Sampling-Based Motion Planning Using Uncertain Knowledge

Technical Report 06-30

Brendan Burns Oliver Brock
Laboratory for Pereceptual Robotics
Department of Computer Science
University of Massachusetts Amherst

June 2006

Abstract

Sampling-based algorithms have dramatically improved the state of the art in robotic motion planning. However, these approaches still make significant assumptions that limit their applicability for real-world planning. This work describes how one of these assumptions: that the world is perfectly known, can be removed. We propose a predictive roadmap planner that incorporates uncertainty directly into the planning process. This enables the planner to identify configuration space paths that minimize the risk due to uncertainty and, when necessary, directs sensing to reduce uncertainty. Experimental results in several domains indicate that the predictive roadmap is adept at planning despite uncertainty in its perception of the workspace.

1 Introduction

In the past decade, sampling-based algorithms [9, 10] have significantly advanced the state of the art in motion planning. However, these planners make several significant assumptions that limit their applicability to real-world robotic tasks. In particular, these approaches assume that the world is static, that the path, once computed, is executed perfectly by the physical robot and that the workspace is perfectly known. Using sampling-based motion planning for real-world robots, such as the autonomous mobile manipulator shown in Figure 1, requires relaxing these assumptions. Real-world environments often contain dynamically moving objects, motions of physical robots are subject to execution error and knowledge of the workspace must be obtained from noisy sensors. In this work we focus on this final assumption. We propose a novel, predictive roadmap planning algorithm that integrates uncertainty directly into the planning process. This integration allows the planner to compute paths that minimize risk despite potential errors in the representation of the workspace.

The integration of uncertainty into the planner also enables it to actively direct sensor refinement. Most robots possess many different sensors for perceiving the world. Each of these sensors has different accuracy characteristics and computational costs associated with it. Even within a single sensor modality, different granularities of perception are possible with associated differences in computational cost. A motion planner that is aware of the uncertainty in its perception should also be capable of adjusting the granularity of its perception of different regions of workspace to adapt sensing to the planning task. Large open regions of the configuration space may permit a significantly coarser granularity of sensing than constricted regions. The integration of uncertainty and sensing into the planner enables a robot to minimize the sensing required to successfully motion plan in a given environment.

Figure 1: A platform for autonomous mobile-manipulation under construction at UMass

This paper proposes an approach to integrating uncertainty and sensing into sampling-based motion planning based upon the idea of a *predictive roadmap*. In contrast to a traditional roadmap, the predictive roadmap does not guarantee that the roadmap graph is unobstructed. Instead, each node and edge in the graph is labeled with the probability that it is obstructed or free. The predictive roadmap allows cycles in the roadmap graph. Maintaining redundant paths provides alternatives in the advent that a particular path is obstructed. Allowing cycles also allows the planner to add new more certain paths between configurations despite the presence of another path in the roadmap. For any particular path query, the predictive roadmap is queried using A* [17] to find the path that is most certain. If this search fails to find a satisfactorily certain path, information obtained in the search can be used to identify areas of the workspace where further sensing is required.

2 Related Work

Noisy sensing and uncertainty in representation has received significant attention in the field of simultaneous localization and mapping (SLAM) for mobile robots. Thrun et al. [18] provide an excellent summary of this work. There has been significantly less work in the field of motion planning for robots with many degrees of freedom.

Classically, the problem of uncertainty in motion planning was addressed with preimage backchaining [12]. This approach deals with planning given uncertainty in the effects of a motion command.

The first use of a predictive roadmap is Burns and Brock [4]. This work does not address perceptual uncertainty, but rather used statistical approximate models to predict the state of roadmap edges to improve the efficiency of the motion planner. FuzzyPRM [14] also explores assigning probabilities to edges in the roadmap.

The only other work to directly address perceptual uncertainty in sampling-based motion planning is Missiuro and Roy [13]. This work describes a method for adapting the configuration space sampling distribution based upon the certainty of the sampled configurations. Like predictive roadmaps, paths are found in the resulting roadmap using A* search and a uncertainty heuristic. This work only addresses simple vertex based model of uncertainty and planning for simple 2-DOF mobile robot.

Another source of uncertainty in motion planning is the movement of dynamic obstacles. Leven [11] explores how to improve the efficiency of PRM planning to allow it to operate quickly enough to facilitate re-planning in the presence of dynamic obstacles. Jaillet [8] introduces cycles in the roadmap graph and labeling of edges with possible obstructing obstacles in order to allow the planner to reason about which paths are feasible given certain moving obstacles. van den Berg [19] also addresses roadmap based planning in dynamic environments.

