniform weight of 1 is used (which means that there is no collision between the reconfigurable object and the obstacles at point P). If d_{\min}^P at P is smaller than $0.5W_{\min}$, the maximal possible weight (= 1000 in practical implementations) is used (which means that there must be a collision between the reconfigurable object and the obstacles at point P). Otherwise, the weighting function is defined as

$$W(P) = 1000 - 999(d_{\min}^P - 0.5W_{\min}) / (r_c - 0.5W_{\min}) \quad (6)$$

which means that the closer the point P is to the obstacles, the greater the possibility of collision between the reconfigurable object and the obstacles, and thus the larger the cost associated with this path.

4.1.3. The Open Regions, Dead Regions and Hard Regions

Along each candidate path, there may exist some regions in which the reconfigurable object can turn around without changing its configuration and without colliding with the obstacles. Such regions are called *open regions*. Clearly, every point P in an open region satisfies $r_c < d_{\min}^P$. There may also exist some regions called *dead regions* in which the reconfigurable object cannot move without colliding with any obstacles, no matter how it reconfigures itself. This occurs if $W_{\min} > 2d_{\min}^P$. Regions other than dead regions and open regions in which the reconfigurable object has to maneuver in order to avoid collisions are called *hard regions*. It is clear that the union of hard regions is equal to the difference between the free space and the union of open regions and dead regions.

4.2. LOCAL PATH PLANNING IN THE HARD REGION

After the best global candidate path has been chosen, it should be verified that the best candidate path is collision-free. Clearly, if the best candidate path contains any dead regions then the problem cannot be solved. At the other extreme, if the best candidate path is composed entirely of open regions then the problem is trivially solved. The interesting case occurs when there are hard regions interspersed with the open regions. The paths in the hard regions are divided into a finite number of small intervals. The connecting point of each pair of adjacent intervals is defined as a node on the path.

The procedure for finding a collision-free path in a hard region can be outlined as follows:

1. Find all the collision free feasible regions for all hard regions along the path.
2. Verify the existence of a transition between the collision free feasible regions obtained in Step 1.
3. Build a connection graph whose vertices are the CFFR's obtained in Step 1 and whose edges are the transitions among these CFFR's obtained in Step 2.
4. Search a collision-free path for the hard region in the connection graph. These steps are discussed in more detail in the following sections.

4.2.1. Finding All CFFR's for the Hard Region Along the Path

To verify the collision-freeness of the best candidate path, all the collision-free feasible configurations (CFFR's) of the reconfigurable object at each node along the path need to be found. Since the orientation α of the carried object can be changed, to find all the CFFR's for each node, α is quantized into n_α intervals, where n_α is a positive integer. For each node in the hard region, at each interval of the orientation α , all CFFR's of the reconfigurable object can be found by using the method described in Section 3.

4.2.2. Verifying the Existence of Transitions Among CFFR's

To find a collision free path, one needs to know if the reconfigurable object can move within one node or can move from one node to an adjacent node without colliding with the obstacles. It is easy to show that at a given node and a given orientation of the carried object, a necessary and sufficient condition for a transition to exist between a pair of CFFR's is that there exists a common value of θ_5 among these pairs of CFFR's. Furthermore, if two CFFR's are at the same node, but at adjacent quantized orientations, the condition for a transition to exist between them is that they have the same values of $\theta_1, \dots, \theta_6$. To study the transition between a pair of CFFR's at different nodes, the notation θ_i , $i = 2, 3, 4, 5$ is used to represent a set of possible values of joint angle θ_i at given values of θ_1 , θ_6 and α . Assuming that the quantization interval is chosen correctly, it

can be observed that there is a transition between a CFFR at node j and a CFFR at node k , if

$$\theta_i \text{ for the } j\text{th node} - \theta_i \text{ for the } k\text{th node} \leq \text{one discrete interval } (i = 1, 6), \quad (7)$$

$$\alpha \text{ for the } j\text{th node} - \alpha \text{ for the } k\text{th node} \leq \text{one discrete interval, and} \quad (8)$$

$$\theta_i \text{ for the } j\text{th node} \cap \theta_i \text{ for the } k\text{th node} \leq \emptyset \quad (i = 2, 3, 4, 5) \quad (9)$$

