Pergamon

Mech. Mach. Theory Vol. 32, No. 1, pp. 21–37, 1997
Copyright © 1996 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0094-114X/97 $15.00 + 0.00

PII: S0094-114X(96)00042-0

COLLISION-FREE PATH PLANNING AND MODIFICATION BASED ON TASK REQUIREMENTS

KUU-YOUNG YOUNG and CHUN-YUH HUANG†

Department of Control Engineering, †Department of Mechanical Engineering, National Chiao-Tung University, Hsinchu 30039, Taiwan, Republic of China

(Received 21 April 1995)

Abstract—In order to enhance integration between CAD and robots, we propose a scheme to planning kinematically feasible paths in the presence of obstacles. Thus, the feasibility of a planned path from a CAD system is assured before the path is sent for execution. The proposed scheme imposes a geometrical analysis on a given robot manipulator and obstacles. Consequently, geometrical information describing the robot workspace, obstacles and the planned path can be derived. Then, by utilizing path modification strategies appropriate to various industrial tasks, the planned path can be modified so that it is kinematically feasible and collision-free and satisfies task requirements. The analyses in this paper are based on multiple arbitrarily-shaped obstacles and general non-redundant wrist-partitioned types of robot manipulators in which the joint axes are either perpendicular or parallel to one another. For demonstration, simulations based on using the PUMA 560 robot manipulator to perform different tasks in the presence of obstacles were conducted. Copyright © 1996 Elsevier Science Ltd

1. INTRODUCTION

It is widely recognized that the inflexibility of manual programming makes it necessary to integrate computer-aided design (CAD) systems and robot manipulators. However, insufficient information about the limitations of a robot manipulator can cause incompatibility between plans prepared using a CAD system and the actual execution of these plans using a robot manipulator. One important factor inducing such incompatibility is an inability to guarantee the feasibility of a robot path in the planning stage. To resolve this problem, we need to consider the kinematic constraints of the given robot manipulator and constraints resulting from obstacles in the robot workspace. In addition, various industrial tasks will involve different task requirements [1]. For instance, if a task is to move a directionless object from one to another location, then only the starting and end positions need to be considered and the intermediate positions and orientations may be arbitrary. On the other hand, if the object to be moved is a cup of coffee, then the orientation specification also needs to be taken into account. Therefore a path planning scheme that guarantees the kinematic feasibility of a planned path in the presence of obstacles and that satisfies task requirements is imperative in the integration of CAD and robots.

There are various approaches to path planning. Basically, they can be divided into planning in Cartesian space and joint space. To plan a path in Cartesian space usually requires both position and orientation for each point on the path to be specified [2, 3], because both position and orientation need to be defined in order to determine the corresponding joint positions. Paths planned using this approach tend to induce unnecessary infeasibility, since extra constraints are imposed. Furthermore, because the kinematic constraints and obstacles are in general not incorporated into the planning, infeasible paths may not be modified and need to be replanned. Moreover, since the planning procedure is trial-and-error, the replanned path may still be infeasible. Path planning using the joint space approach is easy for control and simple in planning [4, 5]. However, in joint space planning the corresponding Cartesian locations cannot be determined beforehand to satisfy certain task requirements in Cartesian space. In addition, to overcome the obstacle avoidance problem in joint space is also complicated.

A famous approach for obstacle formulation inside the robot workspace is the configuration space (C-space) approach [6]. In the configuration space approach, obstacles for a manipulator with n joints are formulated as C-space obstacles of n-dimensional volume. To represent an

n-dimensional C-space obstacle in C-space, the obstacle is recursively sliced into obstacles of fewer dimensions, until a union of one-dimensional volumes is obtained. Then, feasible paths are located by searching the free regions in the configuration space to avoid obstacles and constraints from the kinematics, etc. Basically, the obstacle is dealt with according to the degrees of freedom of the robot manipulator. Tackling the problem of obstacle avoidance in respect to the joints seems to be inevitable due to intrinsic properties of robot manipulators. However, obstacles are in general three-dimensional (3D) geometrical objects. In order to take advantage of their geometrical properties, we propose a new scheme for obstacle formulation according to the robot geometry. Similar concepts can also be found in several previous works [7, 8]. Another aspect is the trade-off between the exactness of obstacle approximation and the efficiency in path planning. When arbitrarily-shaped obstacles are closely approximated, for instance, by applying the cell decomposition approach, or tree-based technique, etc., a complicated planning procedure will be involved [7]. On the other hand, if an arbitrarily-shaped obstacle is approximated by obstacles of regular shapes, the efficiency in planning will improve at the expense of losing feasible regions in the process of approximation. We adopt the latter concept in the proposed obstacle formulation scheme. To note that, certain feasible regions cannot be well utilized due to the structure of the links, e.g. the feasible regions in the zig-zag edges of an obstacle. Therefore, the proposed obstacle formulation will be according to the structure and movements of the links such that less utilizable feasible regions are cast off in the approximation. For comparison, if obstacles are formulated in Cartesian space, the formulation does not take into account the geometrical properties of the links. Consequently, in the process of planning and obstacle avoidance, more utilizable feasible regions become useless.

To sum up, we propose a scheme to design kinematically feasible paths based on task requirements in the presence of obstacles. In this scheme, obstacles are approximated as compositions of identically shaped elements according to the structures and movements of the links. This makes it easy to approximately map obstacles into the robot workspace. The geometric boundaries of different regions corresponding to kinematic constraints and obstacles will then be generated inside the robot workspace. Similarly, a planned path in Cartesian space can be mapped into the workspace by transforming the joint variables specifying the path into geometrical traces in the workspaces. Consequently, geometrical expressions describing the relationship between the robot workspace, obstacles and planned path can be derived. We propose to use geometrical information derived in this way to develop path modification schemes to assist in path planning. The path modification schemes will be applied to the planned path to incorporate obstacle avoidance of all the links according to different task requirements. In our approach, path modification involves a smaller local search space in planning than a global search space in the whole workspace. Thus our approach promises to be efficient. In addition, a merit in the proposed path planning scheme is that task requirements are incorporated into the planning via the means of path modification, which is not well explored in previous works.

The proposed planning procedure will begin by planning a path in Cartesian space by specifying both position and orientation for each point on the path. The specifications of the position and orientation are based on the specific task requirements. Modification strategies appropriate to different task requirements are then employed to modify the infeasible portions of the path by utilizing the derived geometrical information. In addition to multiple arbitrarily-shaped obstacles, the analyses that follow are based on general non-redundant wrist-partitioned types of robot manipulators in which the joint axes are either perpendicular or parallel to one another. This kind of kinematic arrangement is popular for a major class of industrial robot manipulators. To demonstrate the proposed scheme, simulations based on using the PUMA 560 robot manipulators to perform different tasks in the presence of obstacles were conducted.

2. GENERAL TYPES OF INDUSTRIAL ROBOT MANIPULATORS

Industrial robot manipulators may be categorized according to joint type, connection and offset [9]. Our analysis will concentrate on non-redundant wrist-partitioned types of robot manipulators in which the joint axes are either perpendicular or parallel to one another. Since our analysis is for non-redundant wrist-partitioned types, the number of the primary and minor joints is at most three each, providing a maximum of six degrees of freedom. In the following, the discussion will be based on robot manipulators with six degrees of freedom. Robot types with fewer degrees of freedom can be taken as special cases of the cases discussed here. With three degrees of freedom for the primary joints, there can be at most eight combinations of revolute (R) and prismatic (P) joints with offsets among the links, as summarized in Table 1 [10].

