Construction and Use of Roadmaps That Incorporate Workspace Modeling Errors

Nick Malone, Kasra Manavi, John Wood, Lydia Tapia

***Abstract*— Probabilistic Roadmap Methods (PRMs) have been shown to work well at solving high Degree of Freedom (DoF) motion planning problems. They work by constructing a roadmap that approximates the topology of collision-free configuration space. However, this requires an accurate model of the robot's workspace in order to test if a sampled configuration is in collision or not. In this paper, we present a method for roadmap construction that can be used in workspaces with uncertainty in the model. For example, these can be inaccuracies that are caused by sensor error when an environment model was constructed. The uncertainty is encoded into the roadmap directly through the incorporation of non-binary collision detection values, e.g., a probability of collision. We refer to this new roadmap as a Safety-PRM because it allows tunability between the expected safety of the robot and the distance along a path. We compare the computational cost of Safety-PRM against two planning methods for environments without modeling errors, basic PRM and Medial Axis PRM (MAPRM), known for low computational cost and maximizing clearance, respectively. We demonstrate that in most cases, Safety-PRM produces high quality paths maximized for clearance and safety with the least amount of computational cost. We show that these paths are tunable for both robot safety and clearance. Finally, we demonstrate the applicability of Safety-PRM on an experimental system, a Barrett Whole Arm Manipulator (WAM). On the WAM, we demonstrate the mapping of expected collision to robot speeds to enable the robot to physically test the safety of the roadmap and use torque estimation to make roadmap modifications.**

I. Introduction

The motion planning problem consists of finding a valid (collision-free) path from a start state to a goal state. One solution to this problem is to define a roadmap that captures the topology of the collision-free portion of configuration space. However, the complexity of the workspace and robot can make this process challenging. Probabilistic Roadmap Methods (PRMs) have addressed this challenge by constructing a roadmap of randomly sampled robot configurations and testing each configuration for collision [19]. Connections are made between two samples when a collision-free transition can be made. These samples (vertices) and connections (edges) define a roadmap that the robot can safely traverse. Recently, PRMs have been extended to be adaptable [25] [4]. These new methods can deform paths [17], update roadmaps due to moving obstacles [14] [26], map both collision and collision-free states [9], and deal with uncertainty in the motion model [6] [2] [1] [24]. However, despite all these advances, PRMs require that the model of the problem must be accurate, e.g., there must be a clear delineation between collision and collision-free states. Error-prone collision detection can lead to erroneous roadmaps which produce feasible paths in the modeled environment but lead to collisions in the actual world.

Fig. 1: Whole Arm Manipulator (WAM) feeling an obstacle boundary.

Distinguishing between collision and collision-free configurations requires a model of the planning space. These models are often manually constructed, have well defined obstacle boundaries, and can be easily tested for robot-obstacle collision. Advancing technologies are producing 3D environment models at lower costs than ever before [29]. These models are constructed using technology such as sensors [29] and cameras [13]. However, all these technologies are prone to modeling error. Therefore, unlike the manually modeled environment, obstacle boundaries can be fuzzy or approximated thus making collision tests error-prone.

In this paper, we introduce Safety-PRM that accounts for modeling uncertainty in the roadmap. This new method calculates and incorporates a probability of collision during roadmap construction. These probabilities reflect the amount of certainty in the collision-state of a node or edge in the roadmap, thus allowing the robot to have an expectation of safety from an obstacle's surface. The certainty can also be used to weigh roadmap edges, thus allowing robots to easily transition between being safer (higher expected clearance) or take shorter paths (lower regard to expected clearance).

We demonstrate the applicability of these new methods on a series of environments with both rigid body and articulated linkage robots. We show that in most rigid body cases and in all linkage cases, Safety-PRM can generate roadmaps with less computational cost than basic PRM and MAPRM. These methods are known for their low computational cost and clearance maximization, respectively.

Safety-PRM also is particularly relevant to experimental robot systems. In this paper, we demonstrate Safety-PRM on a Barrett Whole Arm Manipulator (WAM). In this experiment, we show how expected safety can be mapped directly to robot speed in order to allow the robot to manually test the validity of configurations and then using torque estimation make roadmap modifications.

[1] Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, {nmalone, kazaz, tapia}@cs.unm.edu, jw@unm.edu

II. Preliminaries

We define a *robot* as a movable object whose possible states are encoded by n parameters, its DOFs, each corresponding to an attribute of the object (*e.g.* position, orientation) or object component (*e.g.* joint angle, link displacement, component orientation). The *configuration* of a robot is a point $(x_1, x_2, ..., x_n)$ in a n dimensional space, where x_i is the ith DOF of the robot. This space, called *configuration space* (C_{space}), consists of all possible robot configurations, regardless of the feasibility of the configurations [21]. *Free* C_{space} (C_{free}) is the subset of feasible configurations in C-space, while *blocked* C_{space} ($C_{obstacles}$) is the subset of infeasible configurations. In this context the MP problem is that of finding a series of continuous changes to a robot's DOFs that take it between initial and goal configurations without ever entering $C_{obstacles}$. Although computing explicit $C_{obstacle}$ boundaries is, in the general case, an intractable problem, it is often possible to efficiently determine if a configuration is feasible or not by performing a *collision detection* (CD) test in the robot's environment.

