Towards Real-time Motion Planning in High-dimensional 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 and 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 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 [8] and their various extentions [1, 2, 6, 7, 11, 13].

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 significanlty from another approach to real-time path planning found in the literature [14].

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. The underlying assumption of decomposition-based motion planning is that 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 global and a local problem. The global problem is to capture the connectivity of the free space and the local problem is to find a motion for the robot, given that connectivity information. The solution to the local problem must be represented in the high-dimensional configuration space of the robot, as it must represent a valid path.

Underlying any decomposition in this planning paradigm is the realization that for most planning problems the dimensionality of the solution space is larger than the dimensionality of the problem space. The planning problem is defined by constraints in the workspace $\mathcal{W}$ of dimension $d_1 \leq 3$; the solution on the other hand is represented in the configuration space $\mathcal{C}$ of much higher dimension $d_2 > 3$. The additional dimensions arise due to the kinematic constraints of the robot. The high-dimensional solution in $\mathcal{C}$ is connected to the low-dimensional problem 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 all 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 then given by

$$S(P) = \bigcup_{1 \leq i \leq n} H(\tau_i).$$

For any given 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 centered around the origin, 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 $\tau \in H(\tau_i)$ such that $V_\tau^\delta \subseteq V_{H(\tau_i)}$. 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, $\sigma \notin S(P)$. This means that a connected component in T might actually not be connected in the free configuration space $\mathcal{F} \subset \mathcal{C}$.

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. There are many such methods and the optimal choice depends on the problem at hand. Section 3 introduces one such method, addressing the planning problem for mobile manipulators.

Once we have obtained a workspace volume T, we define the second planning problem P_2 to consist of finding a path $\tau \in S(P)$ such that $V_\tau \subseteq T$. Again, various planning methods can be employed to accomplish this task; a particular one is presented in Section 3 in the context of motion planning for mobile manipulators.

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

2.3 Discussion

Decomposition-based planning is motivated by the fact that solution paths for most planning problems encountered in service and field robotics, and even many problems in manufacturing, have a relatively large clearance to obstacles along almost the entire path. This lead to the definition of δ-hardness. The novel planning paradigm presented here attempts to solve such problem in real-time by trading completeness, as justified by the notion of δ-hardness, for efficiency.

The tradeoff of completeness for efficiency is a result of two assumptions made during the decomposition of the original planning problem. Let $\tau \in H(\tau_i)$ represents a solution path to the original planning problem. The approximation of $V_{H(\tau_i)}$ by T relies on the assumption that there exists a path $\tau \in H(\tau_i) \subseteq S(P)$ such that $V_\tau \subseteq T \subseteq V_{S(P)}$. For most practical algorithms the volume of T is going to be a proper subset of $V_{S(P)}$, i.e. $T \subset V_{S(P)}$. This approximation is addressed by the notion of δ-completeness.

Some planning approaches presented in the literature exhibit ideas that are reminiscent of decomposition-based planning. A particular instance of decomposition-based planning was applied to the problem of planning for a robot moving in the plane [5]. 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 [10, 15]. Finally, the idea of dimensionality reduction of the planning problem can be traced back to the silhouette method [4], where the planning problem is recursively projected into lower dimension. This particular approach, however, differs significantly in the way the subproblems are treated.

3 A Decomposition-based Motion Planning Method

This section presents a decomposition-based planning algorithm [3] . The goal is the development of a real-time planning algorithm for mobile manipulators with many degrees of freedom. 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, resulting 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. $V_\tau \subseteq T \subseteq V_{H(\tau)}$. In this particular instantiation of decomposition-based planning, the tunnel T will be determined by a wavefront expansion algorithm [12] described in this section.

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. 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 $\mathcal{B}_p$ 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 a termination criterion is met, indicating that a path has been found, 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 V 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.

The computational complexity of the wavefront expansion algorithm described above can empirically be determined to be roughly proportional to the complexity of the environment. This is explained by the adaptive nature of the algorithm. Large areas of free space are rapidly explored by large spheres. Narrow areas in the workspace require an increasing number of spheres.