For the eight combinations of the primary joints listed in Table 1, it can be seen that the combinations involving revolute joints are more complicated to analyze than those involving prismatic joints. This is because the prismatic joint moves the link along a straight line, while the revolute joint moves along a curve. For instance, the workspace of a prismatic joint and a revolute joint with their corresponding links operating in the same plane is bounded by two straight lines and two curves; while that of two revolute joints is bounded by five circles determined by the joint ranges and limits of these two joints [10]. Therefore the RRR type in Table 1 is the most complicated case. The remaining seven cases can be analyzed following procedures similar to but simpler than the RRR type by substituting prismatic joint(s) for some revolute joint(s). In order for the RRR type to be non-redundant, the joints cannot all be connected in parallel. The RRR type can be further divided into three sub-cases, as shown in Fig. 1: (a) last two joints in parallel; (b) first two joints in parallel; and (c) all three joints consecutively perpendicular.

A detailed analysis of these three cases can be found in the authors' previous paper [10]. A brief discussion is given below. Note that offsets may be present among the links. Their effects in the workspace analysis are as follows. These offsets are stationary to the corresponding links and can be divided into those parallel or perpendicular to the corresponding links. The effects of those offsets perpendicular to the corresponding links can be removed by proper assignment of the coordinate frames for the joints [11]. The effects of those parallel to the links can be combined with the corresponding links to form expanded links [10]. With these adjustments, the analysis will be equivalent to that of a three-link robot manipulator with no offsets.

For case (a) of the RRR type with no offsets, since the effect of θ_1 in this geometry can be independently identified without involving θ_2 and θ_3, the workspace analysis can be divided into the workspace of joint one and the combined workspace of joints two and three. The analysis of case (b) will be similar to case (a) if it is performed starting from the third joint, followed by the first two joints. For case (c), if the effect of θ_2 can be identified and removed, the traces of links one and three will be in the same plane. However, the effect of joint two cannot be identified from the wrist position only as in the previous two cases. Nevertheless, if the rotation of joint two can be determined through different approaches or more measured positions, e.g. the end position of link two, the analysis will consist of the workspace of joint two and that of joints one and three. In summary, for the primary joints with three degrees of freedom, via proper arrangement the traces of two of the links will be in the same plane, and the trace of the remaining link will be in a plane perpendicular to that plane. In the analysis below, case (a) of the RRR type with no offsets will be used as a case study. In fact, the kinematic arrangement in case (a) is quite common among industrial robot manipulators, such as PUMA type and ASEA type. While the extension may not be so straightforward, the other cases can be tackled following a similar concept as that for case (a).

As for the minor joints, since the target robot manipulators are wrist-partitioned, there will be no offsets assigned to them and they consist of all revolute jointѕ. In order to be non-redundant, the geometry of the minor joints has to be consecutively perpendicular, as shown in Fig. 2. The analysis for the geometry in Fig. 2 is general for the minor joints with three degrees of freedom. Three separate traces need to be dealt with in the analysis of the minor joints, due to the mutually perpendicular geometry.

Table 1. Classification of the primary joints

Case	1	2	3	4	5	6	7	8
J1	R	R	R	P	R	P	P	P
J2	R	R	P	R	P	R	P	P
J3	R	P	R	R	P	P	R	P

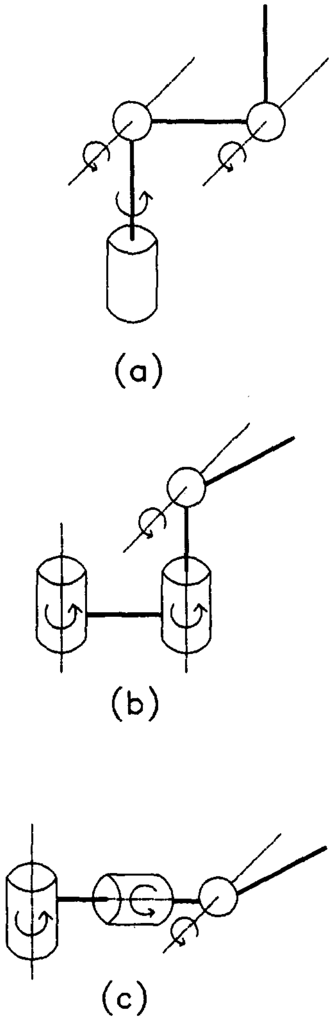

Fig. 1. Geometries of primary joints of the RRR type.

3. WORKSPACE ANALYSIS

In order to formulate 3D obstacles inside the 6D workspace of a non-redundant wrist-partitioned robot manipulator, we propose to separate it into two 3D workspaces by selecting the wrist position $\underline{p}_w$ as the reference point. It is because that there are no link lengths among the minor joints. The robot workspace can then be divided into the workspace of the primary joints and that of the minor joints. In addition, within the workspaces of the primary and minor joints, subworkspaces may be generated in respect to the joints by utilizing kinematic information from the specific geometry

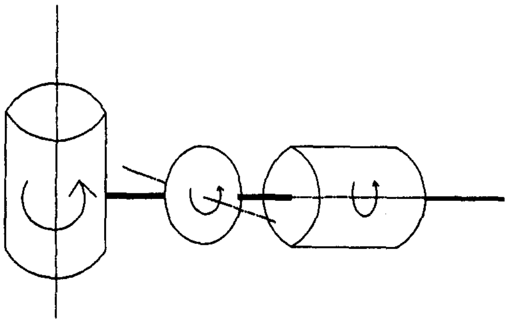

Fig. 2. Geometry of minor joints.

of the given robot manipulator [12]. Thus these subworkspaces are geometrical regions corresponding to joint variables and they may be related to each other according to their geometrical properties, as illustrated below.

The workspace of the primary joints can also be viewed as that of the wrist, denoted by PWK:

$$\mathrm{PWK} = \{\underline{p}_w | \underline{p}_w = f(\underline{\theta}_p) \text{ for every } \underline{\theta}_p\} \quad (1)$$

where $\underline{\theta}_p$ denotes a set of joint variables of the primary joints and f is a forward kinematic function, which transforms a set of joint variables into its corresponding Cartesian position [13, 14]. Inside PWK the feasible regions, regions corresponding to different configurations and singular regions, can also be determined. Here the feasible regions correspond to feasible $\underline{\theta}_p$, i.e. the elements of $\underline{\theta}_p$ are within joint limits, the regions of different configurations are defined as different sets of points with the same arm configurations and the singular regions defined as the boundaries of the feasible regions and their vicinities along with positions where corresponding joint variables are undefined [15].

Based on the discussion in Section 2, the workspace of the primary joints (PWK) can be further divided into two subworkspaces via appropriate arrangement. These two subworkspaces correspond to a single joint and the remaining two joints, denoted by PWK_1 and PWK_2, respectively, are represented as follows:

$$\mathrm{PWK}_1 = \{\underline{p}_1 | \underline{p}_1 = f(\theta_1) \text{ for every } \theta_1\} \quad (2)$$

where θ_1 denotes a joint variable for the single joint and $\underline{p}_1$ the corresponding position and

$$\mathrm{PWK}_2 = \{\underline{p}_2 | \underline{p}_2 = f(\underline{\theta}_2) \text{ for every } \underline{\theta}_2\} \quad (3)$$

where $\underline{\theta}_2$ denotes a set of joint variables for the two joints and $\underline{p}_2$ the corresponding position.

