DESIGN FOR MAINTENANCE BY CONSTRAINED MOTION PLANNING

Tsai-Yen Li
Computer Science Department
Stanford University
Stanford, California

Hsuan Chang
Computer Graphics & Systems Program
GE Corporate R & D Center
Schenectady, New York

ABSTRACT

In designing an assembly product, maintainability is a crucial issue that determines whether certain parts can be accessed for routine maintenance. In the past its study has been largely manual and labor intensive. Either by using physical mockup or computer animation with CAD models of a design, the task relies on human to provide an access path for the part. In this paper we present an automated system that replaces the traditional manual process. By applying results from and developing extensions to the motion planning research in robotics, we demonstrate that this automated system is an effective addition to CAD systems. We specifically describe the extension that accounts for special motion constraints imposed by designers. We also present experimental results from applying such a system to problems in industrial applications. The results show that such an system can not only reduce the turn-around time for checking maintainability of a product but also provide more accurate feedback to the designers on potential maintenance problems.

INTRODUCTION

For many assembly products, certain parts need to be replaced on a regular basis. For example, parts in an aircraft engine need to be replaced after certain amount of service time. This process usually takes place between two scheduled flights. To reduce the amount of time for this maintenance service, direct access paths for these parts without removing other irrelevant parts are preferred. These parts are usually called *Line-Replaceable Unit* or *LRU*. In designing such an assembly, it is crucial to ensure that enough room is kept for an LRU to come out of an assembly. Therefore, after an assembly is designed, the designers or maintenance engineers need to go through a process called *assembly maintainability study* to ensure that there exists an access path for the LRU. In the past, such a process calls for a physical mockup of the assembly being designed, and the path for removing the LRU is found by physically moving the mockup part. However, physical mockups are expensive and time-consuming. Furthermore, it is difficult to capture an LRU's motion that effects its access for future references.

More recently, as more and more assemblies are designed using CAD systems, the trend is toward using more computer animation and less mockup. With computer animation, a removal path can be systematically documented or its swept volume can also be used to constrain possible future design modification. However, this trend presents a tremendous new challenge to the designer as well. For example, moving a CAD model on a computer monitor to find a collision-free path is not an easy task because of the lack of true 3D visualization in general and the lack of force feedback in specific. More importantly, in such a manual animation system, the user needs to come up with a rough path first and then inputs it in the computer system to check for possible collisions at discrete points along the proposed path. Since this collision check process typically is not real-time for complex assemblies, the user needs to wait for a collision report. When a collisions is detected, the user needs to manually modify the path to continue the process. The revised path, however, may result in more collisions. Therefore, this computer-aided manual process is at best a tedious and time-consuming task.

In maintaining an assembly, certain preferences or constraints are often imposed on the motion of an LRU in order to perform effective and safe operations. For example, the designers of an assembly may prefer to move an LRU through certain passages instead of the others because of, for instance, the ease of operation. The designers may also require that in maintaining an assembly, an LRU can only be rotated around certain axis due to safety reasons. In both the mockup or the computer-aided manual methods, such constraints, however, can not be effectively enforced and documented. Especially when studies are performed by other engineers, there is no precise and comprehensive way to specify such constraints to them. Consequently, to generate motions that satisfy such constraints, the designers need to go through either time-consuming communication or several trial-and-error iterations to verify the validity of an

LRU's motion.

In this paper we propose a new approach to automate the study of this maintainability problem. We develop an extended algorithm that employs results from robot motion planning and other fields. We demonstrate an automated system solving the assembly maintainability problem with a convenient way for the designers to specify possible motion constraints for an LRU. In the next section we analyze related motion planning algorithms as they pertain to maintainability problems. Then, we outline our approach to automate maintainability studies. We also present and discuss the results from applying this automated system to different industrial examples. In the last section, we concludes the paper by speculating on how research in motion planning may benefit other disciplines further.

RELATION TO RESEARCH IN MOTION PLANNING

Robot Path Planning

The maintainability problem resembles the robot path planning problem that has been extensively studied in the past two decades. In robotics, one of the goals for path planning research is to automate the trajectory generation process traditionally performed by human operators. Other goals include generating trajectories for autonomous robots that operate in remote or hazardous environments. For example, the path planner described by Faverjon and Tournassoud (1990) is used to compute collision-free paths of an 8-dof manipulator among cooling pipes in a nuclear plant. Recently, results of path planning research are applied to wider range of applications. For instance, Graux et al. (1992) describes a planner that generates paths of a 5-dof riveting machine to assemble portions of an airplane fuselage. Koga et al. (1994) adapts path planning technologies to automatically generating human motion sequences for compute animation.

