2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Sampling-Based Motion Planning With Sensing Uncertainty

Brendan Burns Oliver Brock
Department of Computer Science
University of Massachusetts Amherst

***Abstract*— Sampling-based algorithms have dramatically improved the state of the art in robotic motion planning. However, they make restrictive assumptions that limit their applicability to manipulators operating in uncontrolled and partially unknown environments. This work describes how one of these assumptions—that the world is perfectly known—can be removed. We propose a utility-guided roadmap planner that incorporates uncertainty directly into the planning process. This enables the planner to identify configuration space paths that minimize uncertainty and, when necessary, efficiently pursue further exploration through utility-guided sensing of the workspace. Experimental results indicate that our utility-guided approach results in a robust planner even in the presence of significant error in its perception of the workspace. Furthermore, we show how the planner is able to reduce the amount of required sensing to compute a successful plan.**

I. Introduction

Most sampling-based motion planners rely on two basic assumptions: that the world is perfectly known and that it is either static or evolves in a predictable manner. These assumptions are plausible in a number of application domains, ranging from virtual prototyping and character animation to molecular biology [1]—and sampling-based motion planners have had great success in these domains. In contrast, these assumptions do not hold for physical manipulators acting in uncontrolled, unstructured, real-world environments. In these environments, the planner only has partial knowledge of its surroundings and this knowledge may be inaccurate or outdated. For sampling-based motion planners to be useful for high-dimensional, real-world motion planning, these restrictions must be removed.

In this paper we take an important first step toward enabling such real-world sampling-based motion planning. We introduce a sampling-based planner that incorporates sensor uncertainty into the planning process. This planner also suggests sensing actions which refine its incomplete and uncertain world model. The planner begins with a particular path query and incomplete, inaccurate information about the world. When necessary, it identifies the most useful additional sensing action, refines its understanding of the world based on the resulting information, and recomputes the most reliable motion. By iterating the invocation of this planner from within a "sense, plan, act" control loop, it is possible to generate robust motion for complex robots in uncertain environments.

In principle, such a control loop could use any planner. However, the explicit consideration of uncertainty and sensing actions in the planning process has several advantages. First, it allows the planner to operate despite inaccurate knowledge of the environment. Second, dynamic environments can be addressed simply by increasing the uncertainty in outdated regions of the environment. Third, the consideration of uncertainty enables the planner to determine motions that have a high probability of success in the absence of perfect sensing. Finally, the generation of on-board or off-board (smart room) sensing actions based on the particular motion objective minimizes the cost of sensing required to obtain the information required for successful planning.

Our experimental evaluation demonstrates that the proposed planner can successfully plan using partial and uncertain information about the world, shows that the explicit consideration of uncertainty improves the success rate of plans, and validates that our guided exploration is effective at reducing the sensing necessary for solving a planning problem.

II. Related Work

The work proposed here aims to apply motion planing to real-world scenarios that include uncertainty and thus potentially require feedback. The area of feedback motion planning has recently received increased attention in the literature. Conner et al. [2] propose a hybrid control system which combines a set of local control policies with a discrete planner. Lindemann and LaValle [3] propose a general theoretical approach to feedback planning using cylindrical algebraic decompositions of the configuration space. The elastic roadmap [4] consists of a series of configuration space milestones and local controllers which link the milestones together. Plans computed in the roadmap use the local controllers to maintain task constraints and obstacle avoidance while executing the plan. Van den Berg and Overmars [5] also propose integrating a roadmap and local control. Their planner, which is intended to handle dynamic obstacles, computes a roadmap for static obstacles in the environment and then uses a local obstacle avoiding controller to execute the edges that connect two milestones together. In contrast to our work, none of these approaches explicitly consider integrating uncertainty or sensory refinement and planning.

