Decomposition-based Motion Planning: A Framework for Real-time Motion Planning in High-dimensional Configuration Spaces

Oliver Brock Lydia E. Kavraki
Department of Computer Science
Rice University, Houston, Texas 77005
email: {oli, kavraki}@cs.rice.edu

Abstract

Research in motion planning has been striving to develop faster planning algorithms in order to be able to address a wider range of applications. In this paper a novel real-time motion planning framework, called decomposition-based motion planning, is proposed. It is particularly well suited for planning problems that arise in service and field robotics. It decomposes the original planning problem into simpler subproblems, whose successive solution empirically results in a large reduction of the overall complexity. A particular implementation of decomposition-based planning is proposed. Experiments with an eleven degree-of-freedom mobile manipulator are presented.

1 Introduction

Recent advances in the area of robot motion planning have resulted in the successful application of these techniques to diverse domains, such as assembly planning, virtual prototyping, drug design, and computer animation. Much of the progress can be attributed to the introduction of probabilistic roadmap techniques [10] and their various extensions [1, 2, 8, 9].

Despite these advances, however, some areas of application have still remained out of reach for automated planning algorithms. Applications requiring robots with many degrees of freedom to operate in highly dynamic and unpredictably changing environments fall into that category. To operate robustly and safely in dynamic environments the ability to modify the planned motion in real time is necessary. The planning techniques for high-dimensional configuration spaces described in the literature, however, do not generate plans in real time.

In this paper a new planning paradigm is presented addressing these issues by decomposing the planning problem and applying appropriate planning algorithms to the respective subproblems. This results in a real-time planning algorithm in high-dimensional configuration spaces [3]. The planning paradigm is well suited for planning problems of average difficulty, in which a certain amount of clearance to obstacles along a solution path can be assumed. Such planning problems occur frequently in the areas of field and service robotics. The proposed algorithm differs from a probabilistic approach to real-time path planning found in the literature [14], where a configuration space roadmap is computed in the absence of obstacles and portions of that roadmap are subsequently invalidated by the introduction of obstacles.

2 Decomposition-based Motion Planning

Decomposition-based planning is a motion planning framework addressing motion planning problems of average complexity, as can be encountered in field and service robotics. In those application areas a minimum clearance to obstacles can be assumed.

2.1 Motivation

Most of the motion planning approaches represent the connectivity of the free space in high-dimensional configuration spaces, which are designated by $\mathcal{C}$. The underlying assumption of decomposition-based motion planning is that for many applications relevant connectivity information can be computed and represented more easily in a low-dimensional space, while the motion of the robot must be generated in a high-dimensional space, namely the configuration space associated with the robot. This naturally leads to a decomposition of the overall planning task into a high-dimensional and a low-dimensional subproblem.

The high-dimensional solution in $\mathcal{C}$ is connected to the low-dimensional space $\mathcal{W}$ via the workspace volume V swept by the robot along its trajectory defined in $\mathcal{C}$. In other words, a path or trajectory in a high-dimensional space can be represented as volume in the low-dimensional workspace. Decomposition-based planning uses this as the link between the two spaces to divide the planning task.

2.2 Framework

Consider a planning problem P for a robot R in a configuration space $\mathcal{C}$ of dimension d, with an initial configuration $\mathbf{q}_{init}$ and a final configuration $\mathbf{q}_{goal}$. Assume there exists a path τ from $\mathbf{q}_{init}$ to $\mathbf{q}_{goal}$ entirely in the free space $\mathcal{F} \subseteq \mathcal{C}$. Then let V_τ denote the workspace volume swept by the robot R along the path τ. For now we consider the workspace $\mathcal{W}$ to be the Euclidean space $\mathbb{R}^3$. Furthermore, let $H(\tau)$ denote the set of paths homotopic to the path τ. The workspace volume $V_{H(\tau)}$ is then defined as $V_{H(\tau)} = \bigcup_{\sigma \in H(\tau)} V_\sigma$, representing the combined workspace volume swept along all paths homotopic to τ.

Let us assume that there are n homotopically distinct solution paths $\tau_i, 1 \leq i \leq n$ to the planning problem P. The set of all solution paths $S(P)$ to P is given by $S(P) = \bigcup_{1 \leq i \leq n} H(\tau_i)$. For a solution path τ, the relation $V_\tau \subseteq V_{H(\tau)} \subseteq V_{S(P)}$ must hold.