There are many literature surveys on general motion planning (e.g. (Hwang and Ahuja, 1992) and (Latombe, 1991)). What we are interested in maintainability studies maps into the so-called "Piano Movers' Problem" or "Find Path Problem." This problem is described as finding a collision-free path for an object connecting its initial configuration to some specified goal configuration in a 3D environment. The object has 6 degrees of freedom (dof). Thus, its motion can be embedded in a six-dimensional space called *Configuration Space* or *C-space*. In this C-space, a configuration of the object is represented as a point while the obstacles in the 3D workspace are mapped into forbidden regions, called *C-obstacles*. Therefore, the problem can be translated into finding a path connecting the initial and goal points in the C-space while avoiding collisions with the C-obstacles.

In general, depending on the representation of a C-space, two different approaches can be found in the literature. One direction of research computes an exact boundary representation of the C-obstacles and arranges the geometric features of the boundary (e.g. facet, edge, etc.) in a graph that is used to search for a path (Schwartz and Sharir, 1983). This type of exact algorithms will either find a path or declare with proof that there is no such path. However, most of them typically are too costly to apply to real world problems because of the complexity of the C-obstacle boundary.

The other direction of research uses only discrete points with some user-specified resolution to search for a path in the C-space. Such algorithms do not deal with the complexity of C-obstacles directly but may fail to find a path that otherwise exists with a higher search resolution. Although they do not use the C-obstacle boundary information to search for a path, they might compute the C-obstacle regions before searching the C-space to facilitate collision checks. For problems involving robots/objects with fewer dof (e.g., less than 5), it is feasible to compute an explicit representation of the C-obstacles. For example, Lozano-Pérez (1983) proposes an efficient algorithm to compute the C-obstacle for a polygonal object moving in a 2D environment. However, for more dof cases (e.g. dof > 4), it is more difficult to compute such C-obstacles due to the complexity of of C-obstacle boundary and the limitation of available memory and processing speed with the current computer technology. Nevertheless, there are practical approaches that try to sample the C-space at a discrete resolution (Donald 1987). There are also approaches that incrementally compute the C-space on demand during the path-searching process (Lozano-Pérez, 1991); but such approaches are feasible only if the degrees of freedom for the object can be easily decoupled. For applications involving many degrees of freedom, computing an explicit representation of the C-space becomes impractical even with on-demand computation or with coarse resolutions simply due to the curse of dimensionality. Thus, planners designed for these applications typically check collisions only when a configuration is visited during the path-searching process (e.g. (Barraquand and Latombe, 1990)).

Many heuristics has been proposed in the literature to speed up the search for a path in a C-space (see (Hwang and Ahuja, 1992) for a detail survey). One notable technique is demonstrated in the so-called *Randomize Path Planner (RPP)*, proposed by Barraquand and Latombe (1991). This technique aims at solving problems involving many dof (typically for dof > 4). It takes a probabilistic approach to take advantage of the fact that in a higher dimensional C-space, there usually exist many paths. Although the size of the C-space is large, if the objective is to find just one path, the probability of succeeding in finding one is relatively high.

Assembly Sequencing Planning

A similarity exists between what assembly planning offers and what maintainability planning expects. Several influential research in assembly planning have been published in the literature (e.g. (Wilson and Latombe, 1992), and (Lozano-Pérez and Wilson, 1993)). Both assembly planning and maintainability planning try to find the (dis)assembling motion of parts. However, a close examination reveals significant differences: assembly planning focuses on capturing the (dis)assembly sequencing information that is implied by existing contact information between parts. The motions of parts are usually limited to only one or two dof so that the problem remains tractable. On the other hand, maintainability planning focuses on the problem of how to remove a single part by allowing complex removal motions but without removing other parts. Contact information may be absent. Even if it exists, such information may also be irrelevant to the intended removal motion. Despite the differences, the result of maintainability planning research potentially can benefit assembly planning research by allowing parts to move through a more complex trajectory when simple motions fail.

Characteristics of Maintainability Problem

Despite the similarities between robot path planning and maintainability planning, maintainability studies call for special attention to the following aspects of motion planning: **model complexity**, **resolution dependency**, and **motion constraints**.