The problem of sampling-based motion planning in dynamic environments was first considered by Leven and Hutchinson [6] who propose a roadmap representation that can be efficiently modified as obstacles move. Another approach [7] assumes a finite set of dynamic obstacles, such as doors, whose motion is known. In this work the roadmap is augmented to label edges as possibly obstructed by a dynamic obstacle. When a path is computed in the roadmap, these possibly obstructed edges are re-checked to ensure they are currently free. If a path can not be found, a local planner is used to reconnect and replace the obstructed edge in the roadmap. A further refinement of this hybrid roadmap approach [8] also constructs a roadmap for the static portions of the environment. When the roadmap is queried it is augmented with a time dimension to enable consideration of dynamic obstacles and changing environmental properties. They also propose incremental replanning of the path as new information is obtained in the process of plan execution. However, they do not explicitly guide acquisition of new information.

A significant amount of recent research on uncertainty in robotics has focused on the problem of simultaneous localization and mapping (SLAM) for mobile robots [9]. The work in SLAM provides some of the inspiration for the use of modeling and probability estimation in order to minimize uncertainty. SLAM solves an important problem, namely how a maximally accurate model of the workspace can be constructed from noisy and erroneous sensor data and how this model can be used to localize the robot. A workspace model, however, is not sufficient for motion planning. For motion planning, the uncertainty in the model of the workspace must be translated into uncertainty in the model of the configuration space. In turn, this uncertainty must be integrated into the planner to identify paths that minimize the expected cost, due to failure, of the motion plan. Thus SLAM is a starting point upon which our approach builds, the maps of the world produced by SLAM algorithms (as well as any estimates of the map's certainty) are a possible source for the representation of the workspace used to compute a motion plan.

Missiuro and Roy [10] address sensor uncertainty in the context of complete motion planning. The approach adaptively samples configuration space using a distribution based upon the certainty of the sampled configurations. Paths are found in the resulting roadmap using A* search and an uncertainty heuristic. This work differs from ours because it assumes polygonal obstacles and plans motion for a 2-DOF mobile robot. In contrast, our work can be applied to arbitrary obstacle representations and articulated robots with many degrees of freedom.

There has also been work on guided exploration for a motion planning system. Yu and Gupta [11] use the notion of entropy to guide exploration to maximize the quality of the observations of the world obtained by a eye-in-hand robot. In this work the purpose of the exploration is to minimize entropy in the representation of the entire workspace, not reduce the uncertainty of a particular motion plan.

III. Utility-Guided Planning Under Uncertainty

In the real world, errors in perception introduce errors in the workspace representation a planner uses to compute feasible motions. Given the possibility of error due to uncertainty and the severity of the potential consequences of this error, it is important for a planner to reason about such potential mistakes during planning.

A practical planning method can initially only assume inaccurate and incomplete information about the workspace. To generate motion, it must use this limited information to identify the best possible path available. To achieve this, we use the formalization of Bernoullian utility [12], explored in the context of sampling-based motion planning in our previous work [13]. Based on this formulation, every successful path has a benefit, or positive *utility*, which can be a function of various task-specific considerations. Likewise, every failed path has a *cost*, or negative utility. Cost is computed from the consequences of a path failure such as a physical collision, or simply failing to achieve a goal.

Expected utility combines the notions of utility and cost, weighted by the probability that the path is successful. This probability estimate is derived from an understanding of how errors occur in the representation of the workspace (see Section IV). The expected utility of a path E is the summation over its constituent edges e_i:

$$\sum_{e_i \in E} P(e_i = \text{free}) \cdot U(e_i = \text{free}) + P(e_i = \text{obs}) \cdot \text{C}(e_i = \text{obs})$$

the function U measures utility and the function C measures cost. The path with greatest expected utility maximizes the rewards and minimizes the risks of physically enacting a particular plan.