We define $V_\tau^\delta = V_\tau \oplus b(\delta)$, where $\oplus$ denotes the Minkowski sum and $b(\delta)$ denotes a ball of radius δ, to represent the volume swept by the robot along the path τ grown by δ. The planning problem P is said to be δ-hard if there exists a path $\tau \in S(P)$ such that $V_\tau^\delta \subseteq V_{S(P)}$. This means that at every point along the path τ the robot has at minimum a clearance of δ from the closest obstacle. The decomposition-based planning approach presented here addresses planning problems that are δ-hard.

We want to decompose the planning problem P into two subproblems, P_1 and P_2. The planning problem P_1 can be defined as determining a workspace volume T, called *tunnel*, such that $V_\tau \subseteq T$ for at least one solution path τ. Since τ and therefore V_τ are not known, however, a simplified criterion has to be used to ensure the tunnel T is computed in a manner that $V_\tau \subseteq T$. Such a criterion is called *complete* if for every solution path $\tau \in H(\tau_i) \subseteq S(P)$ the relation $V_\tau \subseteq T$ holds. Note that T might also represent paths σ that are not solution paths, i.e., $\sigma \notin S(P)$, but $V_\sigma \subset T$.

Alternatively, an *incomplete* criterion can be used, meaning that there are solution paths τ such that $V_\tau \not\subseteq T$. Such a criterion can be computed much more efficiently, but introduces incompleteness. In choosing an incomplete criterion the tradeoff between completeness and efficiency needs to be considered carefully. In the remainder of this paper we will be concerned with methods that find a solution path $\tau \in H(\tau_i)$ if $V_\tau^\delta \subseteq V_{H(\tau_i)}$, for a given value of δ. These methods are called *δ-complete*.

Once we have obtained a workspace volume T, we define the second planning problem P_2 to consist of finding a path $\tau \in S(P)$ *using* T to aid the planning process. Again, various planning methods can be employed to accomplish this task and their performance can vary. A method to solve P_2 can determine a path τ such that $V_\tau \subseteq T$, or alternatively use T solely for connectivity information to determine a path τ such that $V_\tau \subset \mathcal{F}$, but not necessarily $V_\tau \subseteq T$.

The real-time performance of the proposed motion planning algorithm is the result of a tradeoff of completeness for efficiency. This tradeoff is based in two assumptions made during the decomposition of the original planning problem. Let $\tau \in H(\tau_i)$ represent a solution path to the original planning problem. The approximation of $V_{H(\tau_i)}$ using T assumes that there exists a path $\tau \in H(\tau_i) \subseteq S(P)$ such that $V_\tau \subseteq T \subseteq V_{S(P)}$. This assumption introduces the first loss of completeness. It is not implied, however, that τ necessarily is the solution path obtained when solving P. For most practical algorithms the volume of T is going to be a proper subset of $V_{S(P)}$, i.e., $T \subset V_{S(P)}$. In Section 3.2 a method is presented that allows to extend the amount of free space used beyond T without incurring additional complexity. Finding a solution path τ' then using T introduces a further loss of completeness. Both these tradeoffs allow decomposition-based planning to yield real-time performance.

So far the workspace $\mathcal{W}$ was assumed to be the Euclidean space $\mathbb{R}^3$. It is worth mentioning that the framework can also be applied to $\mathbb{R}^1$, $\mathbb{R}^2$, and $\mathbb{R}^3 \times t$, where t denotes time.

Various methods can be devised to solve the aforementioned subproblems P_1 and P_2 and in the framework of decomposition-based planning. Section 3 introduces a particular choice in the context of motion planning for mobile manipulators. The algorithms are simple and have been shown to work effectively in practice; other methods addressing other application areas need to be evaluated.

2.3 Related Work

Some planning approaches presented in the literature exhibit ideas that are reminiscent of decomposition-based planning. The freeway method [4] can be viewed as one of the earliest approaches based on similar concepts. A particular instance of decomposition-based planning was applied to planning motion for a robot moving in the plane [6]. The idea of decomposing the planning task into capturing a volume in space and imposing a navigation function onto that space can also be found in an approach to planning feedback motion strategies [16]. Here, the volume of free space is computed in configuration space, resulting in larger computational complexity. Other planning approaches use projection to reduce the complexity of the planning problem; these approaches assume that a solution to P_1 of the decomposition automatically is a solution to P_2 [12, 15]. Finally, the idea of dimensionality reduction of the planning problem is of integral importance to the silhouette method [5], where the problem is recursively projected into lower dimensions. This particular approach, however, differs significantly in the way the subproblems are treated.