To analyze the workspace of the minor joints, denoted by MWK, equation (4) is employed:

$$R_e = R_w * {}^{w}R_e \quad (4)$$

where R_e, R_w and ${}^{w}R_e$ denote the orientational part of a Cartesian point in a planned path, a wrist point and the Cartesian point with respect to the wrist coordinates, respectively. It can be seen from equation (4) that each element within PWK will not only determine a wrist point but also a wrist coordinate frame for mounting the minor joints. The effect from the primary joints needs to be identified and removed before the analysis of MWK. The details for identifying the coupling between PWK and MWK can be found in [16]. In addition, the robot end-effector needs to be mounted upon the wrist to analyze MWK. If the length of the end-effector is present in only one direction, then only the rotations of two of the degrees of freedom can be identified, since the rotation of one of the three degrees of freedom will be redundant. Therefore the end-effector has to consist of lengths in at least two different directions in order to identify MWK for up to three degrees of freedoms [16]. With the end-effector position constrained by the limits and ranges of the minor joints, the analysis of MWK is equivalent to that of the positional feasibility of the end-effector, represented as

$$\mathrm{MWK} = \{{}^{w}R_e | {}^{w}R_e = g(\underline{\theta}_m) \text{ for every } \underline{\theta}_m\} \quad (5)$$

where g is a forward kinematic function, which transforms a set of joint variables into its corresponding orientation and $\underline{\theta}_m$ a set of joint variables of the minor joints. Similarly, the feasible regions, areas of different configurations and the singular areas can be identified within MWK. There will be three subworkspaces inside MWK, since all three minor joints are consecutively perpendicular.

4. OBSTACLE FORMULATION IN THE WORKSPACE

In Section 3, a 6D robot workspace is divided into two 3D workspaces, PWK and MWK. To formulate an obstacle in the robot workspace, we will first formulate it in PWK. Take a robot manipulator as case (a) of the RRR type in Fig. 1(a) for discussion. Then PWK_1 corresponds to joint one and PWK_2 to joints two and three. As shown in Fig. 3, the mapping of an obstacle in PWK in PWK_1 can be obtained by taking its projection onto PWK_1. Its mapping in PWK_2, on

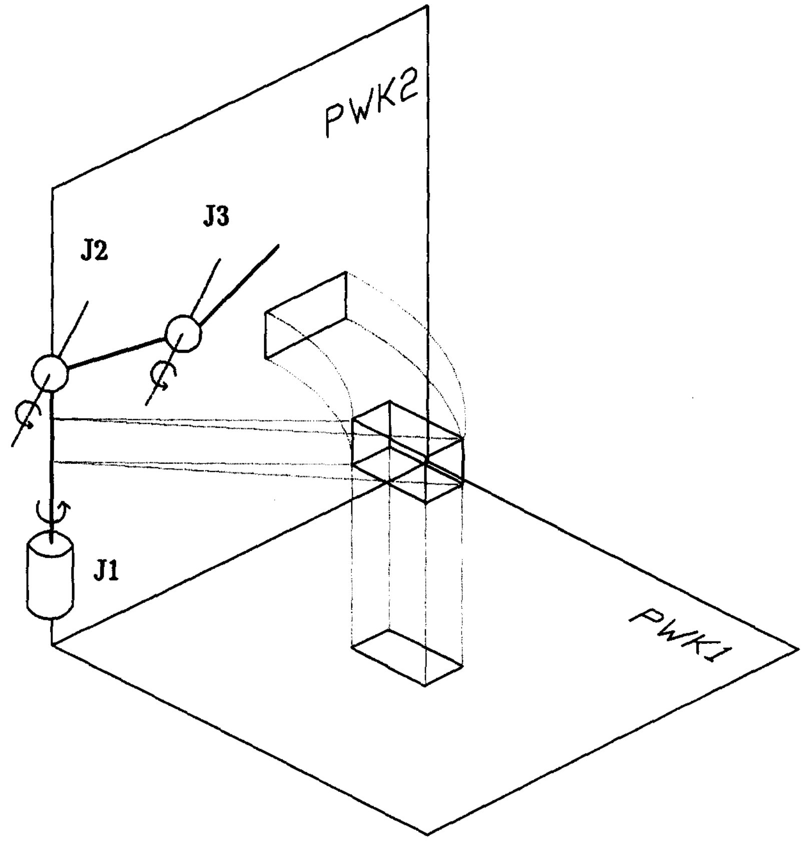

Fig. 3. Mappings of an obstacle in PWK_1 and PWK_2.

the other hand, will not be the direct projection of the obstacle onto PWK_2, because the plane containing PWK_2 is determined by the position of joint one. To remove the effect of joint one, PWK_2 will be rotated to a plane where θ_1 equals to a fixed value, e.g. $\theta_1 = 0°$. To obtain the mapping of the obstacle on PWK_2, the obstacle will first be sliced by the planes containing the joint one axis. These slices will then be rotated to the plane where PWK_2 is located. The resulting trace of these rotated slices will be the mapping of the obstacle in PWK_2. In Fig. 3, it can be seen that the trace of every rotated slice may not be exactly overlap. Consequently, the resulting trace in PWK_2 in fact corresponds to an obstacle larger than the original one. Nevertheless, for certain obstacle for which the trace of every rotated slice exactly overlaps, e.g. a donut-shaped obstacle with its circular side perpendicular to PWK_1, the resulting trace in PWK_2 via the above procedure will not correspond to an expanded obstacle.

With the introduction of obstacles, the feasible regions in the robot workspace will be reduced by an amount including not only the resulting trace itself but also its neighboring area corresponding to the structure of the links [1]. For obstacles of different shapes, the affected areas will vary. Each obstacle shape thus demands its own collision-free workspace analysis. To avoid a case-by-case workspace analysis, we propose approximating obstacles of various shapes by elements of the same shape. The division criterion will be to let the mappings in the workspaces for these elements correspond to obstacles that well approximate the original obstacles; this criterion will be discussed in Section 4.1. Consequently, the workspace analysis needs to be performed only for one specific obstacle shape. On the other hand, to formulate an obstacle in MWK, since the end-effector in general is small compared with the links corresponding to the major joints, we propose to expand the obstacle so that the presence of obstacles will not affect MWK. The expansion criteria will depend on task requirements, discussed in Section 4.2.

4.1. Obstacle in PWK

In order to represent arbitrarily-shaped obstacles using elements of the same shape with good approximation, two main factors need to be considered: (a) the shape of the element, and (b) the division criterion. These two factors are in fact compound. Again, the robot manipulator in case (a) above is taken for discussion. To minimize the waste of feasible regions in approximating the obstacle, the obstacle is first divided according to its mapping in PWK_1 and PWK_2. Each mapping will be divided into a collection of convex shapes via cutting planes, as shown in Fig. 4. To note here that for obstacles that are not polygons, an approximation process will be demanded to transform them into polygons. It is well known that describing a 3-D object from two viewing directions will cause important information to be missed in some cases. For example, if we take the two viewings depicted in Fig. 4 for the obstacle in Fig. 5, the division will result in an over-approximation of the size of the middle portion of the obstacle. Therefore, we propose adding in another viewing during the division process, as demonstrated in Fig. 5. In Fig. 5, a division is performed in the third mapping plane to resolve the over-approximation situation in the middle portion. The entire process is then similar to taking the top, front and side views in describing a 3-D object. Since the mapping in PWK_2 is not a direct 2-D projection, the third projection cannot be selected to be perpendicular to the previous two mappings. Instead, we choose it to be projected on the plane perpendicular to the plane that contains the joint one axis and bisects the obstacle in half, as shown in Fig. 5.