First, CAD models from industrial assembly design typically are more complex than models used in robotics. In robotics, it is acceptable (sometimes desirable) to simplify robot models either by approximating them with geometric primitives (e.g., parallelepipeds, cylinders, and spheres) or by using fewer polygons to represent curved surfaces. The approximated models usually are more conservative than the real objects so that the found path will not lead the robot too close to obstacles. As a result, more efficient but less accurate collision detection algorithms can be employed to reduce the time spent in planning. In contrast, for maintainability problems, contact situations are allowed, and parts usually are more complex. Realistic model representation and accurate collision detection are needed to capture the interaction between the moving part and its environment so that a feasible path is not left out simply because of the accuracy of the models or the employed collision detection algorithm. Therefore, in maintainability studies, the model complexity and the related cost in detecting collision are usually much higher than those encountered in robotics.

Second, modern designs strive to be compact to achieve space efficiency. As a result, assemblies are more densely packed. This density directly translates into a C-space crowded with C-obstacles that divide the C-space into many regions possibly connected by narrow passages. For such cases, planners that employ search algorithms with a fixed resolution will be less efficient because of the reduced probability in finding critical but small openings that lead an LRU toward the goal. In the worst case, they may fail to find a path that otherwise exists with a finer resolution. On the other hand, using finer resolution is not always desirable because of the increased cost for traveling a fixed distance. One can envision cases where various resolutions are needed at various stages of a search to optimize the performance of a planner. However, since geometric and topological information of the C-space is not explicitly available for high dof applications, some heuristics need to be employed to adjust the search resolution at run time during the search process.

Third, in robotics, motion constraints are imposed mostly due to mechanical constraints such as mechanical stops. Such constraints can be easily described by providing a bound on the valid joint angles. For assembly maintenance problem, however, sometimes there are preferred regions through which an access path should be found because of the realistic concerns of the assembly such as the size of the moving object, support regions for tools, heat sources to avoid, or simply access convenience. The example shown in Figure 1 illustrates that while path (a) is more likely to be found in a potential-guided planner, path (b) is a more desirable path because of easy access or safety reasons such as avoiding heat source. We call this type of constraints, which are usually specified in the workspace of an LRU, *translational constraints*. An obvious solution to input such constraints is to place artificial obstacles in the workspace to judiciously block the undesirable passages. But this requires the user to anticipate such undesirable passages, which may not be an intuitive task for complicated 3D assemblies. Another type of constraints that are not common in robotics are the constraints on the rotational degrees of freedom for a free-flying object. For example, it may not be acceptable to rotate an LRU around certain axis that may cause liquid in an LRU to overflow. We call this type of constraints *rotational constraints*. In the current computer-aided manual approach, however, there is no precise way for a user to input access motions that satisfy these constraints.

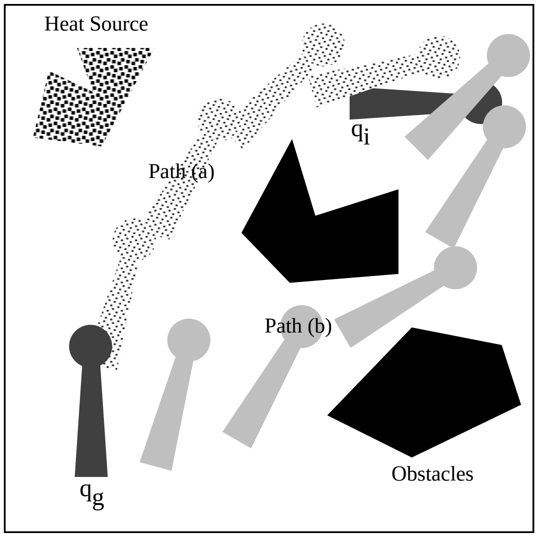

FIGURE 1: TRANSLATIONAL CONSTRAINTS

For the first characteristic about model complexity and precise collision checking, we employed the hierarchical sphere-tree algorithm reported by Quinlan (1994) to reduce the number of calls to the exact collision checking routine for two polygons. The exact collision checking routine is modeled after Gilbert and Johnson (1985). It allows the user to specify desirable accuracy up to double floating point precision. For the second characteristic, we use an adaptive refinement heuristic to dynamically change the search resolution in a C-space as proposed by Chang and Li (1995) in order to overcome the difficulty of densely packed C-obstacles with narrow openings. In the next section we focus on the third characteristic on incorporating motion constraints into a planning algorithm such that a found path satisfies the constraints specified in the maintainability problem.