3 A Decomposition-based Motion Planning Method

In this section the decomposition-based planning paradigm is applied to the motion planning problem of a serial-link mobile manipulator with many degrees of freedom (see Figures 1 and 3). As described in Section 2, the planning problem is decomposed into two subproblems. The first subproblem P_1 of identifying a tunnel T will be addressed by a wavefront expansion algorithm for free space computation. The second subproblem P_2 of determining a solution path in the configuration space will be solved using potential field techniques and a navigation function, computed from the solution of the first subproblem P_1.

3.1 Solving P_1: Wavefront Expansion

The subproblem P_1 consists of determining the workspace volume T, called tunnel, such that the volume V_τ swept by the robot along a solution path τ is contained within T, i.e., $V_\tau \subseteq T \subseteq V_{H(\tau)}$. Here we are particularly interested in motions sweeping a workspace volume tightly enclosed by a tube [7]. This is motivated by the motion capabilities of a free-flying, snake-like robot, as described in the experiments below. We will base the computation of the tunnel T on these tubes. For robots with different motion capabilities the computation of T has to be modified.

In this particular instantiation of decomposition-based planning, the tunnel T will be determined by a wavefront expansion algorithm [13] in the workspace. The algorithm proceeds as follows: We compute the radius r of sphere $\mathcal{B}_s$ centered at the start configuration s of the wavefront expansion by determining the distance to the closest obstacle in the environment. This guarantees that $\mathcal{B}_s$ describes a volume of free workspace. This sphere is inserted into a priority queue, prioritized by the minimum distance between the sphere and the goal location g. With its center p and radius r we store the parent of the sphere, which in this case is the empty set $\emptyset$. If the goal location is designated by g, the priority value according to which the sphere is inserted into the priority queue is given by $\|p - g\| - r$. This represents a best-first planning approach: the sphere nearest to the goal configuration has the highest priority.

The algorithm now iterates until either the initial and final configuration of the robot are connected by a tunnel of free space of given diameter, or the priority queue is empty. Each iteration begins by removing the sphere $\mathcal{B}$ with the highest priority from the queue and inserting into the tree A as a child of its parent. The tree represents the currently explored free space. The surface of $\mathcal{B}$ is randomly sampled; if the sample is not contained in other spheres in the previously explored free space, the spheres centered at those samples are computed. Those spheres are inserted into the priority queue and the process is repeated. Three snapshots of the wavefront expansion are shown in Figure 1. It can be seen that open areas are explored with fewer and larger bubbles than tight spaces.

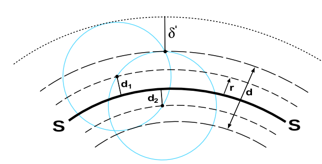

Figure 2: Determining δ

We now argue the probabilistic δ-completeness of this approach to P_1 for tube-like paths. Given the maximum curvature of the the tube κ, a required diameter d of the tube, and a radius r of the surface patch A, we argue geometrically how to determine the value δ. Please refer to figure 2. The solid curve in the center represents the spine S of the tube. The inner dashed lines at distance r from S bound the location of the relevant sample on $\mathcal{B}$; by varying the number of samples n we can guarantee with high probability that a sample will fall within this bound. The outer dashed curves indicate the required amount of free space and hence represent the tube of diameter d. For any given distances d_1, d_2 of the centers of the spheres to S we can minimize δ', as indicated in the figure, by varying the radii of the spheres while maintaining coverage of the tube. The maximum of these minima for all possible distances of the centers to S yields δ. Therefore, the proposed method will with high prob-

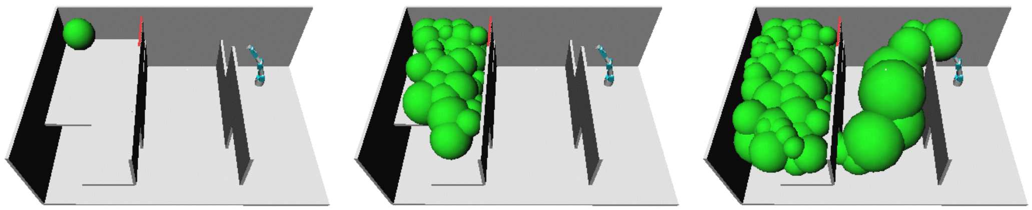

Figure 1: Wave-front expansion to determine a tunnel T.

ability discover a tube-shaped path with diameter d, if there is additional free space of width δ along the tube, i.e., the method is probabilistically δ-complete.