Via the division process above, the mappings of the obstacle will be divided to be convex-shaped in the three viewing planes. By properly matching these convex-shaped mappings in PWK_1 and PWK_2, the elements of which the obstacle is composed can be found. The next step will be to generate the mappings of the elements in PWK. As mentioned in the previous section, the mapping in PWK_1 is a direct 2-D projection. However, the mapping in PWK_2 is obtained by slicing the element with the planes containing the joint one axis and then rotating these planes to the plane where PWK_2 is located. To avoid the procedure of slicing the element by a series of planes and the computation for deriving the resulting mapping of the slices in PWK_2, we propose to bound the element using four planes. Two of the planes are parallel to the ground and contact the element in the highest and lowest locations, and the other two planes are curved surfaces with the z_1 axis as the central axis and contact the element in the innermost and outmost locations. When we combine these planes with the two planes used to bound the element in obtaining the mapping in

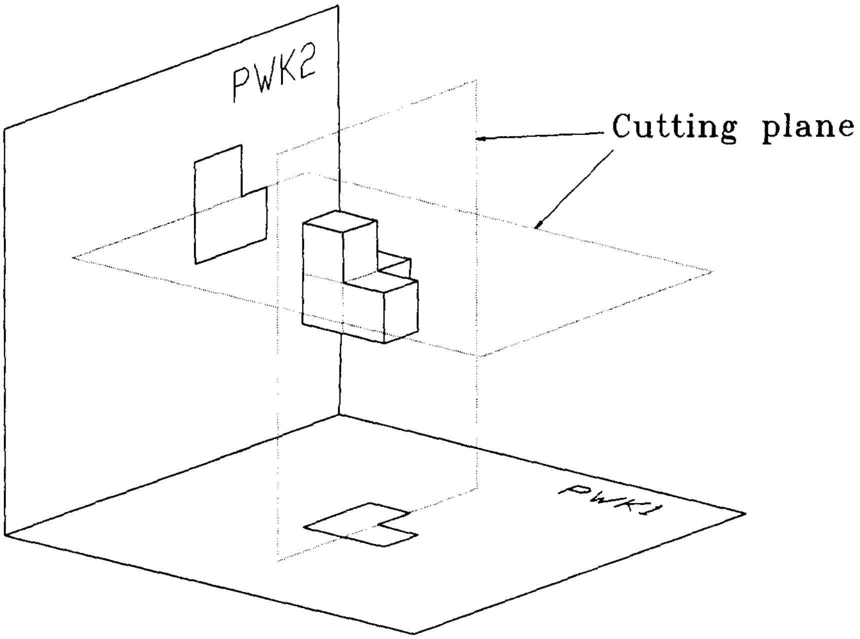

Fig. 4. Obstacle division according to the mappings in PWK_1 and PWK_2.

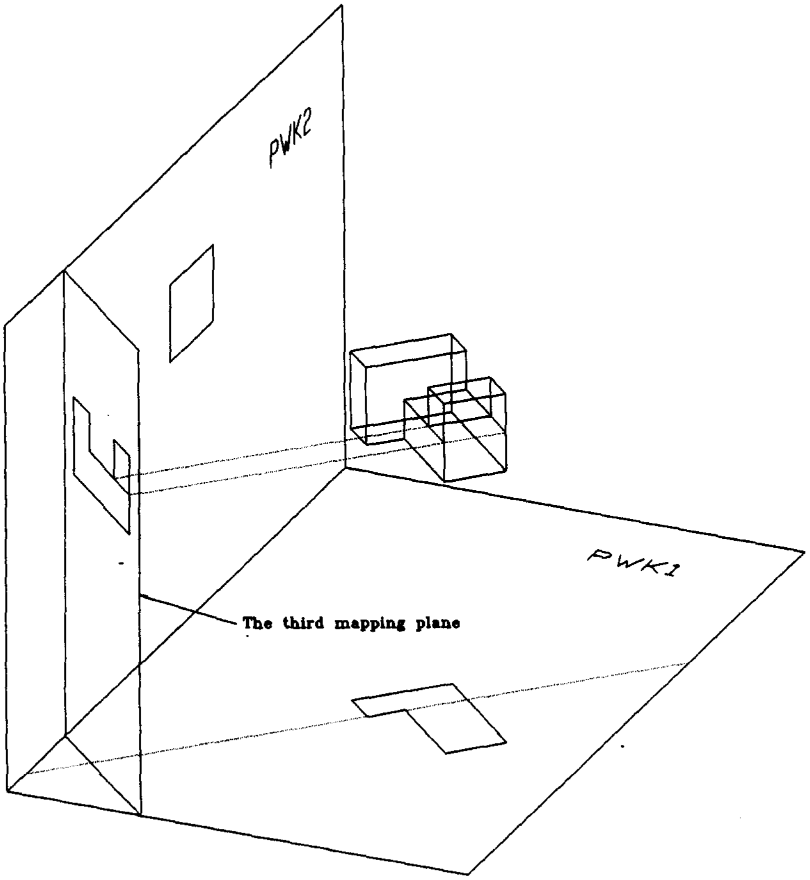

Fig. 5. Obstacle division from three viewing directions.

PWK_1, which contain the z_1 axis and contact the element in the leftmost and rightmost locations, a total of six planes are used to obtain the mappings of an element in PWK, as shown in Fig. 6.

After the mappings of the elements are derived in PWK, the feasibility of the wrist path of a planned path can be checked. When the projections of the wrist path fall outside the area in PWK_1 bounded by the two planes corresponding to the rightmost and leftmost edges of the element, the presence of the element will not affect the feasibility of the wrist path. When the projections fall within the area, it is possible that the element may collide with the wrist path. It will need a further checking on the feasibility in PWK_2. The mapping of an element in PWK_2 via the proposed approach will be a rectangle. However, considering the obstacle formulation in MWK, a circular disk is a better representation of the obstacle in PWK_2; this will be discussed in Section 4.2. In other words, the feasibility in PWK_2 will be analyzed in the presence of circular-shaped obstacles. Therefore the rectangular mapping will be approximated by circular disks. Taking into account the tradeoff between more accurate approximation and the number of circular disks, we propose using two circular disks to cover the rectangle in the initial approximation [17]. Among the circular disks, the two that fully cover the rectangle with the closest approximation to the size of the rectangle will be adopted. These are derived using Theorem 1.

Theorem 1. An $M \times N$ rectangle, $M \geqslant N$, can be approximated by two circular disks for minimum coverage, when these two circular disks are of equal diameter and circumscribe two end points of a short side and the middle points of the two long sides, as shown in Fig. 7.

The proof of Theorem 1 is given in Appendix A. In some cases, when the area of the circular disk that circumscribes the rectangle is close to the area covered by the two circular disks derived in Theorem 1, then to simplify the analysis only one circular disk will be used to approximate the rectangle. On the other hand, if the area covered by the two circular disks is much larger than that of the rectangle, the rectangle will be divided into halves and covered by four circular disks. The dividing procedure may be performed repeatedly.

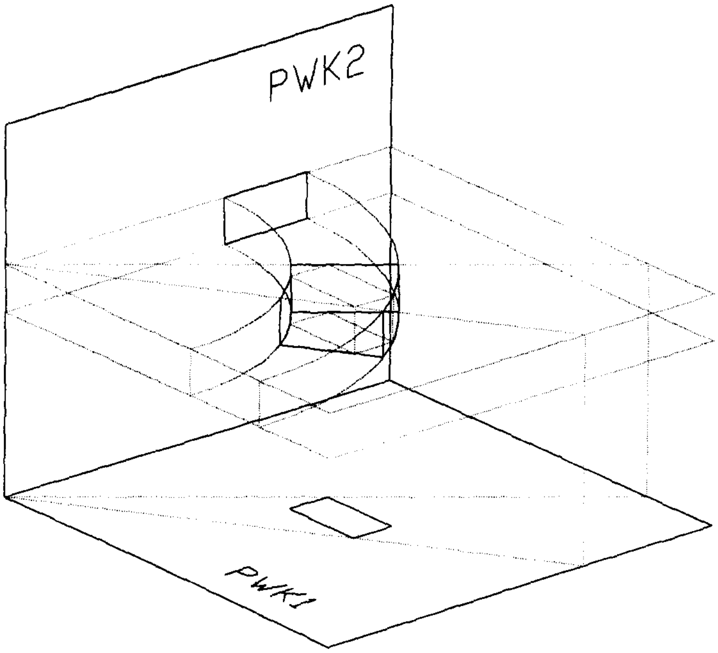

Fig. 6. Generating mappings of elements used to construct the obstacle in PWK.