The initial information about the workspace available to the robot may be insufficient to identify an acceptable path. When uncertainty is great or the cost of failure high, the expected utility of the best available path may be negative. In such cases, a planner's only option is to use additional sensing to explore the world and refine its representation of the workspace. Because there is a cost associated with sensing, information from the previous planning is used to guide further sensing to areas relevant to the particular path query. This sensing reduces uncertainty and provides additional information about the workspace. Using this new information, the planner can replan a new path with maximal expected utility. Planning and sensing alternate until a satisfactory path is identified and sent to the robot for execution. Although this path has positive expected utility, when executed it may turn out to be obstructed. When a path fails, information from the failure can also incorporated into the planner's knowledge of the workspace and lead to the computation of a new maximum expected utility path.

In the following we describe the three components of our new utility-guided planner: constructing a roadmap, searching for a path with maximal utility, and guiding sensing based on the planning problem and the available world model.

Construction

The roadmap constructed by our planner combines several features from previous planning algorithms. Similarly to methods for planning in dynamic environments [7], [6], [5], we allow cycles in our roadmap to add redundancy. We also delay the validation of edges until query processing [14], [15], [16].

In our planner, delaying the concrete construction of the roadmap until queried has two significant benefits. First, it eliminates the memory and computational costs of maintaining redundant paths. Edges are only created in the roadmap as required for the solution of specific queries. Also, delaying the construction of the roadmap ensures that the most up-to-date information about the workspace is used for construction. The robot is constantly acquiring new information about the workspace, either from sensing or from other planning. Delaying the evaluation of edges until they are required means that the latest knowledge is always used to evaluate their expected utility.

For the selection of nodes in the roadmap, the uncertainty roadmap is sampling-strategy agnostic. Numerous sampling strategies for selecting configurations have been proposed [17], [18], [13], [10] including one that directly incorporates uncertainty into the sampling strategy. Any one of these can be used to select the samples that define an uncertainty roadmap.

Querying

The A* algorithm [19] is used for path queries and to construct the uncertainty roadmap. A* is a general approach for searching implicitly defined graphs. For concrete implementations it requires a cost function, a heuristic function, and a way to obtain a node's successors. Note this A* cost is different than the cost of path failure.

The function for estimating the cost of an edge in A* is:

$$G(e) = P(e = \text{obs.})C(e) + \frac{P(e = \text{free})}{U(e)}.$$

$P(e = \text{obs.})$ is the probability that the edge is obstructed. This probability is estimated using knowledge of sensor error. The function $C()$ measures the cost of an edge failure. The function $U()$ measures the utility of the edge. Because the A* heuristic function finds paths with minimal cost, $1/U(e)$ is used in this expression. This favors edges with high utility, but ensures that every edge has at least some cost. In our experiments we used constant cost and utility functions for edges. This implements a path utility function that favors shorter configuration space paths. Other edge utility functions such as kinematic conditioning could easily be substituted. The A* heuristic function $H(q_i)$ uses the Euclidean distance in the configuration space. This keeps the planner "on target" toward the goal.

During the node expansion step of A*, the implicit uncertainty roadmap connected to a particular configuration is computed. Neighbors of this configuration are selected using the traditional PRM algorithm: The set of nearest neighbors to the configuration are found and returned as possible connections. The planner estimates the probability that the edge leading to each neighbor is obstructed. If this probability is greater than a threshold, the neighboring configuration is ignored. Otherwise, an edge is added to the uncertainty roadmap. Because multiple A* searches may be applied to the same roadmap, the probability associated with each possible edge is cached to avoid redundant computation. This cache is invalidated when relevant further sensing is performed. When A* search finds a path, the path and its certainty are returned to a higher level planner.

Exploration

Until a satisfactory path is found, the planner alternates path planning with guided sensing of the environment. However, there is a cost associated with sensing. To maximize planner efficiency, only required sensing should be performed. The current best path suggested by the planner provides information about the areas of the workspace that are relevant to the particular path query under consideration. Our algorithm sorts edges in the path with the highest expected utility by their uncertainty. Edges with the greatest uncertainty are explored first. Highly uncertain edges have high exploration utility for two reasons. If they are still believed to be free after sensor refinement, the corresponding reduction in uncertainty maximally increases the expected utility of the path. If they are found to be obstructed, further sensing is not wasted on refining other edges in the path that are more likely to be free.