CONSTRAINED MOTION PLANNING

By incorporating the solutions to the special problems indicated above, we have developed an automated maintainability planning system. The basic planning algorithm is based on the RPP algorithm. We briefly review the key ideas of the RPP algorithm and then elaborate our extensions on satisfying two types of motion constraints.

The RPP Algorithm

The RPP algorithm uses a combination of randomized search and potential field techniques to generate collision-free paths for robots with many dof. It computes goal-oriented potential fields as a guide to searching for a path. These potential fields are computed in the workspace for some points (called control points) chosen in the moving object. For each control point at the goal configuration, we use the wave expansion algorithm by Latombe (1991) to build a potential field that has a unique global

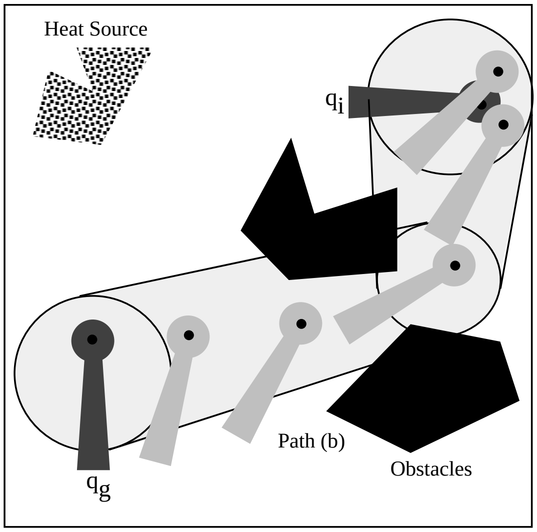

FIGURE 2: CONSTRAINT VOLUME

minimum at the control point. A heuristic potential function U is then defined over these potential fields such that $U(q) = 0$ if and only if $q = q_g$, where q is any configuration and q_g is the goal configuration. An example of the potential function could simply be the sum of the potentials for the control points at a configuration. Let $C(q)$ be a collision checking function that returns `true` if the moving object at q collides with obstacles and returns `false` otherwise. The moving object follows the gradient of U from the current configuration q to its neighbor configuration q' if $U(q') < U(q)$, $C(q') = \texttt{false}$, and $q \neq q_g$. If the gradient motion reaches a local minimum of U, a preset length of random walk is performed. At the end of the random walk, a gradient motion resumes. This process (pair of gradient and Brownian motions) repeats until the global minimum – goal – is reached or a preset number of iterations is reached. In the latter case, a backtrack step is performed to retract part of the path found so far, and the search restarts.

Translational Constraints

There are various ways to impose translational constraints during the search. One obvious way is to block all undesirable passages with artificial obstacles. However, in 3D environments, it may not be intuitive for the user to specify such obstacles. The user needs to know the dimension and position of the moving object as well as the size of the undesirable passages in order to block them effectively. This process is tedious when the number of passages is large. In addition, there is no guarantee that such artificial obstacles will not prevent the planner from finding an otherwise feasible path, e.g., a path that uses the undesirable passages to adjust the orientation of the LRU.

When an assembly is designed, a rough access path for an LRU usually has been considered by the designers. Therefore, an intuitive way to specify a translational constraint is by specifying a rough path traversing through the designed openings. In our system, we build an interface for the user to specify such translational constraints by confining the LRU in a so-called *constraint volume*. This volume is computed from a sequence of spheres with various diameters connected smoothly by trimmed cones. The center of each sphere is placed, by the user, at a critical branching point in the workspace and the radius of the sphere is specified such that hazardous regions, such as heat sources, can be safely avoided. The motion of the LRU is then confined to this constraint volume during the search. More specifically, we choose a certain number of points (called constraint points) in the LRU that is limited to stay inside the constraint volume while the rest of the LRU can move out of the volume to, for instance, make orientation changes. These points could be critical surface landmarks, the centroid, or some other references of an LRU decided by the user. For example, in Figure 2, the constraint volume (depicted in light grey) is specified by placing a set of circles (spheres for 3D) of various sizes linked by cones. The small dots on the moving object indicate the constraint points. With the help of such constraint volume, desirable paths such as path (b) (also shown in Figure 1) can be effectively found.