As stated earlier, the infeasible region in PWK₂ includes not only the mapping of the obstacle itself, but also its neighboring area. With the rectangular mapping approximated by circular disks, the affected infeasible region of PWK₂ will depend on the relationship between the circular disks and links two and three. Depending on the location of a circular disk relative to links two and three, their relation can be separated into three cases: (a) Link two cannot reach the disk; (b) Link two can reach the disk and is tangential to it at the point of contact; (c) Link two can reach the disk and is not tangential to it at the point of contact. The derivation of how to generate PWK₂

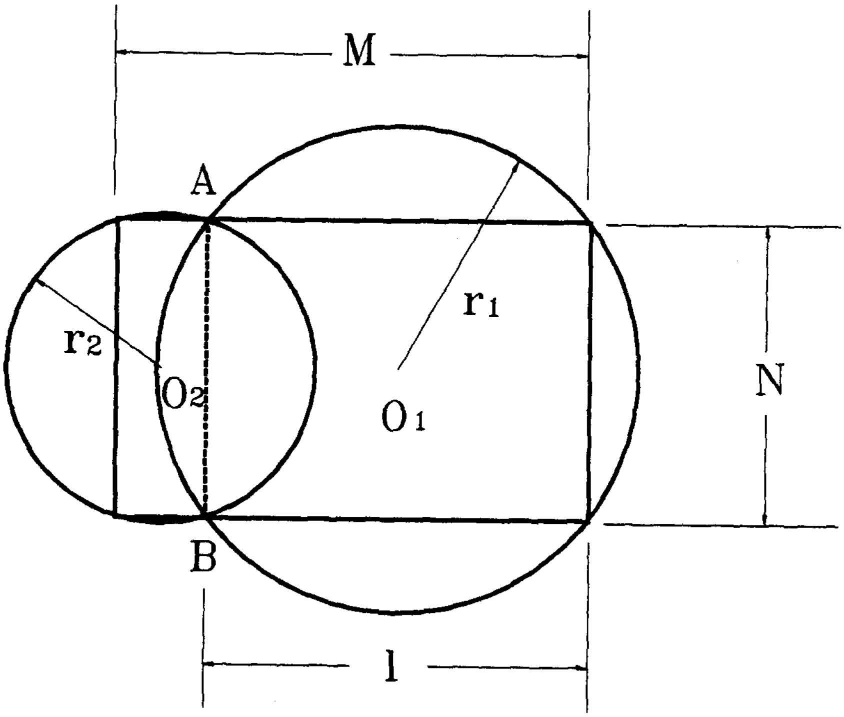

Fig. 7. Minimal coverage of a rectangle by two circular disks.

in the presence of a circular disk can be found in [1]. When there are multiple circular disks in PWK_2, the area in PWK_2 affected by multiple disks can be derived by merging the areas that correspond to each disk. When multiple obstacles are present, if the mappings of these obstacles in PWK_1 are away from each other, they can be tackled separately. If some of them overlap, the corresponding mappings for the overlapping portions in PWK_2 will also be derived by merging the areas that correspond to each obstacle [1].

4.2. Obstacle in MWK

For a wrist-partitioned type of robot manipulator, there is no offset beyond the wrist point. However, in performing industrial tasks, the tool needs to be mounted upon the wrist. In order to ensure that the presence of obstacles will not affect MWK when a robot manipulator is equipped with a tool, we propose expanding the size of the obstacle in proportion to the length of the tool. The expansion will cause us to fail to find a feasible path in the worst case. However, since in general the tool is small compared with the links corresponding to the major joints, expanding the size of the obstacle has proven to be a reliable and efficient method in most practical situations [18].

Previous research in general dealt with obstacle expansion due to the tool by employing a bounding box to cover the tool in various orientations constrained by the minor joints [18]. In this approach, the obstacle is expanded by the length of the tool. In this paper the expansion will depend on different task requirements. For a task in which the orientation is of concern, we propose enlarging the obstacle by the length of the tool. Consequently, the mappings of the obstacle in PWK_1 and PWK_2 will be enlarged on the edges according to the length of the tool. After the expansion, no matter what the orientation of the tool is, its presence will not affect the feasibility of MWK. On the other hand, if the position specification is crucial for a given task, we propose enlarging the obstacle by a portion of the tool length corresponding to the distance between the tool-tip and the obstacle. Less feasible regions will then be discarded, compared with the expansion above. The reason that the obstacle can be expanded by a portion of the tool length is that the distance between the tool-tip and the obstacle is known in this case. For a fixed tool-tip position, the goal of the expansion is to maintain a certain margin between the wrist and the obstacle, so that the tool will be collision free, no matter what the relative orientation of the tool-tip to the wrist is [1].

4.3. Algorithm for formulating obstacles in the workspace

Based on the discussions above, the algorithm for obstacle formulation in the workspace is organized as follows:

Obstacle formulation algorithm. Formulate multiple arbitrarily-shaped obstacles in the workspace of a non-redundant wrist-partitioned type of robot manipulator.

Step 1

Separate the workspace of the robot manipulator into PWK and MWK by selecting the wrist position as the reference point. Then determine PWK_1 and PWK_2 for PWK, as described in Section 3.

Step 2

Divide each obstacle into convex-shaped elements by taking its top, front and side views in PWK, as described in Section 4.1. An approximation process may be required to transform obstacles that are not polygons into polygons.

Step 3

For each convex-shaped element derived in Step 2, utilize the six planes corresponding to the highest, lowest, innermost, outmost, leftmost and rightmost locations of the element to determine its mappings in PWK_1 and PWK_2, as described in Section 4.1. The mapping in PWK_2 will be a rectangle and it is further covered by a certain number of circular disks, as discussed in Section 4.1.

Step 4

Perform collision-free workspace analysis for PWK_2 in the presence of a circular disk, as described in Section 4.1. For those mappings in PWK_1 and PWK_2 corresponding to multiple obstacles, if the mappings of these obstacles in PWK_1 are away from each other, they will be dealt with separately.

If some of them overlap, the corresponding mappings for the overlapping portions in PWK_2 will be derived by merging the mappings corresponding to each obstacle.

Step 5

Determine the expansion of the obstacle for MWK. For a task in which the orientation is of concern, the mappings of the obstacle in PWK_1 and PWK_2 will be enlarged by a length of the end-effector, as discussed in Section 4.2. For a task in which the position is of concern, the obstacle will be enlarged by a portion of the tool length corresponding to the distance between the tool-tip and the obstacle, as described in Section 4.2.

5. PATH PLANNING AND MODIFICATION

Based on the above analyses of the robot workspace and obstacle formulation, a planned path can be mapped into PWK and MWK by transforming the joint variables corresponding to the planned path into geometrical traces in the workspaces. Consequently, geometrical information describing the relationships between the robot workspace, obstacles and the planned path can be derived. In utilizing the derived geometrical information for path planning, task requirements resulting from different industrial tasks need to be considered. In order to satisfy task requirements, path planning in Cartesian space is more suitable than other approaches. However, planning in Cartesian space usually imposes extra constraints as aforementioned. For instance, when a path is planned to follow a curve for a writing task, the orientation can be arbitrary. However, both position and orientation still need to be specified for each point on the path in order to determine the corresponding joint variables. The phenomenon of over-specification in Cartesian path planning is due to the intrinsic properties of joint versus Cartesian spaces.