3.2 Solving P_2: Potential Fields

Using the tunnel T computed by solving P_1, we now determine a path τ for the robot. This will be accomplished by imposing a local-minima free potential function on the free space representation determined by the wavefront expansion algorithm. This potential function will result in forces on the robot, causing it to move to its goal location, while avoiding obstacles.

For a robot to react to obstacles in the environment, proximity information needs to be translated into joint motion. Such proximity information can be easily obtained by distance computation in the workspace. A virtual force $\mathbf{F}$ acting on the robot as a result of proximity to obstacles can be computed. Its direction is determined by the line segment connecting closest points on the robot p_r and on the obstacle p_o; the magnitude is inversely proportional to the length of the line segment. The force $\mathbf{F}$ can then be translated into joint torque $\mathbf{\Gamma}$ using the Jacobian J at configuration $\mathbf{q}$ of the robot: $\mathbf{\Gamma} = J^T(\mathbf{q})\mathbf{F}$, where J is the Jacobian of the manipulator at point p_r, the point at which the force acts. This effectively maps the low-dimensional force vector $\mathbf{F}$ from the workspace into the high-dimensional joint space of the manipulator. Using this mapping reactive obstacle avoidance can be achieved.

During the execution of a task by a robot, it is desirable to link reactive obstacle avoidance with task execution. The framework for combining task behavior and obstacle avoidance behavior relies on the general structure for redundant robot control. In this structure the torques $\mathbf{\Gamma}$ that are applied to the robot are computed as follows:

$$\mathbf{\Gamma} = J^T\mathbf{F}_{task} + \left[I - J^T\overline{J}^T\right]\sum_i J_i^T\mathbf{F}_i \qquad (1)$$

[11], where J is the Jacobian of the manipulator at the current configuration $\mathbf{q}$, $\overline{J}$ designates its dynamically consistent pseudo inverse, J_i is the Jacobian in configuration $\mathbf{q}$ at point p_i, $\mathbf{F}_{task}$ describes the forces defined by the task, and $\mathbf{F}_i$ denotes the repulsive forces exerted on the points p_i on the robot by obstacles. The forces attracting the robot to the goal and repulsing it from obstacles are derived from a potential function. The attractive forces act at the operational point, usually the end-effector, and the repulsive forces act at the point on the robot closest to the obstacle.

Equation 1 provides a decomposition of the joint torques into those caused by forces at the end effector ($J^T\mathbf{F}$) or operational point and those that only affect internal motion of the robot $\left(\left[I - J^T\overline{J}^T\right]\sum_i J_i^T\mathbf{F}_i\right)$. This decomposition can be exploited to use task-independent degrees of freedom of the robot for obstacle avoidance in the nullspace. Simple obstacle avoidance without the incorporation of task behavior can be achieved by mapping attractive and repulsive forces to joint torques using equation $\mathbf{\Gamma} = J^T(\mathbf{q})\mathbf{F}$. Here, the forces $\mathbf{F}$ are the combination of forces to accomplish the task $\mathbf{F}_{task}$ and forces $\mathbf{F}_{obst}$ to avoid obstacles: $\mathbf{F} = \mathbf{F}_{task} + \mathbf{F}_{obst}$. Since there is no decoupling, obstacle avoidance behavior can affect task execution behavior.

We exploit the framework described above and represented by equation 1 to solve the planning problem P_2, using a distance-based, local-minima free potential function imposed on the free space representation computed as described in Section 3.1 [3]. This potential function can be computed based on the parent/child relationship of the spheres in the tree A (see Section 3.1) and the Euclidean distance of their centers. The gradient of this navigation function $\mathcal{N}$, defining the task for the robot, can be used to derive the force $\mathbf{F}_{task} = -\nabla\mathcal{N}$ required to accomplish that task. The repulsive forces $\mathbf{F}_{obst} = -\nabla V_{obstacle}$ derived from a repulsive potential $V_{obstacle}$ associated with the obstacles can be used for collision avoidance.