During refinement, the planner receives new workspace information from additional sensing and performs a new A* search to identify the path with the current greatest expected utility. Planning and sensing alternate until a path whose expected utility is positive is found.

IV. Modeling Sensor Error

The planner introduced in the previous section estimates the probability that a particular location of configuration space is collision free. We assume that the world model employed by the planner does not incorporate uncertainty. Instead, the planner uses a model of sensor error to interpret the uncertainty-free world model. We now present two such models, one for binary occupancy grids obtained from range sensors and one for perceived poses of known objects determined by a vision system. The objective of this paper is not to examine the quality of such models and our planner would work with a wide variety of models. We believe that the two models used for our experiments are representative and capture important characteristics of frequently used sensing modalities.

A. *Sensor model for range sensors*

Range sensors are frequently used to build models of a robot's environment by translating distance measurement into an occupancy grid. For our experiments, we consider a simple binary three-dimensional occupancy grid. A sensing error of a range sensor will result in the incorrect labeling of cells in the occupancy grid. To evaluate a particular configuration, the corresponding workspace pose of the robot is computed. If the robot intersects obstructed cells, the configuration is considered to obstructed.

Given the possibility of error in the occupancy grid, the planner predicts the true state of a configuration space edge using a naive Bayesian model [19]. A naive Bayes model uses observations generated by some hidden state, to generate a prediction about the hidden state. In this case, the hidden state is the true state of the configuration (obstructed or free) and the labels (obstructed or free) are determined by the occupancy grid. Given a set of configurations $(q_1 \dots q_n)$ and obstructed/free labels $(x_1 \cdots x_n)$ determined by the occupancy grid, the probability of an edge being obstructed is

$$\prod_{i=1}^{n} \frac{P(q_i = x_i|\text{obs.})P(\text{obs.})}{P(q_i = x_i|\text{obs.})P(\text{obs.}) + P(q_i = x_i|\text{fr.})P(\text{fr.})},$$

where $P(q_i = x_i|\text{obs.})$ is the probability that the configuration q_i has label x_i, given that the edge is obstructed.

This model of sensor error includes several parameters. The value for $P(q_j = \text{obs.}|\text{fr.}) = 1 - P(q_j = \text{fr.}|\text{fr.})$ is set explicitly and corresponds to the fraction of observations that are erroneous, i.e., the fraction of errors made by the sensor. We estimate the remaining parameters for general categories of workspaces (e.g., office environment, assembly cell) by sampling specific example workspaces. Once we have determined these parameters, we can instantiate the model. $P(\text{obs.}) = |C_{\text{obs}}|/|C| \in [0..1]$ is the estimated ratio of the volume of obstructed configuration space and the total configuration space. $P(q_j = \text{obs.}|\text{obs.}) = (1-e)\cdot f + e\cdot(1-f)$ is determined from the error fraction e and the fraction f of edges that were observed to be obstructed in the example environments.

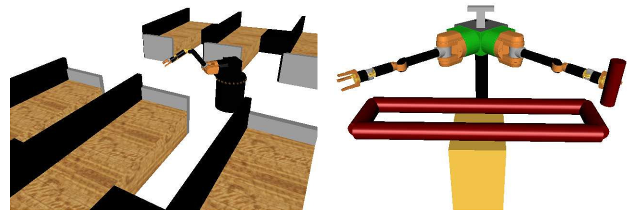
B. Sensor model for vision-based object recognition

Vision-based algorithms can recognize and localize known objects in the environments. The corresponding world model associates poses with each of the identified objects. The sensor model for this type of sensor imposes a Gaussian distribution on the pose estimate contained in the world model. The model's ability to accurately predict the state of a configuration depends on the magnitude of the localization error that would invalidate the prediction. We restrict our experiments to to translational localization error, but the same approach could be extended to rotational error. This error model assumes that error is equally likely in all three translational axis.