To resolve the problem of over-specification, we propose to employ modification in planning by utilizing the freedom available under different task requirements [10]. For tasks with position (orientation) specification, the degrees of freedom in orientation (position) can be utilized for modification. If there are no limitations for either position or orientation, then the degrees of freedom in both position and orientation can be used for modification. Although in the proposed scheme both position and orientation need to be specified for each point on the planned path, some of the specifications may be arbitrary for different tasks. For instance, in a task of following the sides of a table, the position needs to be specified according to the table, while the orientation may be chosen freely. In addition, in some cases different requirements may be imposed for different processes involved in an industrial task. A path may then be planned with different portions governed by different modification strategies. The modification in achieving a feasible path under various constraints can be formulated as a nonlinear optimization problem:

$$\begin{aligned}&\text{Minimize: Deviation between original and modified paths.}\\&\text{Subject to: Task requirements, obstacles and kinematic constraints.}\end{aligned}\qquad(6)$$

In summary, we propose a path planning and modification scheme for designing a kinematically feasible path based on task requirements in the presence of obstacles. The scheme is formulated as follows:

Path planning and modification algorithm. Plan a feasible path satisfying task-level, geometrical and kinematic constraints by utilizing task-dependent modification strategies.

Step 1

Design a path by specifying both position and orientation for each point on the path, where the specification of position and orientation may vary with different task requirements.

Step 2

Construct the workspace corresponding to the kinematic constraints inside the robot workspace utilizing the special geometry of the given robot manipulator.

Step 3

Formulate obstacles inside the workspaces constructed in Step 2 utilizing the Obstacle Formulation Algorithm in Section 4.3.

Step 4

Derive geometrical information about the planned path in the robot workspace described in Step 3. Then apply appropriate task-dependent modification strategies to modify the infeasible portions of the planned path.

In what follows, the planning and modification are discussed according to position and orientation, respectively.

5.1. Planning with position modification

When the orientation specification needs to be followed, position modification will be utilized in the planning. The task requirement in the nonlinear optimization problem described in equation (6) will be the orientation specification. Since the orientation of the tool needs to be maintained, the deviation between the original and modified tool-tip paths will be equivalent to that between the original and modified wrist paths. Thus the minimization is chosen as the minimal deviation between the original and modified wrist paths. Consequently the minimization can be performed in PWK and formulated as follows:

$$\begin{aligned}&\text{Minimize: Deviation between original and modified wrist paths.}\\&\text{Subject to: Orientation specification, obstacles and kinematic constraints.}\end{aligned}\tag{7}$$

To maintain the orientation specification of a given path is not a very strict constraint. Most industrial robots have quite large joint ranges for their minor joints, so the specified orientation can in general be provided. For instance, the PUMA 560 robot manipulator has a range of 280° for joint four, 200° for joint five and 532° for joint six. However, in order not to hit itself, joint five cannot have a range of larger than 360°. This limitation generates infeasible regions inside the joint five workspace. The infeasibility can be resolved by adjusting θ_1, $\theta_2 + \theta_3$ and θ_4 [16]. With a broad feasible range for MWK, the proposed scheme will first modify the traces of PWK to obtain a feasible collision-free wrist path if possible. Then, based on the new R_w and the specified R_e on the planned path, the new traces in MWK can be derived and modified if necessary.

In the modification of PWK to obtain a feasible wrist path, traces in PWK_1 and PWK_2 will be analyzed. The infeasibility in PWK may be induced by the kinematic constraints or the obstacles. The kinematic constraints generate the infeasible regions, areas of different configuration(s) and singular areas. Traces in PWK_1 and PWK_2 should be modified so that they are moved away from the infeasible and singular areas and have proper configurations. The modification will follow the criterion described in equation (7); a discussion of how to achieve an optimal modified path can be found in [16]. The algorithm for planning with position modification is formulated as follows:

Algorithm for planning with position modification (PPM). Maintain the orientation specification by modifying the position.

Step 1

Plan a path to follow an arbitrary curve with the specified orientation. Note that the arbitrary curve can be chosen as a straight line or as a curve defined by a polynomial function, etc.

Step 2

Find the mappings of the obstacles in PWK_1 and PWK_2 and enlarge them according to the length of the tool.

Step 3

Find the wrist path of the planned path. Obtain the corresponding traces in PWK_1 and PWK_2, denoted by tr_1 and tr_2, respectively. If tr_1 falls outside the area in PWK_1 bounded by the two planes corresponding to the rightmost and leftmost edges of the mapping of the obstacle, ignore the presence of the obstacle, because it will not affect the feasibility of the wrist path, as described in Section 4.1. If tr_1 falls within that area, check the feasibility of tr_2 in PWK_2, because it is possible that the obstacle may collide with the wrist path.

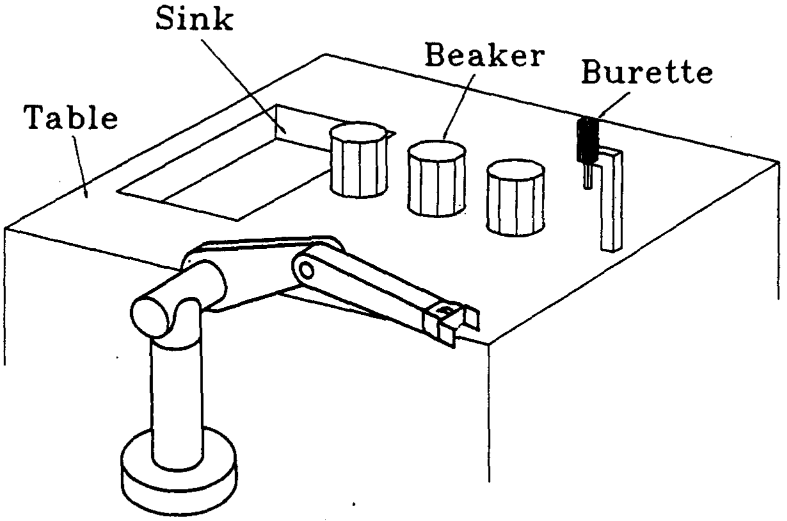

Fig. 8. Laboratory table and PUMA 560 robot manipulator.

Step 4

Modify the infeasible portions of tr_1 and tr_2 to be collision-free and kinematically feasible if possible. The new feasible traces are denoted by $^{n}tr_1$ and $^{n}tr_2$.

Step 5

Find the new wrist orientation $^{n}R_w$ corresponding to $^{n}tr_1$ and $^{n}tr_2$, then obtain the new three traces tr^m in MWK using $^{w}R_e$ derived from $^{n}R_w$ and R_e on the planned path. Check the feasibility of tr^m, if tr^m is infeasible, perform modification when possible.

Step 6

Find the corresponding path in Cartesian space using these five newly-derived traces.

5.2. Planning with orientation modification

For cases with position specification, orientation modification will be used in the planning. To satisfy the position specification, the tool-tip path of a planned path must be collision-free. The orientation will then be modified according to the kinematic constraints and obstacles to generate a feasible path. Since in this case the tool-tip path must be maintained, the minimization will also be based on the wrist path. This minimization is formulated as follows:

Minimize: Deviation between original and modified wrist paths.

Subject to: Position specification, obstacles and kinematic constraints. (8)

Orientation modification must meet a stricter constraint than position modification. With a collision-free tip path, the necessary and sufficient condition for a successful modification is that every point on the path must correspond to at least one feasible wrist position and have feasible orientation(s) in MWK and all points on the tool need to be collision-free [19]. Since the tool-tip position is fixed, the feasible wrist positions will be on the surface of a sphere that is centered at the tool-tip with the length of the tool as the radius. To obtain feasible wrist positions, the sphere must intersect PWK. In addition, at least one of the intersecting points has to be feasible in PWK_1 and PWK_2 and has feasible orientation(s) in MWK that allows the designated tip location to be reached. Furthermore, to obtain a collision-free tool for whatever relative orientation of the tool-tip to the wrist is, a certain margin must be maintained between the wrist and the obstacle. This requirement is met by enlarging the obstacle according to the distance between the obstacle and the point on the path, as discussed in Section 4.2. The algorithm for planning with orientation modification is formulated as follows:

Algorithm for planning with orientation modification (POM). Maintain the position specification by modifying the orientation.