PRMs approach randomized motion planning by building a graph in a subset of C_{free}, the *roadmap* [19]. Roadmap vertices are added during *node generation*, where C_{space} is sampled and collision-free configurations are inserted into the roadmap. During *node connection*, k neighboring nodes selected by a *distance metric* are evaluated by a deterministic *local planner*; if the local planner is successful, an edge connecting the neighboring nodes is added to the roadmap.

III. Related Work

The work in this paper builds on the work of Clearance-informed PRM methods, modifiable roadmaps and planning with uncertainty in order to work in environments that have been modeled with noisy sensors.

A. Modeling Environments With Sensors

Modeling an environment is one of the most challenging tasks in robotics [29]. Some common technologies used for modeling include: GPS, radar, laser, sonar and cameras. However, every technology is subject to error, measurement noise, which is not statistically independent and thus modeling is subject to systematic correlated errors [29].

Two commonly used techniques are RGB-D mapping and Laser range finders. RGB-D mapping uses a RGB camera with distance values for every pixel in the image [13], [15], [10], [28]. Either active stereo [20] or time of flight sensing is used [4]. The measurement noise from models created with RGB-D mapping varies depending on the method used [13]. Laser range finders use lasers to scan and map an environment. For example, [30] proposes a SLAM solution for mapping and a probabilistic method for localization but maps are subject to cumulative error. In other work, 2D laser range finders are used to scan and map the environment [18].

B. Clearance-informed Roadmaps

Clearance-informed methods utilize obstacle information to create high quality paths, more efficient sampling or to handle moving obstacles. Obstacle surfaces are critical in methods such as Obstacle Based PRM (OBPRM) [3] and Medial Axis PRM (MAPRM) [31]. In OBPRM, configurations are placed near the obstacle surfaces in order to traverse narrow passages easier.In MAPRM, configurations are placed on the medial axis of C_{free} to increase clearance and visibility. Here, random samples are generated and retracted towards the medial axis.However, in both methods a complete and accurate model of the environment is needed. Earlier methods related to MAPRM such as Generalized Voronoi Diagram and Hierarchical Generalized Voronoi Graph, were restricted to workspace clearance [11] [7] [8].

PRMs have also been adapted to handle changes in the environment due to moving obstacles. The work in [14] expands PRMs to work under both kinodynamic constraints and with moving obstacles. However, uncertainty is not built into the roadmaps, directly. Sensing errors are handled by growing the obstacles. The work in [26] also utilizes PRMs with moving obstacles. In this method, a first-stage approximate dynamic global roadmap about the connectivity is maintained and a second-stage path is extracted from the dynamic global roadmap to locally plan.

Toggle PRM [9] maps both C_{free} and $C_{obstacle}$. Similar to our method, it does not throw out the in collision information, however Toggle PRM uses C_{free} and $C_{obstacle}$ to aid sample efficiency in narrow passages.

C. Modifiable Roadmap Methods

Modifying a roadmap is a means to construct tunable roadmap paths, handle invalid paths and to accommodate moving obstacles. One type of modifiable roadmap, [27], constructs a coarse roadmap which is refined in the areas of interest relative to a query. The approach generates an approximate roadmap, postponing detailed validation until query time where query preferences are applied to customize the roadmap. [12] takes an initial roadmap and query solution and adds nodes and edges to improve the query solution.

Deformable roadmaps such as [32] replan online paths by using deformation to fix invalid parts of a path. If a portion of a path is found to be in collision, the midpoint of the invalid portion is pushed a specified distance away from the obstacle. A similar approach in [17] looks at the path homotopy class, which relies on the notion of path deformability. This method only looks at homotopy classification and the possibility of deforming a given path to fit another.

The approaches of [25] [4] address real-time obstacle avoidance in dynamic environments. These methods start with an initial path that is collision free and incrementally modify the path to maintain a smooth, collision free path. These methods only rely on workspace clearance by using $protective\ bubbles$ to deform the path.

D. Planning With Uncertainty

The two main types of uncertainty are model uncertainty and motion uncertainty. [5] extends PRMs to work while building a workspace model and is used to guide exploration to areas that have not been sensed, but it does not deal with measurement noise. [16] and [23] are concerned with localization error of the robot. [22] is concerned with model error, but it uses a probability of collision for rejection sampling of nodes in the roadmap.Furthermore the method is tailored to 2D environments where the model noise is quantifiable. In [6], the general PRM and RRT method is followed, however, the cost of connecting two vertices is evaluated through Monte Carlo simulations to deal with uncertainty. The work in [2] instead samples local motions at each state to estimate the state transition probability for

each possible action. A roadmap and the state transition probabilities are then used to formulate a Markov Decision Process (MDP) which is then solved using Infinite Horizon Dynamic Programming.

Motion uncertainty can also be handled by working in belief space. In [1] the authors chose to model a 2D motion planning problem as a Partially Observable Markov Decision Process (POMDP). Belief space and POMDPs are also used to solve the uncertainty problem in [24]. Here the belief space is used to approximate the solution to the POMDP on a 2D motion planning problem.