Another method for dealing with perceptual uncertainty is augmenting the execution of a motion plan with online control that is capable of obstacle avoidance. The elastic strip algorithm [3] dynamically modifies a computed motion plan to avoid dynamic objects. This approach can only generate paths that are homeotopic with the original path. Alami et al. [16] use anticipation of potential obstacle movement and local control to adjust the velocity and trajectory of a mobile robot following a specified path.

Yu and Gupta [20] have proposed work that integrates sensing into motion planning. They use an information theoretic approach that guides a eye-in-hand robot to perform the set of motions that reduce the entropy in the robot's perception of its environment. This exploration is focused on understanding the environment rather than reducing the uncertainty of the motion planning.

Other work is not directly related to uncertainty but has explore concepts that are central to the predictive roadmap algorithm. LazyPRM [2] delays the evaluation of edges in the roadmap until a path query is received and uses A* to search for roadmap paths. Others [15, 7] have also explored the use of cycles in roadmaps for redundant motion planning.

3 Incorporating Uncertainty with Predictive Roadmaps

When sensor data from the real world is used to perceive the workspace, motion planning algorithms must be aware of the inherent uncertainty in this perception. The process of sensing introduces error into the model of the workspace. When the motion planning algorithm uses this model to determine if a configuration is uncertain or free, this error introduces uncertainty into the resulting motion plans and may lead to invalid paths.

Traditional PRM planning assumes that its perception of the workspace is perfect. When this is not true, two important things must be considered: First, it is necessary to maintain redundant paths connecting configurations. This allows for recovery in the advent that a path is found to be obstructed. It also allows the motion planner to add new paths that are more likely to be free than existing uncertain paths connecting pairs of configurations. Second, complete examination of the trajectories that connect configurations is no longer warranted. Edge checking has been found [4] to be the primary computational expensive in PRM planning. This computational expense is not worthwhile in uncertain environments. Given uncertainty, exhaustive examination of a trajectory does not provide significantly better guarantees about the state of an edge compared to less expensive approximations of the edge's state. These distinctions motivate the development of the predictive roadmap for motion planning which allows cycles and labels each edge with the probability that it obstructed or free. Methods for assigning this probability are discussed in Section 4. For the time being, we will simply assume that each edge has been labeled appropriately.

Construction

In traditional PRM planning, as configuration space is sampled, edges are directly added into the roadmap graph by modifying the representation of the graph stored in memory. The predictive roadmap is not built in this way. Instead it is constructed lazily. Pieces of the predictive roadmap are only actually constructed in response to specific motion planning queries.

Delaying the concrete construction of the roadmap until queried has three significant benefits. First, it allows the predictive roadmap to maintain redundant paths with no computational cost. Because edges are not inserted into the roadmap until required, there is no cost associated with allowing redundant edges. Only when those redundant edges are required by actual path planning are they inserted into the graph. Second, delayed evaluation of the roadmap ensures that only those parts of the roadmap that are required to satisfy a query are actually computed. If a portion of the configuration space is never required to solve a query, the roadmap associated with that region is never constructed. This adapts computation in a task-specific manner. Finally, delaying the construction of the roadmap ensures that the most up to date information about the workspace is used for construction. While waiting for path queries, the robot may be refining its perception of its workspace. Delaying the evaluation of edges until they are required means that the latest knowledge is always used.

The predictive roadmap is sampling-strategy agnostic. Numerous sampling strategies for selecting configurations have been proposed [1, 6, 5, 13] including one [13] that directly incorporates uncertainty into the sampling strategy. Any one of these can be used to select the samples that define a predictive roadmap. For simplicity we used uniform sampling in our experimental evaluation (Section 5).

Querying

We use the A* [17] algorithm to perform path queries and construct the predictive roadmap. The A* algorithm is a general approach for searching implicitly defined graphs. For a concrete implementation it requires a cost function, a heuristic function and a method of obtaining the children of a node.

The function for estimating the cost of an edge in the predictive roadmap is:

$$G(q_i, q_{i+1}) = P(e(q_i, q_{i+1}) = \text{obstructed})K_c + K_s$$

$P(e(q_i, q_{i+1}) = \text{obstructed})$ is the probability that the edge is obstructed. Methods for estimating this probability are described in Section 4. K_c is a constant that estimates the cost of moving along an edge that turns out to be obstructed. K_s is a constant that ensures that no edge has a cost of zero even if its probability of obstruction is zero. This plays an important role in balancing A*'s exploration of paths likely to be free with directed search towards the goal.

For the heuristic function $H(q_i)$ we use the Euclidean distance in the configuration space. All else being equal, this biases the A* search toward shorter paths through the configuration space.