Step 1

Plan a path satisfying the position specification by assigning an arbitrary orientation. Note here that the planned tool-tip path needs to be collision-free. Otherwise, declare an infeasible position specification and exit.

Step 2

For each point on the infeasible portion of the planned path, enlarge the obstacle according to the distance between the tool-tip and the center of the obstacle.

Step 3

Search for a feasible path to replace the infeasible portion under the constraint of the enlarged obstacle derived in Step 2. This feasible path should be with a feasible wrist path and each point on the path be with feasible orientation(s) in MWK and correspond to a collision-free tool. For each point of the infeasible portion the search will be performed upon the intersections of PWK and the sphere, which is centered at the tool-tip and with the length of the tool as the radius.

6. SIMULATIONS

To demonstrate the proposed planning and modification scheme, we shall use it to plan feasible robot paths for a chemistry laboratory. In Fig. 8, a PUMA 560 robot manipulator is used to move the beakers and burette during chemistry experiments. On the laboratory table, there are three beakers placed 5 cm apart. Each beaker is 8 cm high, with a radius of 4 cm. The burette is 10 cm long and its stand is 8 cm away from the right-hand beaker. The sink is located 8 cm from the left-hand beaker. The robot manipulator is placed 40 cm in front of the middle beaker. The length of the gripper is $h_z = 0.04$ m. To simplify the demonstration, the manipulator is assumed to move above the table and the sink and will not collide with them. For illustration, in Fig. 8 the PUMA 560 robot manipulator is depicted at the front, below the table and smaller than the actual size.

Two tasks are simulated: (a) moving the rightmost beaker to the sink and pouring out a solvent, and (b) moving the burette and dropping liquid into each beaker. For the first task, the beaker must be kept in an upright orientation so that the solvent is not spilled. Therefore, the planning

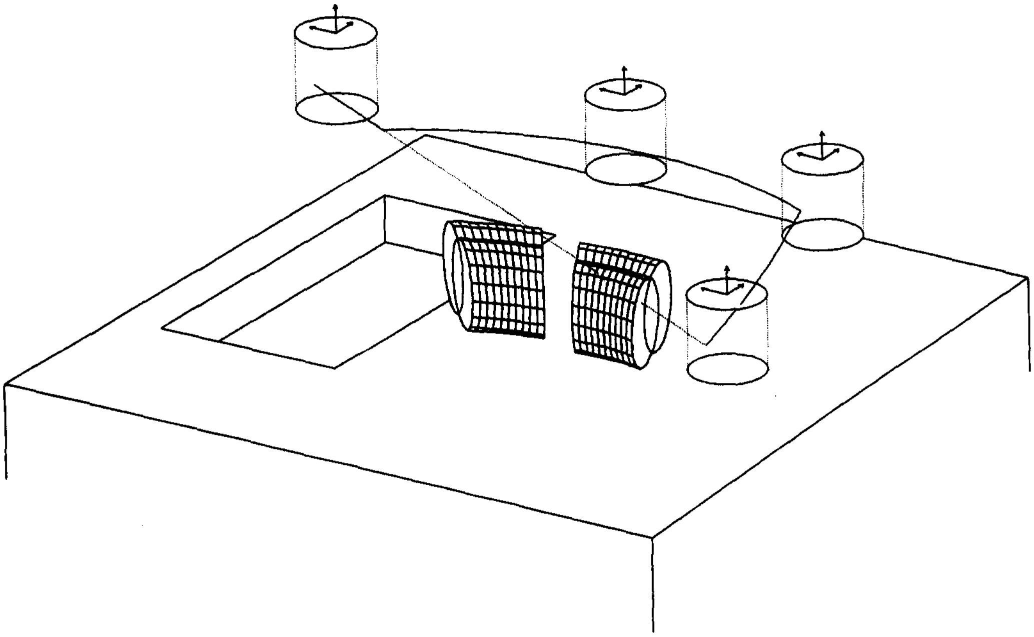

Fig. 9. Moving a beaker to the sink while maintaining an upright orientation.

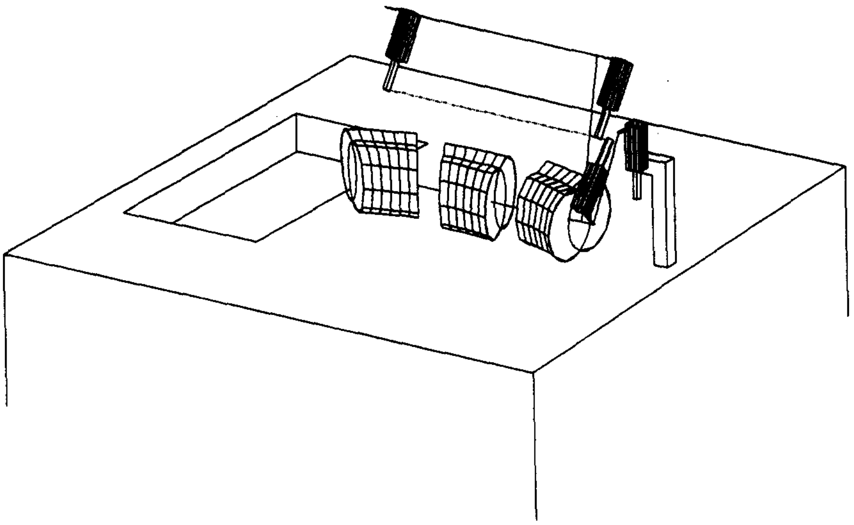

Fig. 10. Moving the burette in a straight-line path reach the beakers.

will be with position modification. In Fig. 9, the original wrist path is designed to be a straight line from the beaker to the sink. It is found that the other two beakers act as obstacles to this planned path. They will be formulated inside the robot workspace using the obstacle formulation algorithm in Section 4.3. Two circles are used to bound the mapping of each beaker in PWK_2, as shown in Fig. 9. In Fig. 9, it can be seen that the mappings of these two beakers are different in shape and orientation, because the two beakers are different distances away from the robot manipulator. In other words, from the viewpoint of the robot manipulator, these two beakers are two different kinds of obstacles, although they are in fact the same size and shape. After applying the PPM algorithm in Section 5.1, the straight-line wrist path is modified to be a curved path, and the beaker is maintained in an upright orientation, as shown in Fig. 9. Note that the wrist path is modified to be quite far away from the obstacles, which bound the beakers. The extra distance is to avoid a larger virtual obstacles resulting from the obstacle enlargement due to the tool and the neighboring infeasible regions induced by the obstacles, as described in Section 4.

The second task involves passing the tip of the burette over the three beakers in a straight line. In Fig. 10, the original path will cause the burette to hit the obstacles formed by the beakers. In order to maintain the tip path of the burette, planning with orientation modification will be employed. By applying the POM algorithm in Section 5.2, a modified collision-free path that follows the designed tip path is derived, as shown in Fig. 10.

7. DISCUSSION AND CONCLUSION

In this paper, a path planning scheme is proposed to designing kinematically feasible paths based on task requirements in the presence of obstacles. The proposed scheme imposes geometrical analysis on the given robot manipulator and obstacles. Geometrical information describing the robot workspace, obstacles and the planned path can then be easily derived. Infeasible paths are modified by applying path modification strategies appropriate to various industrial tasks. Simulations are performed to demonstrate how this scheme can be used to help perform a chemistry experiment. The scheme may also find similar applications in different areas. The success of the proposed scheme allows designers to guarantee that a path planned in a CAD system is feasible before it is sent for execution. Integration between CAD and robots is thus enhanced.