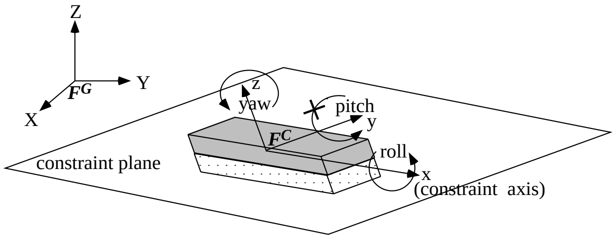

FIGURE 3: ROTATIONAL CONSTRAINTS

Although the constraint volume is a convenient way for a user to specify translational constraints, requiring constraint points to stay in the volume could actually introduce extra local minima that trap the LRU during the search in the RPP algorithm. Consequently, the running time becomes longer due to the resultant ineffective potential fields. To cope with this difficulty, we build the potential fields with the constraint volume taken into account. In the wavefront expansion algorithm that computes the potential fields, we first propagate the potential values only in this constraint volume. Then from the boundary of this volume we propagate the potential values to the rest of the free space. With this modification, the potentials in the volume are locally lower than those outside. As a result, the search is effectively biased to the configurations that position the constraint points inside the volume.

Rotational Constraints

In addition to translational constraints, there could be rotational constraints that need to be taken into account in maintainability studies. These constraints typically can be described as limitations on the rotations about certain axes, called the *constraint axes*. For example, the user can specify that the LRU is allowed to rotate around a certain axis only or that certain axis of the LRU is allowed to move in a plane (we call it the *constraint plane*) only in order to, for example, level the LRU while moving it. In the former case the dof of the LRU is reduced to four while the in the latter case the dof is reduced to five. This axis could be any arbitrary axis that is not necessarily aligned with any axis of the coordinate frame that the LRU is referenced to.

In our system, a configuration is described as a six-parameter variable, where three parameters represent the translation of a rigid-body displacement and the other three represent the orientation. We represent the orientation by (*roll*,*pitch*,*yaw*), which corresponds to three consecutive rotations about the fixed X-, Y-, and Z-axes of the global coordinate frame F^G. The coordinates of the moving object is described with respect to a local coordinate frame F^L. A configuration q of F^L can also be described as a homogeneous 4×4 transformation $T(q)^G$ with respect to F^G.

To effect rotational constraints, we create a new local reference frame F^C, where one of its axes is aligned with the constraint axis. For example, in the 4-dof case, the x-axis of F^C is aligned with the constraint axis. In the 5-dof case, since the user essentially would like to confine the motion of an axis on a plane, we align the constraint axis and the axis perpendicular to the constraint plane with the x-axis and z-axis of F^C, respectively (as shown in Figure 3). A configuration transformation $T(q)^G$, as represented in F^C, becomes $T(q)^C = R_L^C T_G^L T(q)^G$, where R_L^C is the rotational transformation from F^L to F^C, and T_C^G is the transformation from F^G to F^L. When represented in F^C, the three orientation parameters (*roll*,*pitch*,*yaw*) for the initial configuration q_i become (0,0,0) because of the axis alignment. Any orientation changes are then specified with respect to F^C to effect the rotational constraints. For example, in the 4-dof case, we only allow the *roll* parameter to vary during the search while in the 5-dof case, we allow both *roll* and *yaw* to change to effect the rotation about the constraint axis as well as the rotation about the axis perpendicular to the constraint plane.

Since the configurations of an LRU now are specified with respect to F^C, before update the coordinates of the vertices on the LRU for collision checking purpose, we need to transform the configuration being considered back to F^G. That is, for a vertex P^C of the LRU under a configuration q, we compute its coordinate with respect to F^G by the following equation: $P^G = T_L^G R_C^L T(q)^C P^C$. The configurations along a path are also updated with a similar transformation before the path is output.

RESULTS AND DISCUSSION

Implementation

The proposed automated system for maintainability studies has been implemented as part of an animation software package called *Product Vision* developed at the GE Corporate R&D Center. This planning system has been implemented in the C language and runs on various machine architectures. The running times reported in the next section are on an SGI RealityEngine workstation, with two R4400 processors at 150 MHZ and 128 MBytes of main memory. The core of the system is an general motion planner modeled after the Randomized Path Planner developed at the Stanford University. The input to this system is a set of polyhedral models describing the geometry of the assembly and a data file describing user-specified parameters such as the initial search resolution, constraint axes, etc. The output of the system is a smooth access path of the LRU that can be described in various path formats.