When A* chooses to expand a node q, the set of nearest neighboring configurations Q are found and returned as possible children of q. For each configuration $q' \in Q$, the planner evaluates the probability that the edge from q to q' is obstructed. If this probability is greater than a threshold, the q' is discarded. Otherwise, an edge $e(q, q')$ is added to the predictive roadmap. Both the edges and their probabilities are cached across multiple instances of A* search in to prevent redundant computation. This method of node expansion is modeled on the traditional PRM connection step.

When A* search finds a path, its certainty is evaluated. If the minimal certainty that the path is free exceeds a threshold, it is returned as the result of the query. If this threshold test fails, the planner indicates that further refinement of the workspace representation is required to compute a path with the requisite certainty.

Refinement

For a given representation of the workspace, a planner may be unable to find a path whose certainty satisfies the requirements of the robot's task. However, in such circumstances, the process of searching indicates workspace regions that require further sensing to find a satisfactory path. This refinement proceeds as follows: Each edge in the path whose uncertainty is below some task-dependent certainty threshold indicates a region of the workspace that requires further exploration. Given these workspace regions, sensing is initiated by the planning algorithm to reduce their uncertainty. Once sensing is completed, the certainty of the path is re-evaluated. If the certainty of the path is now great enough, the path is returned as the result. Otherwise, the uncertain edges are removed from predictive roadmap and to enable the search for an alternate path. This post-process step ensures that sensing fidelity is adapted as required by motion planning queries. This minimizes the cost of sensing required for successful motion planning.

4 Modeling Uncertainty

The predictive roadmap requires an estimate of the probability that each edge is obstructed. This estimate is dependent upon the underlying representation of the workspace. Each representation of the workspace introduces different types of error. Analysis of the source of this error provides a method for reasoning about the resulting uncertainty. In this work, we consider two different representations: a workspace occupancy grid and a representation of localized obstacles with known geometry.

4.1 Occupancy Grid Representations

An occupancy grid subdivides the workspace into series of cells. Although some occupancy grids allow mixed or probabilistic labels for cells, we assume that the grid is binary. Cells are marked as either obstructed or free. To evaluate the state of a configuration, the workspace location of the robot given the configuration is computed. If any of the cells that overlap this location are obstructed, the configuration is considered obstructed. If all of the cells are free, the configuration is free. Error in this representation causes a cell that is obstructed to be marked as free or vice-versa. In either case, error introduces uncertainty in the state of roadmap edges. To evaluate this uncertainty we examine the problem as an example of observed data emitted by some underlying unknowable hidden process. In this context, the examination of an edge results in a series of *observations* of whether a particular configuration is obstructed or free. The planner can not know the true state of each configuration. It must instead obtain information through the fallible workspace model. This means the planner receives noisy observations of the true state of each configuration. Given a sequence of noisy observations, the task is to predict the underlying hidden state. There are many well known methods for solving this problem. We examine two: a naive Bayes model and a hidden Markov model (HMM) [17]. The following describes the details of these models. Section 5 examines their accuracy.

Naive Bayes

A naive Bayes model derives its name from its assumption that all observations are independent of each other. For edge checking, this assumption is false, hence the model's naivete. However, the falsehood of this assumption does not significantly impact the model's predictive performance. By assuming each observation is independent, the probability of a hidden state given a series of observations is simply the product of each individual probability. For our work, the probability of an obstructed edge is given by:

$$\prod_{j=1}^{n} \frac{P(q_j = x|e = \text{obs.})P(e = \text{obs.})}{\sum_{y=\text{obs.,free}} P(q_j = x|e = y)P(e = y)}$$

Applying this model, requires parameter estimates: $P(\text{obs.})$, the probability that an edge is obstructed, $P(q_j = \text{obs.}|\text{obs.})$ the probability of an obstructed observation given an obstructed edge and $P(q_j = \text{obs.}|\text{free})$ the probability of an obstructed observation given a free edge. We have found that estimates of these parameters are generally applicable across similarly structured environments, such as all office spaces.

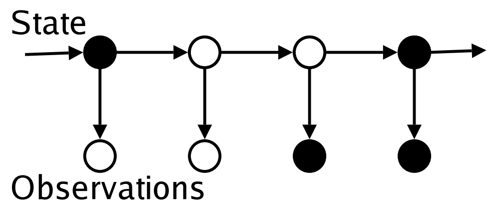


Figure 2: The hidden Markov model used for predicting the state of an edge given a set of observations.

Hidden Markov Models

Hidden Markov models (HMMs) are a graphical model [17]. Unlike naive Bayes models, HMMs do not assume that each observation is independent. Thus they are a more accurate representation of sequential observations like the examination of an edge between configurations. We can model the noisy examination of a sequence of trajectories as the simple HMM shown in Figure 2. There is a vertical slice in the model for each sequential observation of a configuration along the linear edge. In each slice the hidden node represents the true, unknowable, state of the configuration. The observed node contains the noisy observation generated by the planner's representation of the workspace.