IV. Methods

In order to handle environment models with inaccuracies, the PRM method must be modified. In previous PRM work, collision checking of the robot to the environment is often done as a binary check (either free or in collision). In order to handle an environment with noise, a probability of collision is stored with each configuration. These probabilities are also used to guide connection and to find feasible paths. Therefore, the Safety-PRM provides flexible methods for tuning between planning goals (expected clearance and path length), works on many robot types (rigid bodies and linkages), and is inexpensive to compute.

A. Node Generation

The first step in PRM methods is generating a set of samples that approximates C_{free}. In an environment modeled with noise, the boundary between C_{free} and $C_{obstacle}$ is fuzzy. Thus, unlike a standard PRM method, we do not discard nodes in collision. Rather, we associate a probability of validity to each node that is dependent on its expected distance from the obstacle surface. This ensures that not all nodes are weighted with an equal measure of quality.

$$P_v(X) = \frac{-1 * atan(D(X) - 1) + \frac{\pi}{2}}{\pi} \quad (1)$$

A probability of collision is stored with each node based on the amount of perceived clearance/penetration. The clearance/penetration probability, P_v(X), of configuration X is calculated in Equation 1. $D(X)$ is the distance of configuration X to the nearest obstacle surface in the noisy model. This is just one example to produce a collision probability due to noisy obstacle boundaries. The $\frac{\pi}{2}$ shifts the equation so that being close to the obstacle, but not necessarily in collision, has a high collision probability. This provides an extra buffer around obstacle surfaces and causes the algorithm to favor higher clearance vertices. Figure 2 shows a plot of the probability function defined in Equation 1.

Since the introduction of the first PRM method [19], there have been many PRM variants introduced [3], [31]. Since we are keeping and associating a probability to all samples, we opted to generate configurations using a uniform random distribution. This method is able to produce many samples quickly at a low computational cost. However, many PRM variants could be used for sample generation.

B. Node Connection

For each node in the map, we attempt to connect it with its k nearest neighbors with a straight-line in C_{space} (other methods could be used to define a local transition between two nodes). However, since the collision status of the configuration is only partially known, the definition of nearest neighbor considers the probability of collision. Thus, we define a function that combines the neighbor's probability of collision and its distance. This causes the best candidate neighbors to likely be free and proximal.

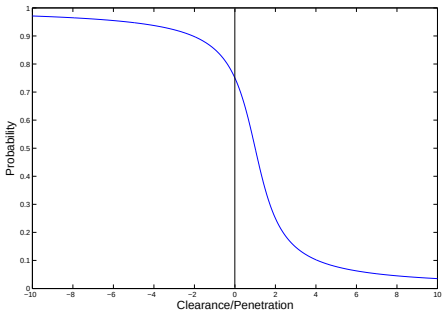


Fig. 2: Probability of a configuration being in collision based on the clearance/penetration of the configuration. Based on Equation 1.

Equation 2 provides a calculation of the distance between two configurations, c_i and c_j. In this equation λ is a weighting term, c_i is the vertex, $P_v(c_j)$ is the probability that c_j is in collision, and $dist(c_i, c_j)$ is the Euclidean distance of c_j from vertex c_i. This metric will provide smaller scores to neighbors which are close and have a low probability of being in collision. The k closest neighbors are then chosen for connection. $P_v(c_j)$ is evaluated by using Equation 1 where the input to equation 1 is the clearance/penetration of configuration c_j.

$$d(c_i, c_j) = (\lambda)P_v(c_j) + (1 - \lambda)dist(c_i, c_j) \quad (2)$$

In the results shown, edges are computed between neighbors c_i and c_j by using a straight-line in C_{space}. The weight for the edge e_{ij} = (c_i,c_j) is a function of the probability of collision, $P_e(e_{ij})$, and the length of an edge. The probability of collision of an edge, $P_e(e_{ij})$, is a function of the intermediate conformations along the edge $c_i = c_0, c_1, c_2, ..., c_{n-1}, c_n = c_j$, where the number of intermediate conformations depends on the resolution, a parameter of the method. In the results shown, $P_e(e_{ij}) = max(P_v(c_0), P_v(c_1), P_v(c_2), ..., P_v(c_{n-1}), P_v(c_n))$. $P_v(c_i)$ is evaluated using Equation 1 where the input to the equation is the clearance/penetration for each c_i along the edge.

$$Weight(e_{ij}) = (\gamma)P_e(e_{ij}) + (1 - \gamma)nlen(e_{ij}) \quad (3)$$

Equation 3 shows the weight calculation for the edge, e_{ij} from c_i to c_j. The γ value allows for customizable scaling of clearance to edge length. This is particularly important because modeling errors can be highly variable. $nlen(e_{ij})$ is a normalized length of an edge. In the results shown, these values are normalized by the maximum edge length in order to provide intuitive scaling between probabilities of collision and edge length.

Queries are then done using the standard Dijkstra's algorithm on the weighted roadmap. Since the roadmap edge weights capture the uncertainty, Dijkstra's algorithm will choose paths with the least uncertainty to the goal.

V. EXPERIMENTS

We explore Safety-PRM with rigid body and articulated linkage robots in two environments. Figure 3 depicts the environments. In each environment the query is designed to show the trade-off in paths with high collision-free probability (clearance) versus path length by varying γ in the edge weighting function.