The minimum size of the sphere contributing to T and the amount of overlap between adjacent spheres needs to be chosen appropriately for a given δ-hard problem. We assume the existence of a solution path τ such that $V_\tau^\delta \subseteq V_{S(P)}$. This means in order to capture V_τ in T we cannot underestimate $V_{S(P)}$ by more than δ. The particular choice of δ and the parameters that influence it depend on the planning problem.

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. As a result, a virtual force $\mathbf{F}$ can be computed, indicating a direction and a magnitude of force acting on the robot caused by a nearby obstacle. This force $\mathbf{F}$ can then be translated into joint torque $\mathbf{\Gamma}$ using the Jacobian J at configuration $\mathbf{q}$ of the robot: $\mathbf{\Gamma} = J^T(\mathbf{q})\mathbf{F}$. 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. This is particularly relevant in situations where the task to be accomplished requires fewer degrees of freedom than the robot has. 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:

$$\mathbf{\Gamma} = J^T(\mathbf{q})\mathbf{F} + \left[I - J^T(\mathbf{q})\overline{J}^T(\mathbf{q})\right]\mathbf{\Gamma}_0 \qquad (1)$$

[9], where J is the Jacobian of the manipulator, $\overline{J}$ designates its dynamically consistent pseudo inverse, $\mathbf{F}$ describes the forces defined by the task, and $\mathbf{\Gamma}_0$ denotes the torques to implement obstacle avoidance. Equation 1 provides a decomposition of 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 $\left(\left[I - J^T(\mathbf{q})\overline{J}^T(\mathbf{q})\right]\mathbf{\Gamma}_0\right)$. 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.

A distance-based, local-minima free potential function can be imposed on the free space representation computed as described in Section 3.1 [3]. The gradient of this navigation function can be used to derive $\mathbf{F}_{task}$, when the task consists of the end-effector of the robot reaching a certain position in the workspace: $\mathbf{F}_{task} = -\nabla\mathcal{N}$, where $\mathcal{N}$ is the adaptive numerical navigation imposed on the tunnel T, which is a subset of the tree V. The gradient of $\mathcal{N}$ defines the task to be executed by the robot. Combining the forces resulting from $\mathcal{N}$ with repulsive forces $\mathbf{F}_{obst}$ derived from proximity information to obstacles, allows the real-time computation of a solution to subproblem P_2. To compute the solution to P_2 efficiently, the solution to subproblem P_1 as represented by T after augmentation with the navigation function $\mathcal{N}$ is exploited.

In certain situations following the gradient of $\mathcal{N}$ might not lead to a solution for the planning problem P_2. This problem arises for structural local minima of the robot and is a result of the conscious tradeoff of completeness for efficiency. Methods allowing to minimize the impact of structural minima need to be developed.

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. The experimental setup can be seen in Figure 1.

Depending on the complexity of the environment and the size of its local minima, the computation of the wavefront expansion algorithm (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 1 shows a series of snapshots from a preliminary implementation of the algorithms described above. Figure 1 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