When a configuration is tested for collision in the world model, the collision checker determines the penetration depth (if the configuration is obstructed) or the distance to the nearest obstacle (if the configuration is free). For the calculated state of the configuration to be incorrect, the localization error must be greater than the value returned by the collision checker. If it is less, this value can be used to determine the certainty that the measured state of the configuration is correct. This is done using the cumulative distribution function (CDF) of the Gaussian:

$$1 - \frac{1}{2}(1 + \text{erf}(\frac{d}{\sigma\sqrt{2}})),$$

where "erf" is the Gauss error function and d is the penetration depth or distance to the nearest obstacle. This represents an approximation of the true probability of error, as it only considers interactions between the robot and the localization of the closest obstacle.



(a) 10-DOF Mobile Manipulator (b) 14-DOF Humanoid Torso

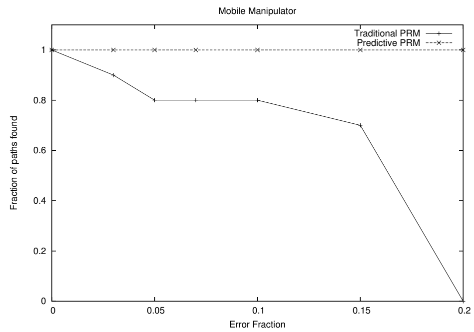
Fig. 1. The experimental workspaces and robots used to evaluate the uncertainty roadmap.

V. Experiments

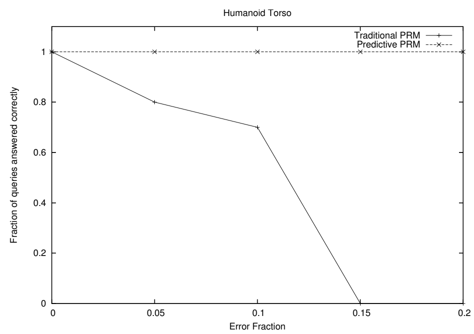
Sampling-based motion planners are generally evaluated by comparing execution times for difficult planning problems. This method of evaluation is not appropriate here. Instead, we compare how well the proposed sampling-based planner can handle uncertainty in comparison with the traditional PRM method [20]. Since we are not concerned with execution time (although we will report it), this is an adequate comparison. Using planners that employ non-uniform sampling would make an experimental evaluation more difficult, due to interactions between uncertainty and the sampling distribution.

In our experiments, we employed three different robots: a cylindrical mobile base (not shown), a mobile manipulator with ten degrees of freedom, and a humanoid upper body with fourteen degrees of freedom (both robots are shown in Figure 1). The experiments were performed in two workspaces, shown in Figure 1. The first represents a simple cubicle environment, the second emulates a construction environment. In the cubicle environment, the mobile manipulator only has 10cm clearance when inside the narrower side passage. In the construction environment, the rectangular frame splits the reachable configuration space of each arm into two regions connected by a relatively narrow passage (the arm above or below the frame).

In each of the experiments reported below, the planners were asked to solve 50 planning queries between the same set of randomly generated start and goal configurations. For each trial, the start and end configurations were sampled from a hyper-sphere located at opposite ends of the configuration space. This ensures that each of these planning problems is solvable with perfect sensing. For each sensor model, we varied the accuracy of the sensor (fraction of mislabelings



(a) 10-DOF Mobile Manipulator



(b) 14-DOF Humanoid Torso

Fig. 2. Fraction of paths successfully found by traditional PRM and the utility-guided uncertainty planning as a function of occupancy grid error.

for the occupancy grid and mean localization error for the pose-based world model). For these varying accuracies, we recorded the fraction of paths returned by the respective planners that were collision-free when validated with a certain collision checker.

Occupancy grid: The experimental results for the mobile manipulator in the cubicle world and the humanoid upper body in the construction environment are shown in Figure 2. In these experiments, the proposed planner was not allowed to perform any refinement.