Estimating the obstructed probability of a particular sequence of observations begins by labeling the observation nodes with the observed values. The Viterbi algorithm [17] is used to calculate the maximum likelihood sequence of hidden states given the observations. This estimate of the hidden state is used to predict if the edge is obstructed or free. The likelihood of the sequence of hidden states is used to estimate the certainty of this prediction.

4.2 Localized Obstacle Representations

An alternative method for representing the workspace is to assume that the geometry of all obstacles in the workspace is known a priori and that these obstacles can be identified and localized using sensor information. We call this a *localized obstacle* representation. Given this representation, error takes the form of mispredicting the location of an obstacle rather than mislabeling the state of a cell. By assuming that the magnitude of this localization error follows a probability distribution, we can calculate the certainty of each observation of the state of a configuration in the workspace representation.

Gaussian Error Models

In the following, we assume that the magnitude of the localization error follows a Gaussian distribution. We also assume that localization only introduces translational error in the position of an obstacle. We do not consider rotational error. Extending this approach to consider rotational error is part of our future plans.

Whether a configuration is obstructed or free the certainty of that observation is related to the magnitude of localization error that would invalidate the observation.

For obstructed configurations, we estimate this magnitude by calculating the penetration distance in the positive and negative directions along all three translational axes. This provides two translational magnitudes in each direction that would invalidate the observation. A simple example of this along one axis is illustrated in Figure 3. In this example, the localized obstacle (the gray box) in the workspace model indicates that configuration q is obstructed. If the magnitude of error in localization of the obstacle is between d_1 and d_2 then the true position of the obstacle overlaps q and the observation is correct. By assuming the magnitude of localization error is drawn from a Gaussian distribution, then the probability that the magnitude of error that

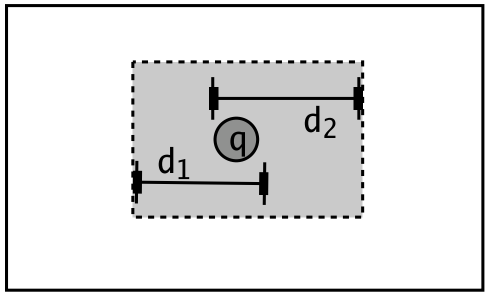


Figure 3: The possible results of localization error. If the localization error for the gray obstacle is between $d_1 and d_2$, then the the configuration q's state is correctly known.

lies within a range that doesn't invalidate the observation is given by the cumulative distribution function (CDF) of the Gaussian function:

$$\frac{1}{2}(\text{erf}(\frac{d_1}{\sigma\sqrt{2}})) - \frac{1}{2}(\text{erf}(\frac{d_2}{\sigma\sqrt{2}}))$$

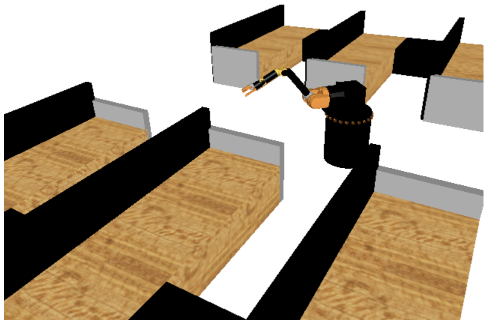
where erf is the Gauss error function, d_1 and d_2 are magnitudes in along the positive and negative axis respectively. We calculate this probability for each axis. The lowest probability is used as the certainty estimate. This is only an approximation of the true probability but seems to work reasonably well in practice. In general, penetration distances are challenging to calculate. Better approximations are possible but incur significantly greater computational costs.

For free configurations we measure the distance from the robot to the closest obstacle (d_o) and use this value as the magnitude of error required to invalidate the free observation. The probability of this is also computed using the CDF.

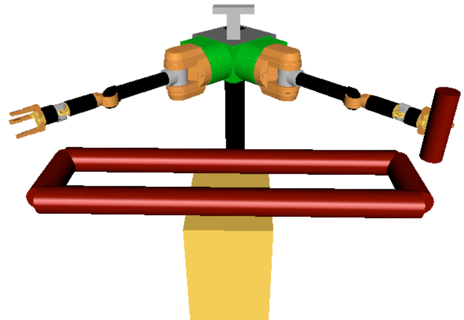
5 Empirical Evaluation

We ran numerous experiments to demonstrate the suitability of the approach described in previous sections. This evaluation addressed two important questions. First, can the models discussed in Section 4 provide reasonable values for use in constructing the predictive roadmap? Second, does the predictive roadmap significantly improve the ability of a motion planner to compute collision-free paths in the presence of uncertainty?