- **Narrow:** consists of an elongated environment with three boxes dividing the space. The first and second boxes produce a narrow corridor while the second and third box produce a wide corridor. The query is built so that the narrow corridor has a shorter path to the goal but higher probability of being in collision, while the wide corridor has a low probability of being in collision but a longer path length.
- **Plank:** The plank environment has several long planks running the same direction but with minor offsets in their angles. This produces several narrow corridors through which the robot must navigate length wise and transverse across.

Each environment is run with three different rigid body robots, however, results are only shown for one robot in each environment. The results for each robot in each environment are comparable. These robots are designed to have varying difficulty for Safety-PRM, MAPRM and Uniform PRM.

- **Stick:** is a long thin object. Its long side is too long to fit through most of the paths, but when oriented correctly it can pass through most of the passages with ease.
- **Big Arrow:** The Big Arrow is a pyramid and will just narrowly fit through many of the passages.
- **Linkage:** The Linkage is a serial three link robot. Each joint can be moved independently for a total of nine degrees of freedom.

The value of γ in Equation 3 determines the trade-off between short paths and paths with high clearance. Thus, we will show experiments with varying γ. Each value of γ is shown for 10 runs with 10 different random seeds. The parameter used to identify neighbors in all environments is fixed at $\lambda = 0.75$. This value was emperically found to make well connected roadmaps at $k = 5$, k being the number of neighbors each node has. Safety-PRM was implemented within the Parasol Motion Planning Library (PMPL) developed at Texas A&M University. Experiments were run on a single core of an Intel 3.40 GHz CORE i7-2600 CPU and 8 GB of RAM.

In order to demonstrate modeling error, we chose to use a simple error model in our simulations. Error is introduced into every collision detection test. The error is modeled as $\pm$ 5% of the maximum length of the robot to scale the problem with the robot. The error model we use most closely matches a sensor that would produce uniform errors. While this simplified model does not exactly match the error one would see from sensed environments, it approximates the error enough to demonstrate our approach. However, the goal of Safety-PRM is to create a roadmap which can compensate for many types of sensor error models.

VI. RESULTS

A. Rigid Bodies

Path quality and performance are the two metrics we use to evaluate Safety-PRM. Path quality is determined by path length and clearance. Unfortunately, in certain planning problems these two parameters can be at odds with each other. Paths with high clearance can have longer path lengths and short paths can have lower clearance because of the obstacles in the environment. Performance is determined primarily by execution time which is directly related to the number of nodes, edges and collisions detection calls.

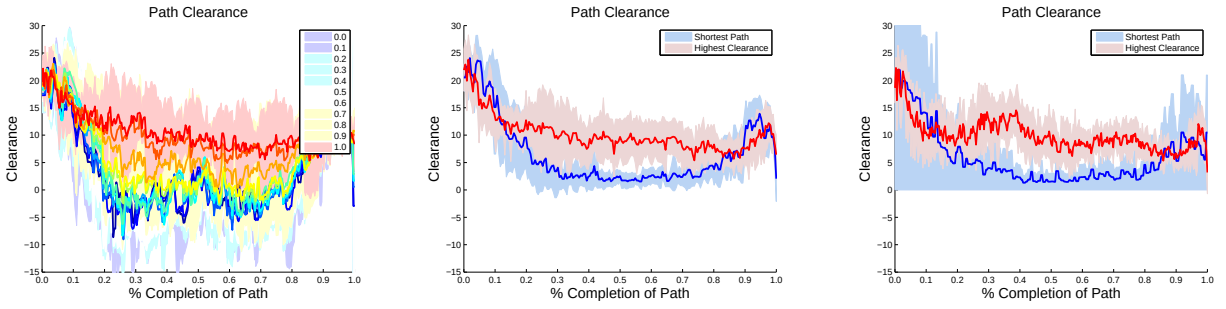
Figure 4a shows the path clearances for $\gamma = [0, 1.0]$ with a step size of 0.1 on a roadmap of 500 nodes. Figure 4f shows the path lengths for the corresponding γ values. The graphs indicate the γ values under 0.7 produce very poor paths. However, γ values above 0.7 produce significantly different paths. Figure 4d shows the Safety-PRM paths for $\gamma = 0.0, 0.6, 0.7, 0.8, 0.9, 1.0$. Figures 4a and 4d directly show the tunability as $\gamma = 0.7$ goes through the shortest path and as the γ value increases beyond 0.7 the paths have slightly higher clearance. Note that for $\gamma > 0.7$ all the paths go through the higher clearance section of the environment.