These experiments demonstrate that the consideration of uncertainty by the proposed planner significantly improves the ability to determine collision-free paths, even for a large fraction of mislabelings in the occupancy grid. The proposed planner maintains 100% accuracy, even when 20% of the occupancy grid cells are mislabeled due to sensor error.

The rapid deterioration in performance for the PRM planner can be explained by the error model for the sensor. Significant sensor error will result in a large probability that long edges are erroneously classified as collision-free when in fact they are not. Consequently, the roadmap constructed by the planner will contain many invalid edges, leading to many invalid solution paths. In addition, error in the workspace representation sometimes caused the initial or goal states to be mislabeled as obstructed. Also, we halted the operation of the PRM planner after 30 minutes and considered the planning operation as failed.

Object pose experiments: Figure 3 shows the experimental results for the posed-based world model described in Section IV-B. Results for the cylindrical robot in the cubicle environment are shown on left, for the mobile manipulator in the cubicle environment in the middle, and for the humanoid torso in the construction environment on the right. In these experiments, we also varied the percentage of refinement performed by the planner. This percentages indicates the fraction of edges in a chosen solution path that are refined by additional sensing operations. Only the most uncertain edges are refined. In this particular experiment, we assumed a smart room environment that is able to direct accurate sensors to particular areas of the environment to reduce the localization error. Consequently, refinement consisted of removing the localization error for obstacles near to the edge being refined. An alternative approach to refinement could reduce the localization error for those areas perceivable by the robot; this would correspond to a scenario in which all the sensors are mounted on the robot itself.

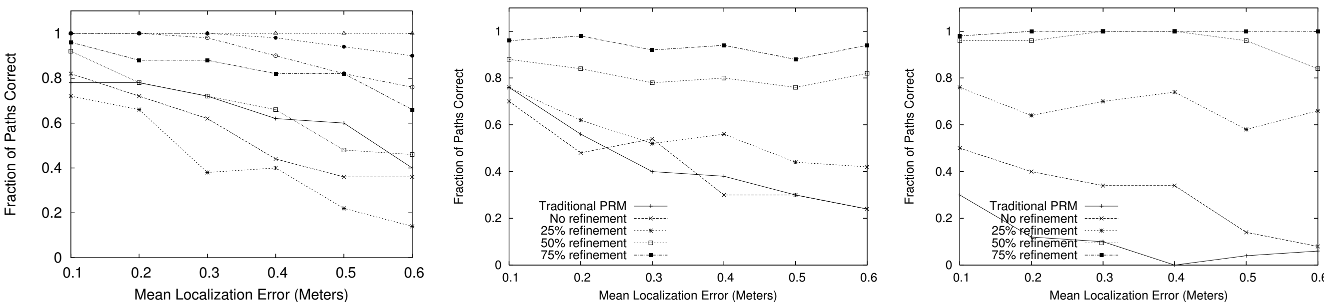
In a pose-based world model, the performance advantage of the proposed planner is less pronounced. Nevertheless, in two out of the three worlds, the proposed planner without refinement outperforms traditional PRM, selecting paths that are more likely to be free. For the mobile manipulator in the office environment, traditional PRM and the proposed planner perform roughly equivalently. This can be explained by the fact that the path that was easiest to find for the PRM planner (down the center of the hallway) also was most likely to be free.

The experiments also show that refinement significantly improves the reliability of the paths returned by the proposed planner. Interestingly, even relatively sparse refinement shows significant improvement in plan reliability. Refining only half the edges leads to over 90% reliability. This shows that guided exploration is able to direct sensor actions to relevant regions of the workspace while relying on a coarse representation in other regions. Depending on the cost of sensing, a reduction of the required sensory refinement by 50% can translate into significant performance improvements.