To answer these questions, we ran the planner in three simulated worlds designed to resemble real-world environments. Two of these are pictured in Figure 4: a 10-DOF mobile manipulator platform in an office environment and a 14-DOF humanoid torso in a construction environment. A third world with a 2-DOF cylindrical robot in the same workspace as the mobile manipulator was also used. We ran experiments in each of these worlds simulating both methods of representing the workspace. To simulate error in the occupancy grid representation we used an adjustable probability p_{err} that any particular cell in the grid was mis-labeled. To simulate error in the localized obstacle representation, we added translational error to each obstacle's true position. The magnitude of the error was sampled from a Gaussian distribution and was different for each obstacle.



(a) 10-DOF Mobile Manipulator



(b) 14-DOF Humanoid Torso

Figure 4: The experimental workspaces and robots used to evaluate the predictive roadmap.

5.1 Modeling

We first examine the models used to assign values to the predictive roadmap.

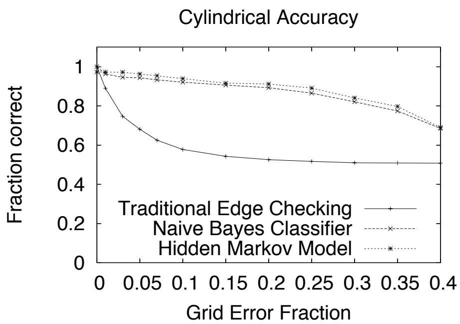
Occupancy Grid

To evaluate the naive Bayes and hidden Markov models we generated 500 free and 500 obstructed edges in configuration space. The edges were chosen uniformly at random from the set of edges that were shorter than the radius used for the nearest neighbor query in motion planning. In this way we believe they are representative of the edges encountered in motion planning. For each edge, each model was asked to predict if the edge was obstructed or free. The traditional edge checking model simply returns obstructed if any configuration along the edge is observed to be obstructed. The fraction of the edges that each model correctly predicted as a function of the probability of error is shown in Figure 5.

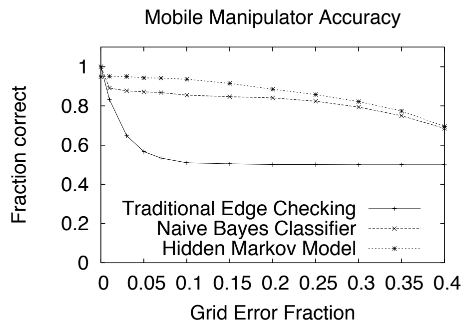
In all three worlds, as error increases traditional edge checking quickly devolves into predicting that all edges are obstructed. This occurs because the probability of moving along an edge and not receiving an erroneous observation rapidly becomes quite small. This means that even completely free edges are likely to result in an obstructed observation. In contrast, both the naive Bayes and hidden Markov models are significantly more robust to error. Interestingly, HMMs only slightly outperform naive Bayes models in the environments. From this we conclude that the added accuracy does not justify the increased computational cost of the HMM and in our planning experiments we chose to use the naive Bayes model for predictions.

Localization

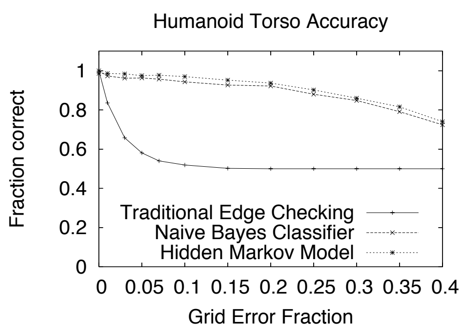
Unlike the occupancy grid model, the model of uncertainty in localization can not be used to provide predictions about the state of an edge, instead it provides the reliability of the workspace model's predictions. This uncertainty is used to bias the posture of the robot toward configurations that are more certain to be free. An image of this uncertainty function for the 2-DOF cylindrical robot in the office cubicle environment is shown in Figure 6. In the image, black indicates obstructed with absolute certainty and white indicates free with absolute certainty. Looking at the image it is easy to see that a planner that selects motions biased by this uncertainty function will choose configuration-space paths that are more likely to be unobstructed.



(a) 2-DOF Cylindrical Robot



(b) 10-DOF Mobile Manipulator



(c) 14-DOF Humanoid Torso

Figure 5: Edge predictions accuracy for HMM, Naive Bayes and traditional edge checking as a function of error for three different robots with varying degrees of freedom.