However, the tunability is not the primary contribution of this work. Primarily, we are concerned with an efficient way to produce high clearance paths in an inaccurate model cheaply. Figure 4e shows tunable MAPRM paths. This graph shows the shortest path produced by MAPRM and the highest clearance path for a roadmap of 500 nodes. These paths are comparable to the paths produced by Safety-PRM in figure 4d. In this particular experiment, Safety-PRM produces slightly higher clearance paths but more importantly Safety-PRM often produces these paths at lower cost. In sampling methods the number of Collision Detection calls is the primary factor in performance as most of the computational time is spent determining the collision state of a configuration. Table I for the Big Arrow robot at 500 nodes shows that Safety-PRM makes 72% fewer collision detection (CD) calls than MAPRM. For the Big Arrow in the narrow environment Safety-PRM makes at worst 45% fewer CD calls and at best 75% fewer CD calls. In this environment Safety-PRM is able to produce a comparably high clearance path to MAPRM but at a cheaper cost. We also show MAPRM run to completion in Table I, however, this is just shown for convenience. Our goal is to produce a cheap reusable roadmap which well approximates the C-Space of the inaccurately modeled environment, so we can find high expected clearance paths. Running MAPRM to completion produces roadmaps which solve the query but are not necessarily general. For example, Table I shows MAPRM producing at most 33 nodes for any robot environment combination.

For the slightly more complex plank environment, Safety-PRM performs slightly worse than on the narrow environment. In the plank environment for 100 nodes in Table I Safety-PRM requires 14% more CD calls than MAPRM. However, for the rest of node values Safety-PRM does better than MAPRM. For more than 100 nodes Safety-PRM does at worst 18% fewer CD calls than MAPRM (500 nodes) and at best 40% fewer CD calls (2000 nodes). For the one instance where Safety-PRM does worse this is due to MAPRM on rigid bodies being able to use workspace clearance and the fact that on 100 nodes MAPRM has 40% fewer edges than Safety-PRM while for the other node values MAPRM has between 24% to 14% fewer edges. Furthermore, since MAPRM is using this workspace clearance it is able to

(a) (b) (c) (d) (e)

Fig. 3: Environments: (a) Narrow and (b) Plank and Robots (c) Stick, (d) Big Arrow, and (e) Linkage.



(a) Safety-PRM Clearance (b) MAPRM Clearance (c) Uniform PRM Clearance

(d) Safety-PRM Clearance Paths (e) MAPRM Clearance Paths (f) Path Lengths

Fig. 4: Narrow Environment with the Big Arrow robot and 500 nodes. The paths have been normalized to the range of $[0, 1]$ so that multiple runs can be compared. 0 is the start of the query while 1 is the goal configuration of the query. The shaded regions indicate the standard deviation over 10 runs for each experiment. For 4d and 4e Each Color indicates a different path based on clearance. Red is the highest clearance path and blue is the shortest path. For MAPRM the shortest and highest clearance paths are shown.

push nodes to the medial axis in a single step and this environment essentially has three hallways thus favoring MAPRM's medial axis solution as the solution path lies along the medial axes of the hallways. Also, this is not using inflated obstacles. If these were sensed obstacles, the standard way of using MAPRM would be to inflate the obstacle boundaries, however, doing so could potentially lose the narrow passageways. It is important to note that for rigid bodies and roadmaps greater than 500 nodes Safety-PRM is always cheaper than Uniform PRM.

B. Linkages

Now we increase the difficulty of the problem by moving to linkage robots. In order for MAPRM to push accurately it must use an approximate ray casting solution [31]. Our method does not need to use such a technique as we are using a probabilistic encoding of the clearance.

In this experiment, we use a 3 link planar robot. Each link is connected via parallel revolute joints, to form a simple planar linkage. Thus, the robot has 9 Degrees of Freedom, 3 position, 3 orientation and 3 joint angles. Orientation and position is relative to the first joint in the linkage. The linkage experiments are run in the narrow and plank environments with the same starting position and goal position as the rigid body experiments. The only exception is that the joints are set to be slightly offset from straight.

Similar to the rigid bodies Safety-PRM produces comparable if not better clearance paths than MAPRM. Figure 5a shows the clearances for Safety-PRM and 5b shows the clearances for MAPRM for a roadmap of 500 nodes. These two graphs show that Safety-PRM produces paths that are comparable to MAPRM. However, Table I shows that Safety-PRM makes between 95% to 98% fewer CD calls than MAPRM. Similarly, to demonstrate the tunability Figure 5g shows the path lengths for varying γ values. As γ increases the path lengths become longer and Figure 5a shows that as γ increases the path clearance become higher.

Figures 5d, 5e and 5f show the paths for Safety-PRM, MAPRM and Uniform PRM. The MAPRM and Uniform graphs show the shortest path and the highest clearance path in the roadmap, while Safety-PRM shows $\gamma = 0.6, 0.7, 0.8, 0.9, 1.0$. These graphs show that Uniform

produces more angular paths than either MAPRM or Safety-PRM. Similarly, it shows that the paths produced by Safety-PRM are comparable to the paths produced by MAPRM, however, at a cheaper computation cost. It is important to note that while we show results for MAPRM to completion we are not not simply trying to solve one query. We are attempting to produce a cheap map with high expected clearance, which approximates the C-space of the environment in order to solve multiple queries in an inaccurate model of the environment.

The rigid body and linkage experiments show that Safety-PRM is comparable to MAPRM and significantly cheaper than MAPRM in the linkage environments. However, unlike MAPRM and Uniform PRM, Safety-PRM is built to take inaccurate models into account. This means that given an inaccurate model, Safety-PRM will produce cheap expected clearance paths better than MAPRM or Uniform PRM. Both Safety-PRM and MAPRM are more expensive than uniform but Safety-PRM provides higher quality paths.