Running time: The consideration of uncertainty during the planner process requires computation. It therefore is to be expected that the computation time of the proposed planner exceeds that of the PRM planner.

a) Occupancy grid experiments: When there is no sensing error for the occupancy grid, the computational overhead associated with the proposed planner causes it to be outperformed by the PRM planner. As error increases, however, the runtime of traditional PRM planning increases significantly. Even when the probability of error is only 5%, the proposed planner is actually more efficient than the PRM planner. In large part this is because sensing error has a similar effect to narrow passages on PRM planning. As error increases, the probability of finding a collision free edge decreases, requiring traditional PRM planning to perform significantly more exploration to determine a valid edge.

b) Object pose experiments: For the pose-based world model, the run time of the planners did not vary significantly as error increased. The consideration of uncertainty



(a) Two Dimensional Cylindrical Robot (b) 10-DOF Mobile Manipulator (c) 14-DOF Humanoid Torso

Fig. 3. Fraction of correct paths found by each planning algorithm as a function of localization error

in planning requires on the nearly an order of magnitude more computation time than the PRM planner. This increase is mostly due to the significant amount of additional configuration exploration that is required to find paths with high certainty. An additional factor is the cost of the repeated nearest obstacle and distance queries incurred when evaluating the certainty of an edge. The increased computational cost has to be weighed against the risk of collision as well as against the tremendous cost of executing uncertain and unnecessary paths that are generated by a planner without the consideration of uncertainty.

VI. CONCLUSIONS

We presented a sampling-based planner that is capable of incorporating sensor uncertainty into the planning process. This planner represents an important step towards high-dimensional, real-world motion planning problems in which knowledge about the environment is uncertain and incomplete. The planner considers uncertainty when determining the solution paths, resulting in motion that is safe in the presence of world model inaccuracies. Throughout the planning process, the planner guides sensing in order to refine the world model. Our experimental results show that the proposed planning method significantly improves the robustness of planning in the presence of uncertainty. Further, by identifying areas of the workspace for which additional sensing provides a useful refinement of the world model, sensing resources can be employed effectively to maximally improve the resulting solution path.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF) under grants CNS-0454074, IIS-0545934, and MIT/NASA cooperative agreement NNJ05HB61A.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] A. R. D. Conner, H. Choset, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2006.

[3] S. Lindemann and S. LaValle, "Computing smooth feedback plans over cylindrical algebraic decompositions," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2006.

[4] Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation," in *Proceedings of the Robotics: Science and Systems Conference*, Philadelphia, USA, August 2006.

[5] J. P. van den Berg and M. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, October 2005.

[6] P. Leven and S. Hutchinson, "Toward real-time path planning in changing environments," in *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, 2000.

[7] L. Jaillet and T. Simeon, "A PRM-based motion planner for dynamically changing environments," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004.

[8] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proceedings of the International Conference on Robotics and Automation*, 2006.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robitics*. Cambridge, Massachusetts: MIT Press, 2005.

[10] P. Missiuro and N. Roy, "Adapting probabilistic roadmaps to handle uncertain maps," in *Proceedings of the International Conference on Robotics and Automation*, 2006.

[11] Y. Yu and K. Gupta, "An information theoretic approach to viewpoint planning for motion planning of eye-in-hand systems," in *Proceedings of the International Symposium on Industrial Robotics*, 2000.

[12] N. E. Jensen, "Introduction to bernoullian utility theory," *Swedish Journal of Economics*, pp. 163–183, 1967.

[13] B. Burns and O. Brock, "Toward optimal configuration space sampling," in *Proceedings of the Robotics: Science and Systems Conference*, Cambridge, Massachusetts, 2005.

[14] R. Bohlin and L. E. Kavraki, "Path planning using lazy PRM," in *Proceedings of the International Conference on Robotics and Automation*, vol. 1, San Francisco, USA, 2000, pp. 521–528.

[15] B. Burns and O. Brock, "Model-based motion planning," in *Proceedings of the International Conference on Robotics and Automation*, 2005.

[16] C. L. Nielsen and L. E. Kavraki, "A two level fuzzy PRM for manipulation planning," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000, pp. 1716–1722.

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

[18] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proceedings of the International Conference on Robotics and Automation*, 2003.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 2003.

[20] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.