4.2.3. Building the Connection Graph

After finding all the CFFR's at each node and the collision-free transitions between each pair of adjacent CFFR's, one can construct a graph called a connection graph by assigning each CFFR to a vertex and each collision-free transition between a pair of CFFR's to an edge connecting the corresponding vertices. Subsequently, a set of connected subgraphs can be found from the graph. If a CFFR at the starting node and a CFFR at the final node are in the same subgraph, then a collision-free path exists in the hard region and the collision-free path can be found by using a graph search method. If a CFFR at the starting node and a CFFR at the final node are in different subgraphs, the best candidate path does not provide a collision-free path. In this situation, a new best candidate path is chosen according to the cost function and the above procedure is repeated. If a collision-free path still cannot be found, then more nodes need to be added to the connection graph. A line which is perpendicular to a free space valley and passes through a node on the path provides an additional degree of freedom. In this case, the part of these lines which is inside free space is quantized. Each quantized value can be considered as a node and they are added to the connection graph. If a collision-free path still cannot be found in the modified connection graph, then it can be concluded that there is no collision-free path for the reconfigurable object in this workspace. For illustration, a connection graph is given below:

EXAMPLE 4-1. Suppose that in Figure 6, valley AB is chosen as the best candidate path, then nodes N_1, N_2, \dots, N_n are obtained. A part of the connection graph for the best candidate path is given in Figure 8. This connection graph is represented in a three-dimensional space with axes N, α and CFFR. Axis N is used to represent the nodes along the path; axis α is used to represent the orientations at each node; and axis CFFR is used to represent the CFFR's at each node and each orientation. Since a 6-joint closed chain can be represented by variables θ_1, θ_6 and θ_5 , each CFFR can be further represented in a three-dimensional space with axes θ_1, θ_6 and θ_5 . In Figure 8, each CFFR is represented by a 3-tuple in which the first parameter is the value of θ_1 , the second parameter is the value of θ_6 and the third parameter is the range of θ_5 . For illustration, only two possible values for θ_1 and θ_6 are chosen. They are $A = 135^\circ$ and $B = 225^\circ$. The transitions between the CFFR's are shown in the figure. With a graph search