Using this scheme for reactive motion generation, we want to compute a path τ using T such that $V_\tau \subseteq S(p) \supseteq T$. The assumptions we have made in Section 3.1 based on the δ-completeness of the problem guarantee that a path τ' exists such that $V_{\tau'} \subseteq T$.

But now that relevant free space connectivity is represented in the tunnel T, we do not need to restrict the search for τ to the volume of T. Using the navigation function $\mathcal{N}$ defined on T, we can guide the end-effector through T to the goal position. For the remaining links of the robot the reactive scheme of equation 1 is used, effectively extending the volume within which we search for τ to $V_{H(\tau_i)}$, where the path τ is homotopic to τ_i: $\tau \in H(\tau_i)$.

Using this reactive scheme to generate a path in conjunction with the connectivity information represented by T and obtained as a solution to P_1, it is possible to compute a solution path to the original motion planning problem in real time. The simplifying assumptions that were made to obtain a solution to P_1 are partially compensated for by the powerful reactive scheme applied to determine a solution to problem P_2. In certain situations, however, this scheme can fail. These situations are characterized by the occurrence of a structural local minima of the robot; while the potential function is free of local minima for a point robot, an articulated robot with many joints can get trapped when the various forces acting in the workspace do not result in a motion of the robot. This fact can be attributed to the conscious tradeoff of completeness for efficiency.

Note that this particular implementation of decomposition-based planning not only determines a solution path to the given planning problem, but implicitly defines a trajectory in real time. This is a significant advantage over other planning approaches, where subsequent to the path planning process a time-parameterization has to be imposed onto the resulting solution path.

4 Experimental Results

The real-time motion planning algorithm described above was implemented on a 175MHz SGI O2. It was applied to an eleven degree-of-freedom manipulator, consisting of a free-floating base with four degrees of freedom and a Mitsubishi PA-10 manipulator arm with seven degrees of freedom (Figure 3).

Depending on the complexity of the environment and the size of its local minima, the computation of the wavefront expansion (problem P_1) was performed at rates between 3 and 100 Hz. The computation of the tunnel T and the numerical navigation function can be performed in parallel with the control loop for reactive motion generation of the robot (problem P_2). Each time a new solution to P_1 becomes available, the control loop for P_2 uses the new navigation function to determine the motion of the robot.

Figure 3 shows a series of snapshots from a preliminary implementation of the algorithms described above. Figure 3 a) shows the environment, the robot in its initial position, and the initial result of the adaptive wavefront expansion algorithm, shown as a branching tree-like graph in space, with its root at the goal configuration for the end-effector. In part b) of the figure an obstacle is blocking the original path for the end-effector and a new free space representation and navigation function are computed, as can be seen in c). Figures 3 d) and e) show the result of subsequent real-time computations of the navigation function, following invalidation by an unforeseen obstacle. Note that repulsive forces originating from obstacles in the environment cause the robot to avoid collisions in a reactive manner, as can be seen in Figures 3 d) and e), where the robot passes a narrow region of free space. All degrees of freedom of the robot are used to avoid the obstacles.

5 Conclusion

To achieve real-time motion planning for robots with many degrees of freedom, a motion planning paradigm based on problem decomposition was proposed. The paradigm addresses planning problems in which a minimum clearance to obstacles can be guaranteed along the solution path. The overall planning problem is decomposed into two planning subtasks: capturing relevant connectivity information about the free space in a low-dimensional space and planning for the degrees of freedom of the robot in its high-dimensional configuration space. The solution to the lower-dimensional problem is computed in such a manner that it can be used as a guide to efficiently solve the original planning problem. This allows decomposition-based planning to achieve real-time performance for robots with many degrees of freedom.

This paper also presented a particular implementation of the decomposition-based planning framework, using an adaptive wavefront expansion algorithm to efficiently capture a volume of free space, which is in turn used to guide reactive motion control to find a trajectory for the robot, solving the original planning problem. Preliminary experimental results with an eleven degree-of-freedom robot were presented, verifying the real-time performance of the planner. The proposed framework can be extended in various directions [3], including alternative solutions to the planning subproblems, a more complete theoretical grounding, or the integration with configuration space planners.