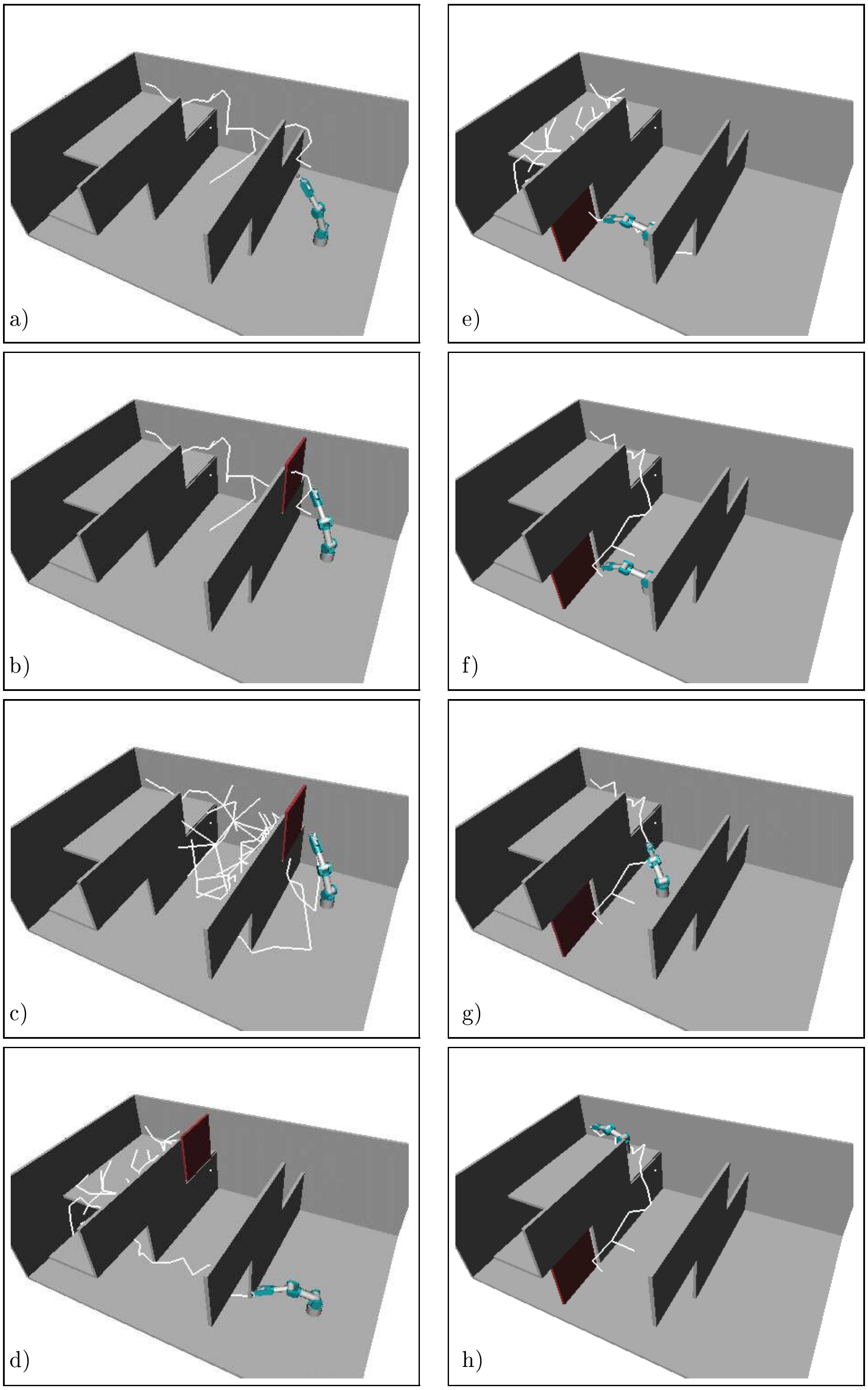

Figure 1: Real-time planning in a dynamic environment.

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 1 d) and f) 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 1 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 the connectivity of 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.

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] Nancy 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] Robert Bohlin and Lydia E. Kavraki. Path planning using lazy PRM. In *Proc. of the Intl. Conf. on Robotics and Automation*, volume 1, pages 521–528, 2000.

[3] Oliver Brock and Lydia 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] John F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[5] Wonyun Choi and Jean-Claude Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Proc. of the Intl. Conf. on Intelligent Robots and Systems*, volume 1, pages 24–29, 1991.

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

[7] David Hsu, Lydia E. Kavraki, Jean-Claude Latombe, Rajeev Motwani, and Stephen Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, pages 141–154. A K Peters, 1998.

[8] Lydia E. Kavraki, Peter Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

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

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

[11] James J. Kuffner, Jr. and Steven M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proc. of the Intl. Conf. on Robotics and Automation*, volume 2, pages 995–1001, 2000.

[12] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[13] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Iowa State University, 1998.

[14] Peter Leven and Seth Hutchinson. Toward real-time path planning in changing environments. In *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

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

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