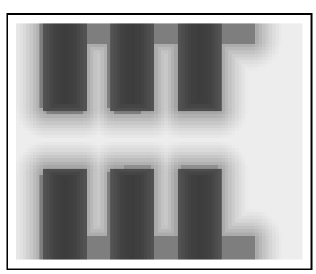
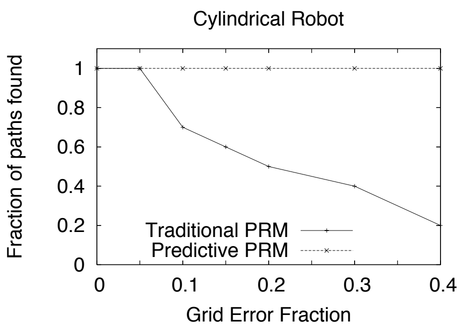
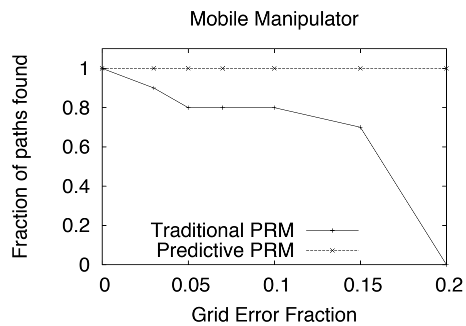


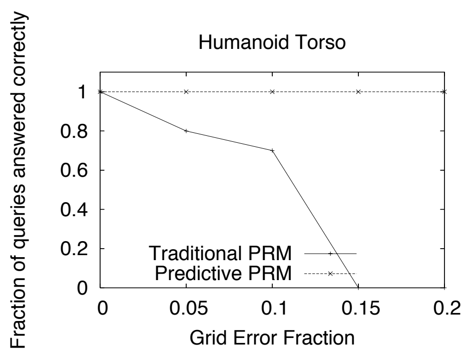
Figure 6: A graphical representations of the uncertainty function for the 2-DOF cylindrical robot using a localized obstacle model of the workspace. Black indicates absolute certainty of obstruction, white indicates absolute certainty of free space.



(a) 2-DOF Cylindrical Robot



(b) 10-DOF Mobile Manipulator



(c) 14-DOF Humanoid Torso

Figure 7: Fraction of paths successfully found by tradition PRM and the predictive roadmap algorithm as a function of occupancy grid error.

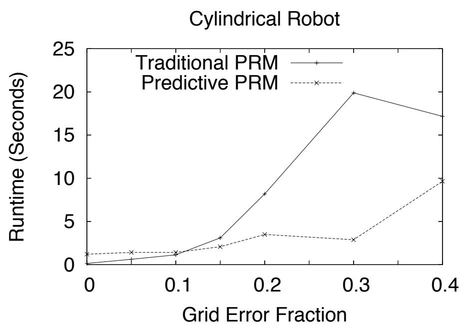
5.2 Planning

In addition to assessing the uncertainty models, we performed experiments to evaluate the use of these models in predictive roadmap motion planning. To evaluate the planner we generated a series of random path queries for each environment. We then asked both a traditional implementation of PRM planning and the predictive roadmap planner to find a path that satisfied the query. We varied the amount of error present in the workspace representation and observed both the percentage of queries that the planner could solve correctly and the average runtime that the planner took in computing these solutions. All of the path queries had solutions given accurate representations of the configuration space. All experiments were implemented in C++ and were run on a 3Ghz Pentium 4 with 1 GB of RAM running the Linux operating system.

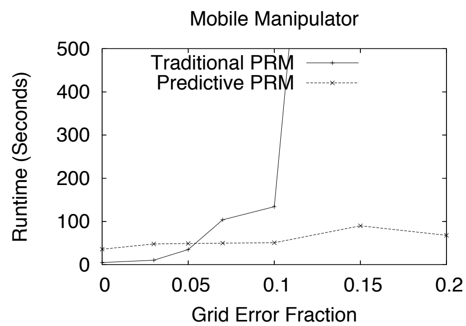
Occupancy Grid

The experiments using an occupancy grid representation introduced error by increasing the probability that a cell in the grid was mis-labeled. The naive Bayes model was used by the predictive planner to estimate the state of edges in the predictive roadmap. The fraction of path found and the average runtime for each planner in each environment as a function of error is shown in Figures 7 and 8.