C. Whole Arm Manipulator

As a final test of Safety-PRM, we created a roadmap in simulation for the Barrett Whole Arm Manipulator (WAM) and then used this roadmap to drive the actual WAM. The WAM is a 7 DoF serial link manipulator as seen in Figure 1. It is a cable driven system controlled with position encoders and torque sensors. For the experiments in this paper, the WAM has been connected to a GE Intelligent Platforms reflective memory network in a spoke design that allows multiple computers to share memory at speeds ranging from 43 MB/s to 170 MB/s. The reflective memory network allows remote computers to handle the planning and learning processing, while leaving a small and fast computer on-board the WAM to handle simple motion control.

Using Safety-PRM with the WAM we demonstrate the following capabilities:

- Cheap method of finding high expected clearance paths
- Flexible tunability of expected clearance and path length
- Direct encoding of robot speed for safety near obstacles based on expected clearance

To validate Safety-PRM on real robots we first create a simulation of the WAM environment and run Safety-PRM. The roadmap and path produced in simulation is then transferred to, and executed on the actual WAM hardware. Figure 6g shows the WAM simulation path. There is a single obstacle, a box, that the WAM must reach over. The path chosen by Safety-PRM creates a high expected clearance path despite requiring to navigate near the obstacle to reach the goal. This roadmap and path were then transferred to the actual WAM and used to navigate the WAM in a similar query. The query is only similar because the box is approximately placed in the same location as it is in simulation. To increase the safety of the motion path we use the clearance probability to determine the speed of the robot. The speed of a move is determined in seconds by $2 + (1 + P(Edge))^2$ where $P(Edge)$ is the probability that an edge is in collision, however any function can be used to determine the speed. The general form would be $c+f(P(Edge))$ where c is a constant and f is some function. Figure 6 shows a snapshot sequence of the WAM executing the path created in simulation. Each subfigure shows the starting location for the node and the time it will take to reach the next node in the path. For this path the edge probabilities are [0.31, 0.33, 0.19, 0.62, 0.80]. Thus, Figure 6a has a move time of 3.71 seconds to Figure 6b, which has move time of 3.76 to Figure 6c. Figure 6g shows 6a to 6f labeled, which correspond to the nodes in the simulation path.

Another useful application of the Safety-PRM roadmap and the WAM is to utilize the torque estimation of the WAM to refine the uncertain roadmap. The Safety-PRM method only produces expected clearance paths. Collision are still possible. As such we propose a method to dynamically update the roadmap if a collision occurs on the real hardware. In this experiment, the stiffness of the WAM is reduced to allow the WAM to safely bump into obstacles. Torque estimation is then used to determine when the WAM has collided with an obstacle and to immediately stop the arm from damaging itself. If the arm collides with an obstacle and sends a stop signal the algorithm knows the edge that is being traversed is actually in collision. This information is then used to remove the in collision edge from the roadmap. The WAM is backed up to the last known safe node in the roadmap and the path is replanned in the pruned roadmap. This allows for intelligent refinement of the roadmap given the expected clearance of the Safety-PRM method. Figure 7 shows a sequence of this process. In this experiment, the γ value is set to 0.5, so that the planner will choose a short path which collides with the obstacle. Figure 7c is when the WAM collides with the obstacle and Figure 7d shows the WAM backing up. The remaining figures show the replanned route to the goal.

VII. Conclusions

We have shown that the Safety-PRM roadmap offers several advantages over basic PRM roadmaps for real robots. Using Safety-PRM allows for cheap tunable roadmaps to be produced for complex robots and environments. Safety-PRM is computationally cheaper than MAPRM and it allows for inaccurate environment models without the need to scale up obstacles, but still provides high expected clearance paths. These advantages of Safety-PRM allow it to be used on real robotic hardware as demonstrated by the WAM applications.

VIII. Acknowledgments

This work was supported by Sandia National Laboratories PO# 1074659. Tapia is supported in part by the National Institutes of Health (NIH) Grant P20RR018754 to the Center for Evolutionary and Theoretical Immunology.

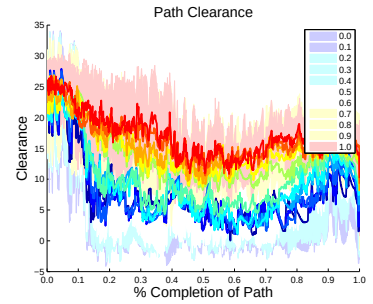
References

[1] Agha-mohammadi, A., Chakravorty, S., and Amato, N. On the probabilistic completeness of the sampling-based feedback motion planners in belief space. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 3983–3990.

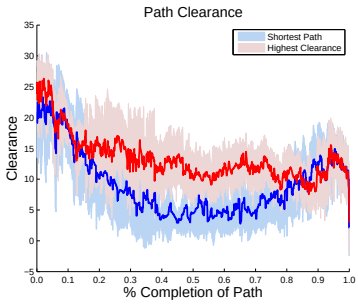
[2] Alterovitz, R., Siméon, T., and Goldberg, K. The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty. In *Robotics: Science and Systems* (2007), Citeseer, pp. 246–253.