A swept surface outlining the physical space required of the LRU moving through the found path is then generated using the Swept Volume algorithm by Schroeder et al. (1994). This swept surface is used to help the designers to impose further constraints on other add-on parts/subassemblies in later stages of a design.

Results

In this section we show several experimental results of our automated maintainability system. These examples are all taken from real industrial designs.

Translational Constraints: We used several cases to study the effectiveness of using constraint volumes to specify translational constraints. Figure 4 shows a case where the LRU is located in a cluttered environment under a set of pipes. An opening between the pipes is designed intentionally for accessing the LRU from the maintenance operator's point of view. The LRU in its initial configuration is orthogonal to this opening (as shown in Figure 4(a)). A study was basically performed to evaluate whether there is enough room for the object to make such an orientation change to get out of the designed opening.

Our study shows that without using any translational constraints, the planner found a feasible path without using the designed opening (Figures 4(e) through 4(g)). Such a path, although feasible, was not expected by the designer. If this path is actually an acceptable one, the designer can modify the design to eliminate the unnecessary opening. Otherwise, the user needs to specify some translational constraints to guide the motion of the LRU through the designed opening as we did. We specified three spheres: two at the initial and goal configurations, respectively, and one near the center of the opening. The system hence forms a constraint volume that cuts off the path shown on the right side of Figure 4. Figures 4(b) through 4(d) show the path found after applying this constraint volume.

Rotational Constraints: We have studied the effectiveness of rotational constraints in several cases. Figure 5 illustrates one such study. In this design, the LRU is surrounded by several pipes as shown in Figure 5(a). The requirement for this study is to show that the given design allows the LRU to be removed while maintaining its mouth sub-assembly's orientation (facing up). To meet this requirement, we performed a comparative study where, as shown on the left side of the figure, a full 6 dof motion path was generated. Clearly without any rotational constraints, the LRU tumbles all over along the generated path, violating the given requirement. On the right side of the figure, we show that with a rotational constraint enforced along the axis of the LRU, the system generated a 5 dof motion path that meets the requirement.

Figures 5(b) and 5(c) show two intermediate configurations of the part along the 6 dof path; 5(e) and 5(f) show the corresponding 5 dof motions. In 5(d) we show a multiple exposure of the LRU along the 6 dof path. In 5(g), the corresponding 5 dof multiple exposure illustrates a much more refined motion where the mouth piece maintains its orientation during the access.

Discussion

Overall, the above examples are difficult ones for the traditional, CAD-based move-and-detect type of manual maintainability systems. With the new automated system, we have obtained tremendous productivity gains over the old systems. In the past a study typically takes a few days of tedious interactive work. Now, it typically takes only a few minutes to several hours to generate an access path with the automated system if it can find one. In the automated system the user only needs to specify some parameters to set up a study and the rest is taken care by the system automatically. Therefore, the user can actually run several studies simultaneously. Depending on how complex a study is, the productivity gain ranges from several factors to an order of magnitude.

In addition to the productivity gains, the automated system also serve as a more precise and powerful tool for maintainability studies. In some retrospective studies, we have found that the automated system is capable of finding paths that the user failed to find. In other words, we have shown that in complex environments there could be paths that are not intuitive to human by viewing the models on a computer screen. Our automated system, in contrast, is equally good at capturing these non-intuitive paths. When the user fails to find a path, the designers will then revised the designs by reserving more space for the LRU, and then repeat the maintainability study for the new design. Therefore, in these cases, the potential productivity gains of using a more precise automated system like ours could be even higher.

However, the system is not without problems. In certain cases (that are not included in this paper) we experienced pronounced effects from the probabilistic nature of the RPP algorithm. Some of our studies are non-conclusive after several days. On the other hand, if the cases were studied using the old manual system, the user might spend about the same amount of time before he/she can conclude that there is no feasible path. Nevertheless, it is more desirable if the automated system can conclude such cases with a probability analysis and suggest necessary design modifications to the designers. To address this problem, we are investigating other techniques such as the Roadmap algorithm by Kavraki and Latombe (1994) and a planning algorithm allowing parts in the environment such as potential obstructive parts to be moved to make space for the LRU.