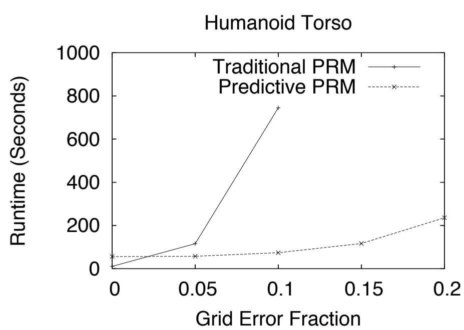
From these graphs it is easy to see that the because the predictive roadmap algorithm is aware of uncer-



(a) 2-DOF Cylindrical Robot



(b) 10-DOF Mobile Manipulator



(c) 14-DOF Humanoid Torso

Figure 8: Runtime required to successfully find a path as a function of occupancy grid error.

tainty and the possibility of error it is significantly capable of always finding a path as error increases. There were two causes of failure for the traditional PRM algorithm. First, error in the workspace representation caused the initial or goal states to be mis-classified as obstructed. Second, we halted the operation of the traditional PRM algorithm at 30 minutes, path queries that were halted, were considered to have failed. With sufficient error, we expect that the runtime performance of the predictive approach to degrade to the point of failure as well.

Interestingly, when there is no error, the excess computation associated with the predictive roadmap makes its average runtime slower than traditional PRM planning. As error increases, the runtime of traditional PRM planning increases significantly faster than predictive roadmap planning and even when the probability of error is only 5%, predictive roadmap planning is more efficient than traditional PRM planning. In large part this is because error has a similar effect to narrow passages on PRM planning. As error increases the probability of finding a collision free edge significantly decreases, requiring traditional PRM planning to perform significantly more exploration.

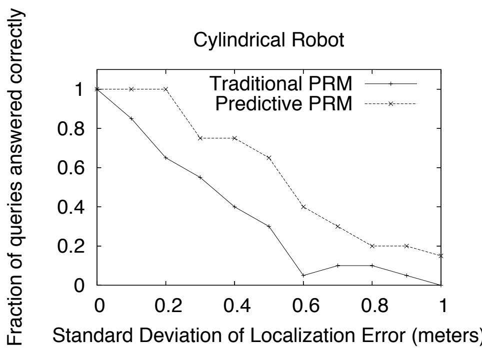
Localization

Predictive roadmap planning was also used with a localized obstacle model of the workspace. As before, error was introduced adding Gaussian noise to the true location of each obstacle prior to providing the workspace representation to the planner. Each planner was then asked to find a path between a random pair of configurations drawn from opposite sides of the workspace. Once the planner had solved they query, the correctness of the path was tested against the true state of the workspace. Whether or not the path computed by the planner was actually collision free was recorded. Experiments were run in each of the three workspaces for both traditional PRM and predictive roadmap planning. The fraction of paths returned by each planner that were in fact collision free as a function of error in localization is shown in Figure 9. The numbers reported are the average over twenty different path queries. In general, these graphs show that the predictive roadmap planner is more robust to localization error. It maintains near perfect accuracy for up to twenty centimeters of error. This is a significant amount of error in worlds where the average size of objects is on the order of a half a meter. Because the humanoid torso is fixed to the ground, it is significantly more susceptible to error in localization. While the predictive roadmap outperforms traditional PRM, the performance of both decays rapidly. In general, once performance begins to degrade, both planners show linear decreases in reliability as error increases.

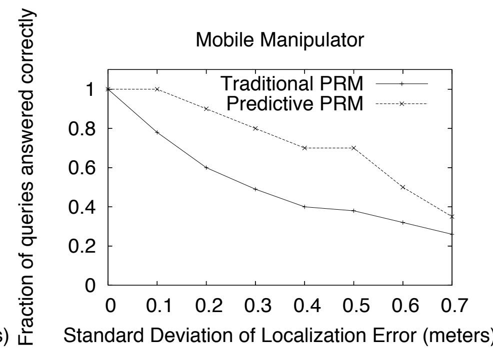
Unlike planning with the occupancy grid representation, the runtime of the planner did not vary significantly as error increased. For the two more complex workspaces the predictive roadmap planner is nearly an order of magnitude slower than traditional PRM planning. This is largely due to the repeated nearest obstacle and distance queries incurred in evaluating the certainty of each edge. We are investigating methods of speeding this computation.

5.3 Refinement

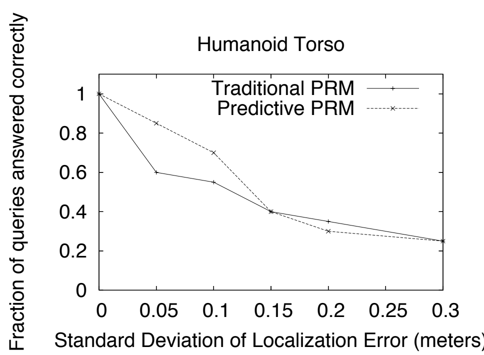
The final aspect of the predictive roadmap planning is the incorporation of sensory refinement into the planner. To demonstrate this, we ran experiments in each world using a certainty threshold. Whenever the certainty of an edge in this path was below this threshold, the planner made a request to refine the sensing associated with that edge. Perfect information about the edge was then obtained. For the same planning experiments described previously, the fraction of edges that required refinement was recorded. This is given as a function of error for the occupancy grid and localization workspace representations in Figure 10. In these graphs it is clear that refinement required is a function of error. However, in many cases, the refinement is extremely minimal. An exception to this is the torso world in using localization. In this world, the proximity of the torso's base (which can not move) to the obstacle made every prediction uncertain. This