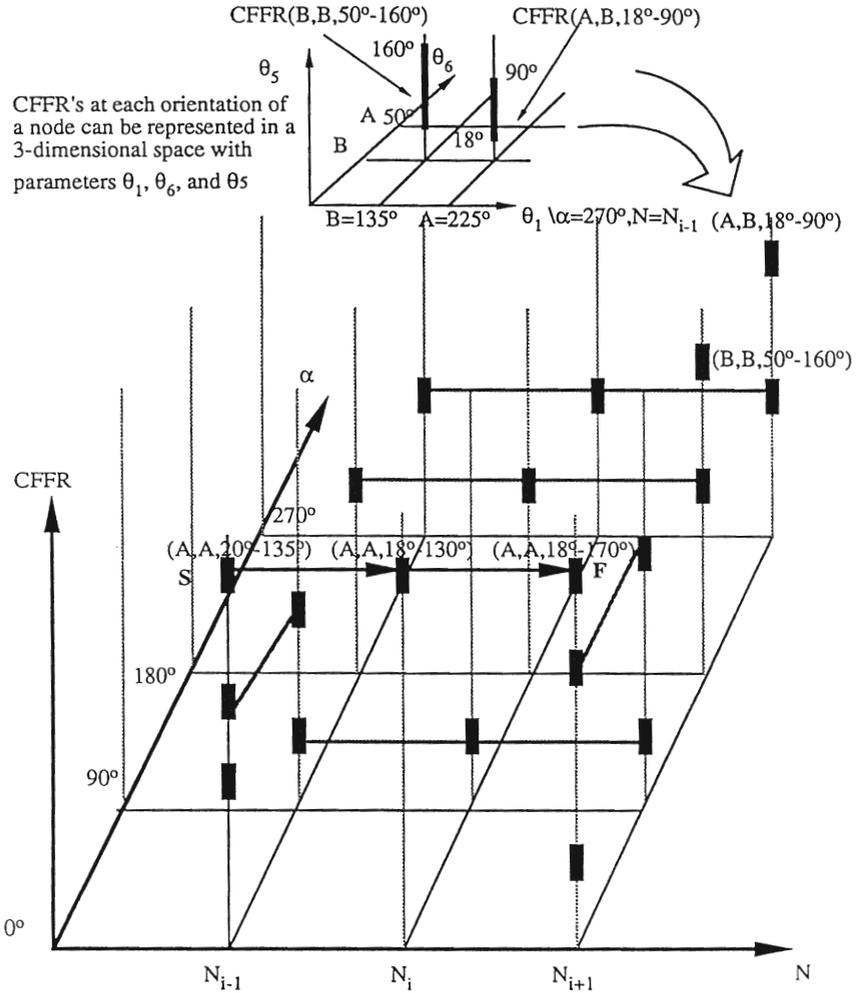


Figure 8. A part of the connection graph for Figure 4-1 where each vertex, i.e., a CFRR, is labeled by a 3-tuple where the first parameter is the value of θ_1 , the second parameter is the value of θ_6 and the third parameter is the range of θ_5 . In order to simplify the illustration, only two possible values for θ_1 and θ_6 , i.e., A and B where $A = 135^\circ$ and $B = 225^\circ$ are shown.

method, a shortest path which connects a given CFRR at the starting node *S* and a CFRR at the final nodes *F* can be found as shown in Figure 8.

5. Conclusion

This paper describes an approach to determine a collision-free path for a reconfigurable robot (a mobile robot equipped with two manipulator arms) in a two-dimensional space with clustered obstacles. To the best of our knowledge, all

previous path planning algorithms assume that mobile robots have fixed shapes. These robot descriptions work well when the working environment is not crowded. When the robot work space is crowded, the maneuvering ability of robots should be used to achieve collision avoidance. This point leads to our approach of taking a reconfigurable formulation instead of a fixed shape formulation for the robot.

Our proposed approach is composed of two major algorithms: (1) the collision-free feasible configuration finding algorithm which finds the collision-free feasible configurations of the robot when the position and the orientation of the carried object is given, and (2) the collision-free path finding algorithm which employs two levels of planning (i.e., global path planning and local path planning) to find a collision-free path by using a graph search method. The proposed algorithm provides an optimal path according to the cost function which takes into account of both the length of the path and the difficulty (related to the amount of maneuvering) of following the path. The variable shape representation of the robot provides a greater possibility of finding a collision-free path in a cluttered environment. The future direction of this research includes extending the proposed path planning approach to three-dimensional space.

References

1. Lozano-Perez, T.: Automatic planning of manipulator transfer movements, *IEEE Trans. Systems, Man, and Cybernetics* **SMC-11**(10) (1981), 681–698.
2. Gouzenes, L.: Strategies for solving collision-free trajectories problems for mobile and manipulator robots, *Int. J. Robotics Res.* **3**(4) (1984), 51–65.
3. Bajaj, C. and Kim, M.-S.: Generation of configuration space obstacles, I. The case of a moving sphere, Technical Report CSD-TR-565, Department of Computer Science, Purdue University, West Lafayette, IN, December 1985.
4. Erdmann, M. and Lozano-Perez, T.: On multiple moving objects, in *Proc. IEEE Int. Conf. Robotics and Automation*, 1986, pp. 1419–1424.
5. Lozano-Perez, T.: A simple motion-planning algorithm for general robot manipulators, *IEEE J. Robotics and Automation* **RA-3**(3) (1987), 224–238.
6. Schwartz, J. T. and Sharir, M.: On the piano movers' problem, III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers, *Int. J. Robotics Res.* **2**(3) (1983), 46–75.
7. Kedem, K. and Sharir, M.: An efficient motion planning algorithm for a convex polygonal object in two-dimensional polygonal space, Technical Report No. 253, Computer Science Department, Courant Institute, October 1986.
8. Brooks, R. A.: Solving the find-path problem by good representation of free space, *IEEE Trans. System, Man, and Cybernetics* **SMC-13** (1983), 190–197.
9. Khatib, O.: Real time obstacle avoidance for manipulators and mobile robots, *Int. J. Robotics Res.* **5**(1) (1986), 90–98.
10. Hwang, Y. K. and Ahuja, N.: Robot path planning using a potential field representation, in *Proc. of IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, San Diego, June 1989.
11. Carriker, W., Khosla, P. and Krogh, B.: The use of simulated annealing to solve the mobile manipulator path planning problem, in *IEEE Int. Conf. Robotics and Automation*, March 1986, pp. 204–209.
12. Wong, E. K. and Fu, K. S.: A hierarchical orthogonal space approach to three-dimensional path planning, *IEEE J. Robotics and Automation* (March 1986), 42–53.

13. Xue, Q., Sheu, P. C.-Y. and Maciejewski, A. A.: Determining the collision-free joint space graph for two cooperating robot manipulators, *IEEE Trans. Systems, Man, and Cybernetics* **23**(1) (1993), 285–294.