[3] Amato, N. M., Bayazit, O. B., Dale, L. K., Jones, C. V., and Vallejo, D. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective* (Natick, MA, 1998), A.K. Peters, pp. 155–168. Proc. Third Workshop on Algorithmic Foundations of Robotics (WAFR), Houston, TX, 1998.

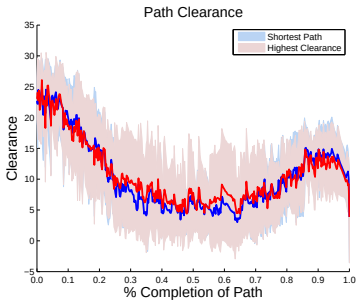
[4] Brock, O., and Khatib, O. Elastic strips: Real-time path modification for mobile manipulation. *International Symposium of Robotics Research 8* (1997), 5–13.



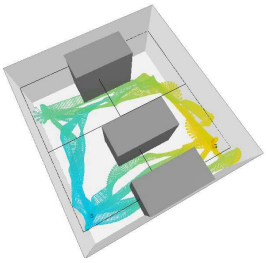
(a) Safety-PRM Clearance



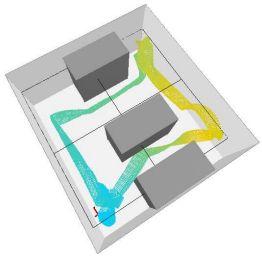
(b) MAPRM Clearance



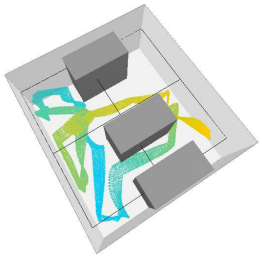
(c) U-PRM Clearance



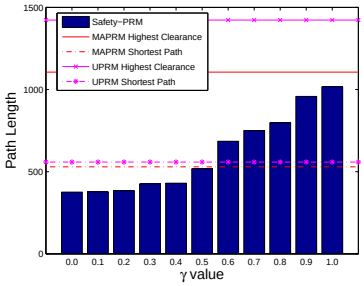
(d) Safety-PRM Paths for varying γ values



(e) MAPRM shortest and highest clearance paths



(f) Uniform PRM shortest and highest clearance path



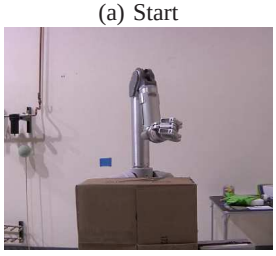
(g) Path Lengths

Fig. 5: Link Robot in Narrow Environment with 500 nodes. 5d 5e 5f Show the Linkage Paths on the Narrow Environment. The colors of each path indicate the start (blue) and the end (yellow) of the paths. The paths have been normalized to the range of $[0, 1]$ so that multiple runs can be compared. 0 is the start of the query while 1 is the goal configuration of the query. The shaded regions indicate the standard deviation over 10 runs for each experiment.

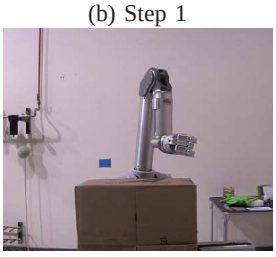
(a) Start

(b) Step 1

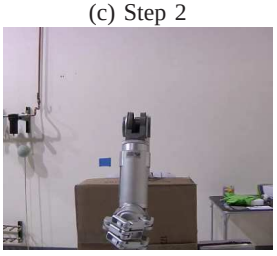
(c) Step 2



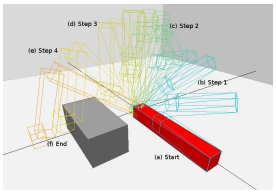
(d) Step 3



(e) Step 4

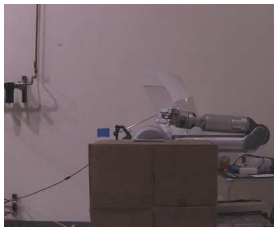


(f) End

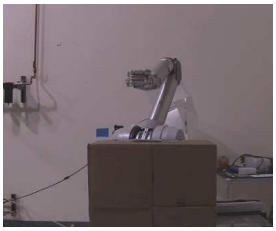


(g) Simulation Path

Fig. 6: Sequence of the WAM Path, The move time is determined by $2 + (1 + P(Edge))^2$ where $P(Edge)$ is the probability of an edge being in collision.

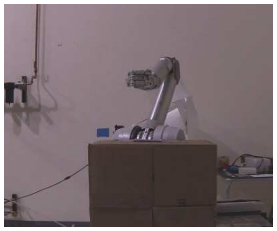


(a) Start



(b) Step 1

(c) Step 2



(d) Step 3

Fig. 7: Sequence of WAM Path using torque sensing, 7c is where the WAM collides with the obstacle and replans the path. The WAM then returns to configuration 7b and follows the same path as in Figure 6d, 6e and 6f.