(a) 2-DOF Cylindrical Robot

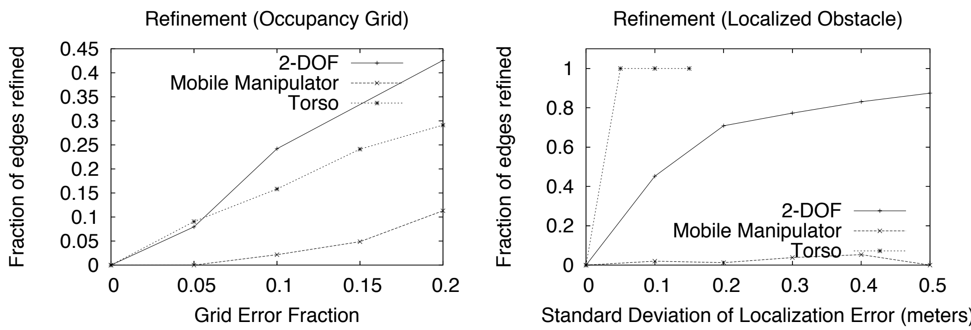


(b) 10-DOF Mobile Manipulator



(c) 14-DOF Humanoid Torso

Figure 9: Fraction of correct paths found by each planning algorithm as a function of localization error



(a) Occupancy Grid Representation

(b) Localized Obstacle Representation

Figure 10: Sensor refinement required to reduce uncertainty as a function of error

points to the need to refine the certainty metric to only consider certain parts of the robot, such as the arms in the case of the torso. Generally, the refinement experiments show that adapting the sensing granularity at the direction of the predictive roadmap planner can significantly reduce the sensing required for motion planning.

6 Conclusions

Sampling-based algorithms have significantly advanced the state of the art in motion planning. However, these approaches still make significant assumptions that prevent the application of these methods to real-world robotics. In the preceding we have proposed an approach for eliminating one of these assumptions, in particular, the assumption that the state of the workspace is perfectly known. When real-world sensors are used to perceive the workspace, sensor error introduces uncertainty into this perception. The predictive roadmap method for motion planning directly incorporates the uncertainty into the roadmap by labeling each edge with its probable, rather than absolute, state. This allows the planner to reason about uncertainty and select paths that minimize the risk due to sensor error. Incorporating uncertainty into the planning algorithm also allows the planner to suggest additional areas of the workspace where further sensing is needed to enable the computation of successful plans. We have described several models for estimating the state of an edge for two different methods of representing the workspace. Empirical experiments demonstrate that these models successfully capture the state of the world, even in the presence of error. Finally, planning experiments show that the predictive roadmap approach is significantly more robust. It can correctly solve path queries even in the presence of significant error.

Acknowledgment

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] 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.

[2] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 521–528, San Francisco, USA, 2000.

[3] O. Brock and O. Khatib. Elastic strips: A framework for integrated planning and execution. In P. Corke and J. Trevelyan, editors, *Proceedings of the International Symposium on Experimental Robotics*, volume 250 of *Lecture Notes in Control and Information Sciences*, pages 328–338. Springer Verlag, 1999.

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

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

[6] 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.

[7] Y. Huang and K. Gupta. A delaunay triangluation based node connection strategy for probabilistic roadmap planners. In *Proceedings of the International Conference on Robotics and Automation*, New Orleans, USA, April 2004.

[8] 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.

[9] 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*, 12(4):566–580, 1996.

[10] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, pages 293–308, 2000.

[11] 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.

[12] T. Lozano-Pérez, M. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1), 1984.

[13] P. Missiuro and N. Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *Proceedings of the International Conference on Robotics and Automation*, 2006. To Appear.

[14] 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*, pages 1716–1722, Takamatsu, Japan, 2000.

[15] D. Nieuwenhuisen and M. H. Overmars. Useful cycles in probabilistic roadmap graphs. In *Proceedings of the International Conference on Robotics and Automation*, pages 446–452, New Orleans, USA, April 2004.

[16] K. M. K. R. Alami, T. Simeon. On the influence of sensor capacities and environment dynamics onto collision-free motion plans. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2002.

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

[18] S. Thrun, W. Burgand, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[19] J. P. van den Berg and M. Overmars. Roadmap-based motion planning in dynamic environments. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 1598–1605, 2004.

[20] 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.