In the current system, the rotational constraints are specified by forbidding the rotations about a certain axis. However, in real maintenance tasks it may be more desirable to specify a range of forbidden/permitted rotational angles about the axis. This can be easily incorporated into the current system by limiting the corresponding parameter in a configuration to some appropriate intervals. However, there are other constraints that are difficult to describe in the current system without providing special constraint routines to the system. For instance, a cup-like part with liquid in it can be tilted along any axis as long as the liquid does not overflow. To specify such constraints, the user needs to either provide a special routine to eliminate illegal configurations of the LRU or explicitly specify all legal ranges of orientation to the system. However, this type of constraints currently are not considered in the implemented system.

CONCLUSION

We have developed an automated planning system for assembly maintainability studies by incorporating results from motion planning research in robotics. To cope with the special problems encountered in maintainability studies, we developed a planning system that allows the user to conveniently specify translational and rotational constraints to the motion of an LRU. We show how effective this automated system is by running it on a number of complex examples from the industry. The results show that it not only solves otherwise-difficult problems but also lends significant productivity and quality improvements over the existing computer-aided manual process. In the context of concurrent engineering, such a system can benefit the designers by reducing the turn-around time of verifying the maintainability of an assembly. With these results, we are confident that this automated maintainability system points to a new direction of applications for motion planning research in robotics. We hope that it will serve to provide additional driving force for motion planning research in general.

REFERENCES

Barraquand, J. and Latombe, J.-C., 1990, “A monte-carlo algorithm for path planning with many degrees of freedom,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, Cincinnati, OH, pp. 1712–1717.

Barraquand, J. and Latombe, J.-C., 1991, “Robot motion planning: A distributed representation approach,” *The Int. J. of Rob. Res.*, 10(6).

Chang, H. and Li, T.-Y., 1995, “Assembly maintainability study with motion planning,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, to appear.

Donald, B., 1987, “A search algorithm for motion planning with six degrees of freedom,” *Artificial Intelligence*, Vol. 31, pp. 295–353.

Faverjon, B. and Tournassoud, P., 1990, *Rob. Res.*, The MIT Press, Cambridge, Massachusetts, Vol. 5, pp. 425–433.

Gilbert, E. G. and Johnson, D. W., 1985, “Distance functions and their application to robot path planning in the presence of obstacles,” *IEEE Tr. on Rob. and Aut.*, RA-1(1), pp. 21–30.

Graux, L. et al, 1992, “Integration of a path generation algorithm into off-line programming of airbus panels,” *Aerospace Automated Fastening Conference and Exp.*

Hwang, Y. K. and Ahuja, N., 1992, “Gross motion planning - a survey,” *ACM Computing Surveys*, Vol. 24, pp. 220–291.

Kavraki, L. and Latombe, J.-C., 1994, “Randomized preprocessing of configuration space for fast path planning,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, San Diego, CA, pp. 2138–2145.

Koga, Y., Kondo, K., Kuffner, J., and Latombe, J.-C., 1994, “Planning motions with intentions,” *Proc. of SIGGRAPH’94*, Orlando, Georgia.

Latombe, J.-C., 1991, *Robot Motion Planning*, Kluger Academic Publishers, Boston, MA.

Lozano-Pérez, T., 1983, “Spatial planning: A configuration space approach,” *IEEE Tr. on Comp.*, 32(3), pp. 108–120.

Lozano-Pérez, T., 1991, “Parallel robot motion planning,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, Scramento, CA, pp 1000–1007.

Lozano-Pérez, T. and Wilson, R., 1993, “Assembly sequencing for arbitrary motions,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, Kyoto, Japan.

Quinlan, S., 1994, “Efficient distance computation between non-convex objects,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, San Diego, CA.

Schroeder, W. J., Lorensen, W. E., and Linthicum, S. L., 1994, “Implicit modeling of swept surfaces and volumes,” *Proc. IEEE Visualization’94*, pp. 40–45.

Schwartz, Z., and Sharir, M., 1983, “On the Piano Movers’ Problem: II,” *Advances in Applied Mathematics*, Vol. 4, pp. 298–351.

Wilson, R. and Latombe, J.-C., 1992, “On the qualitative structure of a mechanical assembly,” *Proc. IEEE Int. Conf. on Rob. and Aut.*, Nice, France.