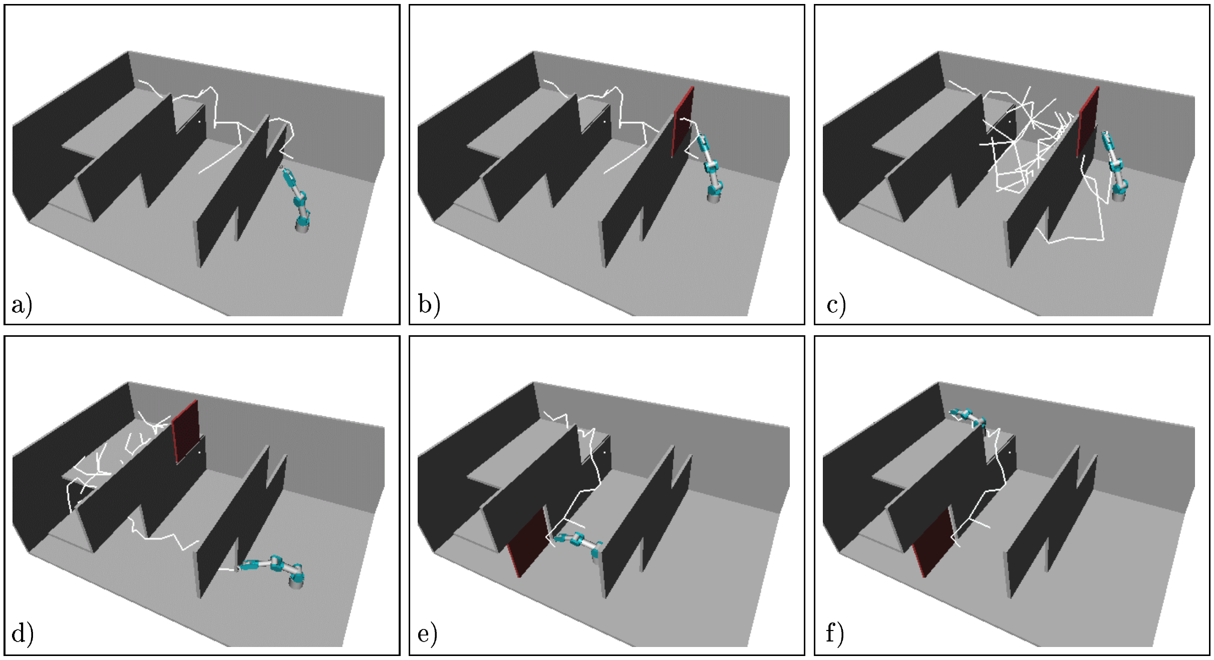

Figure 3: Real-time planning in a dynamic environment.

Acknowledgments

The authors would like to thank Attawith Sudsang and Christopher Holleman for their helpful insights and discussion in preparing this paper. Work on this paper has been supported in part by NSF IRI-970228, NSF CISE SA1728-21122N and by a Sloan Fellowship to Lydia Kavraki.

References

[1] N. Amato, B. Bayazid, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective*. AK Peters, 1998.

[2] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *Proc. Intl. Conf. on Robotics and Automation*, volume 1, pages 521–528, 2000.

[3] O. Brock and L. E. Kavraki. Decomposition-based motion planning: Towards real-time planning for robots with many degrees of freedom. Technical Report TR00-367, Rice University, August 2000.

[4] Rodney A. Brooks. Solving the find-path problem by good representation of free-space. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-13(3):190–197, 1983.

[5] John F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[6] W. Choi and J.-C. Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, volume 1, pages 24–29, 1991.

[7] Alfred Gray. *Tubes*. Addison-Wesley, 1990.

[8] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in PRM planners. In *Proc. Intl. Conf. on Robotics and Automation*, volume 2, pages 1408–1413, 2000.

[9] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. Workshop on the Alg. Found. of Robotics*, pages 141–154. A K Peters, 1998.

[10] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[11] O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *Intl. J. of Robotics Research*, 3(1):43–53, 1987.

[12] J. J. Kuffner. Goal-directed navigation for animated characters using real-time path planning and control. In *Proceedings of Captech'98*, 1998.

[13] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[14] P. Leven and S. Hutchinson. Toward real-time path planning in changing environments. In *Proc. Workshop on the Alg. Found. of Robotics*, 2000.

[15] Tsai-Yen Li and Jean-Claude Latombe. On-line manipulation planning for two robot arms in a dynamic environment. *Intl. J. of Robotics Research*, 16(2):144–167, 1997.

[16] L. Yang and S. M. LaValle. A framework for planning feedback motion strategies based on a random neighborhood graph. In *Proc. Intl. Conf. on Robotics and Automation*, volume 1, pages 544–549, 2000.