A quantitative analysis on the issue of efficiency is not discussed in this paper, such as complexity analysis or running time comparison with others' approaches. Nevertheless, from the qualitative point of view, the proposed scheme promises to be efficient in both obstacle formulation and path planning. An arbitrarily-shaped obstacle is approximated as compositions of identically shaped elements in the proposed scheme. And the selection of the basic element is according to the structure and movements of the links. Thus less utilizable feasible regions are cast off in the approximation. On the other hand, a close approximation of the obstacle can be achieved by applying the cell decomposition approach, or tree-based technique, etc. In turn, it will be complicated in the search of feasible paths in the workspace. The efficiency in path planning for these approaches may be enhanced by properly re-grouping the feasible regions according to certain criteria [7, 18]. However, those criteria for grouping the feasible regions are in some sense imposing geometrical meaning in the representation of the obstacle similar to our approach, for instance, casting off the feasible regions close to the zig-zag edges of an obstacle. Thus in fact a larger obstacle than the original one is present when it comes to path planning in their approaches. In other words, these approaches also lead to a conservative approximation of the obstacle. Another merit in the proposed path planning scheme is that task requirements are incorporated into the planning, which is not well explored in previous works. We utilize modification strategies to assist in this task-dependent path planning, which involves a smaller local search space and consequently demands less running time.

Because the proposed scheme uses circles of equal radii to approximate the mapping of an obstacle, certain shapes of obstacles may demand much computation time in approximation. For instance, a large number of circles will be needed for a close approximation of the mapping of an obstacle of a long wire. One resolution may be a tradeoff between losing more feasible regions and spending more computation time in approximation. In fact, although the area of a long wire is small, the region affected by a long wire in the robot workspace is significant. On the other hand, the sink in Fig. 8 may represent another type of obstacle of interest. This type of obstacle forms an outside barrier to the movement of the robot manipulator. For instance, the experiment described in Section 6 may demand the robot manipulator holding a beaker operate around the area inside the sink. Under such circumstances, the proposed scheme will view the walls of the sink as a combination of a series of tightly connected objects. Thus, the walls will be divided into a number of obstacles and circles of appropriate radii be used to approximate these obstacles. Consequently, the approximation of the mapping of various kinds of obstacles may deserve more exploration. In other future works, a further analysis on the quantitative aspect in obstacle formulation and path planning of the proposed scheme is demanded. Besides, the obstacles under discussion are solid. The current results may be extended to hollow obstacles, e.g. window-shaped obstacles, which can be found in various industrial applications. In addition, in this paper the end-effector is assumed to be small compared with the links. Long end-effectors may be required in some industrial tasks, however, so modifying the present scheme to cope with long end-effectors would be another worthwhile future work.

Acknowledgements—This work was supported in part by the National Science Council, Taiwan, R.O.C. under grant NSC 82-0422-E-009-403.

REFERENCES

1. Wu, T. H. and Young, K. Y., *Journal of Robotic Systems*, 1994, **11**(8), 703.
2. Paul, R. P., *IEEE Transactions on Systems, Man and Cybernetics*, 1979, **9**(11), 702.
3. Taylor, R. H., *IBM Journal of Research Development*, 1979, **23**(4), 424.
4. Lin, C. S., Chang, P. R. and Luh, J. Y. S., *IEEE Transactions on Automatic Control*, 1983, **28**(12), 1066.
5. Luh, J. Y. S. and Lin, C. S., *IEEE Transactions on Systems, Man and Cybernetics*, 1984, **14**(3), 444.
6. Lozano-Perez, T., *IEEE Transactions on Computer*, 1982, **32**(2), 108.
7. Latombe, J.-C., *Robot Motion Planning*. Kluwer Academic Publishers. Dordrecht, 1991.
8. Lumelsky, V., *IEEE Journal of Robotics and Automation*, 1987, **3**(3), 207.
9. Colson, J. C. and Perreira, N. D., Kinematic arrangements used in industrial robots, *13th ISIR*, 1983, Vol. 20, p. 1.
10. Young, K. Y. and Wu, C. H., *Journal of Robotic Systems*, 1992, **9**(5), 613.
11. Paul, R. P., *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, MA, 1981.
12. Lee, C. S. G. and Ziegler, M., *IEEE Transactions on Aerospace and Electronics Systems*, 1984, **20**(6), 695.
13. Hansen, J. A., Gupta, K. C. and Kazerounian, S. M. K., *International Journal of Robotics Research*, 1983, **2**(3), 22.
14. Lee, T. W. and Yang, D. C. H., *Transactions of the ASME, Journal of Mechanisms, Transmissions and Automation in Design*, 1983, 1.
15. Litvin, F. L., Yi, Z., Castelli, V. P. and Innocenti, C., *International Journal of Robotics Research*, 1986, **5**(2), 52.
16. Shao, K. C. and Young, K. Y., *Transactions of the ASME, Journal of Mechanical Design*, 1994, **116**(1), 36.
17. Xu, Y., Mattikalli, R. and Khosla, P., Two-disk motion planning strategy, *IEEE International Conference on Systems, Man and Cybernetics*, 1991, p. 991.
18. Lozano-Perez, T., *IEEE Journal of Robotics and Automation*, 1987, **3**(3), 224.
19. Jou, C. C., On the orientational feasibility of robot manipulators, *International Conference on Automation, Robotics and Computer Vision*, Singapore, 1990, p. 524.

APPENDIX A

This appendix gives the proof of Theorem 1 in Section 4. In Fig. 7, it is straightforward to see that the minimal area is obtained when the two circular disks intersect at two points and the line connecting these two points is parallel to the short or long side. In addition, the two circular disks should circumscribe these two intersections and one of the sides. Assume that points A and B are two intersections on the long sides, $\bar{A}\bar{B}$ is parallel to the short side, and the distance between $\bar{A}\bar{B}$ and the right short side is l, as shown in Fig. 7. The area $A(l)$, which is the sum of the areas of the two circular disks minus that of the overlapping portion, and its derivative relative to l, $\partial A(l)/\partial l$, can be formulated as follows:

$$A(l) = \pi r_1^2 + \pi r_2^2 - \tan^{-1}\left(\frac{N}{l}\right) r_1^2 - \tan^{-1}\left(\frac{N}{M-l}\right) r_2^2 + \frac{Nl}{4} + \frac{(M-l)N}{4} \tag{A.1}$$

and

$$\frac{\partial A(l)}{\partial l} = \frac{l^2 - (M-l)^2}{4} + \pi\left(l - \frac{M}{2}\right) - \tan^{-1}\left(\frac{N}{l}\right)\left(\frac{l}{2}\right) + \tan^{-1}\left(\frac{N}{M-l}\right)\left(\frac{l-M}{2}\right) \tag{A.2}$$

where $r_1 = \sqrt{(\frac{N}{2})^2 + (\frac{l}{2})^2}$ and $r_2 = \sqrt{(\frac{N}{2})^2 + (\frac{M-l}{2})^2}$.

The minimum of $A(l)$ occurs when $[\partial A(l)/\partial l] = 0$, which results in $l = \frac{M}{2}$. Similarly, it can be shown that the minimal value of $A(l)$ occurs at $l = \frac{N}{2}$ when the intersections are on the short sides. Comparing these two minima, we see that the minimum corresponding to the intersections on the long side is smaller. Therefore the minimal area that fully covers the rectangle is obtained when the two circular disks are of equal diameter and circumscribe two end points of the short side and the middle points of the two long sides.