		Safety-PRM			MAPRM			Uniform Random PRM		
Robot	Env	Nodes	Edges	CD's	Nodes	Edges	CD's	Nodes	Edges	CD's
Big Arrow	Narrow	Comp	N/A	N/A	33	122.0	8,385.7	25	100.4	2,437.4
		100	1,173.4	15,255.5	100	435.6	27,758.7	100	570.6	12,171.5
		500	4,725.2	52,470.2	500	2,250.2	184,380.1	500	3,666.0	65,983.9
		1000	8,703.0	90,103.4	1000	4,764.6	306,870.6	1000	7,854.8	128,984.7
		1500	13,170.0	130,688.2	1500	7,381.0	508,205.1	1500	12,178.4	188,379.4
		2000	17,688.4	170,725.3	2000	10,010.6	652,091.8	2000	16,594.2	245,356.5
Stick	Plank	Comp	N/A	N/A	14	44.0	1,872.1	11	37.6	1,542.3
		100	904.0	20,702.4	100	551.79	18,158.8	100	539.2	20,082.9
		500	4,551.2	81,534.0	500	3,472.4	99,259.1	500	3,367.4	101,648.8
		1000	9,152.6	149,258.1	1000	7,472.0	208,658.0	1000	7,261.4	201,228.4
		1500	13,767.0	213,742.3	1500	11,621.8	329,380.4	1500	11,256.2	300,961.3
		2000	18,420.0	277,172.4	2000	15,858.6	458,286.9	2000	15,409.2	401,185.2
Linkage	Narrow	Comp	N/A	N/A	Comp	61.8	266,273.7	Comp	66.6	7,099.5
		100	1,516.4	101,187.6	100	478.6	1,934,083.5	100	1,649.2	139,243.9
		500	7,709.2	427,025.7	500	2,565.0	10,285,723.8	500	5,455.0	432,761.8
		1000	15,568.8	807,674.6	1000	5,244.0	20,001,180.5	1000	8,634.1	655,001.1
		1500	23,413.2	1,167,438.8	1500	7,959.0	30,300,016.4	1500	11,667.7	854,130.0
		2000	31,313.2	1,517,071.4	2000	10,676.4	39,538,284.2	2000	15,292.2	1,077,846.8
Linkage	Plank	Comp	N/A	N/A	Comp	40.2	303,423.2	Comp	49.0	3,221.4
		100	1,627.3	68,258.7	100	353.8	2,485,202.3	100	382.0	25,191.7
		500	8,332.6	271,513.3	500	2,238.0	10,958,301.2	500	2,331.2	149,392.9
		1000	16,695.6	491,709.8	1000	4,739.8	20,904,573.9	1000	4,993.6	330,395.5
		1500	25,191.2	705,611.7	1500	7,320.2	31,503,340.8	1500	7,781.0	533,506.3
		2000	33,653.8	905,726.3	2000	9,975.6	41,851,387.7	2000	10,635.8	747,872.1

TABLE I: Rigid Body and Linkage Experiments for different environments and robots with k = 5.

[5] BURNS, B., AND BROCK, O. Sampling-based motion planning with sensing uncertainty. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 3313–3318.

[6] CHAKRAVORTY, S., AND KUMAR, S. Generalized sampling-based motion planners. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 41*, 3 (2011), 855–866.

[7] CHOSET, H., AND BURDICK, J. Sensor-based exploration: The hierarchial generalized voronoi graph. *Int. J. Robot. Res. 19*, 2 (2000), 96–125.

[8] CHOSET, H., WALKER, S., EIAMSA-ARD, K., AND BURDICK, J. Sensor-based exploration: Incremental construction of the hierarchical generalized voronoi graph. *Int. J. Robot. Res. 19*, 2 (2000), 126–148.

[9] DENNY, J., AND AMATO, N. Toggle prm: Simultaneous mapping of c-free and c-obstacle - a study in 2d -. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on* (2011), pp. 2632–2639.

[10] ENGELHARDA, N., ENDRESA, F., HESSA, J., STURMB, J., AND BURGARDA, W. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden* (2011), vol. 2011.

[11] FOSKEY, M., GARBER, M., LIN, M. C., AND MANOCHA, D. Sm01-144: A voronoi-based framework for motion planning and maintainability applications. Tech. rep., University of North Carolina, 2001.

[12] GERAERTS, R., AND OVERMARS, M. H. Creating high-quality roadmaps for motion planning in virtual environments. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (2006), pp. 4355–4361.

[13] HENRY, P., KRAININ, M., HERBST, E., REN, X., AND FOX, D. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *the 12th International Symposium on Experimental Robotics (ISER)* (2010).

[14] HSU, D., KINDEL, R., LATOMBE, J. C., AND ROCK, S. Randomized kinodynamic motion planning with moving obstacles. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)* (2000), pp. SA1–SA18.

[15] HUANG, A., BACHRACH, A., HENRY, P., KRAININ, M., MATURANA, D., FOX, D., AND ROY, N. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Int. Symposium on Robotics Research (ISRR),(Flagstaff, Arizona, USA)* (2011).

[16] HUANG, Y., AND GUPTA, K. Rrt-slam for motion planning with motion and map uncertainty for robot exploration. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (2008), IEEE, pp. 1077–1082.

[17] JAILLET, L., AND SIMÉON, T. Path deformation roadmaps: Compact graphs with useful cycles for motion planning. *I. J. Robotic Res. 27*, 11-12 (2008), 1175–1188.

[18] KATZ, R., MELKUMYAN, N., GUIVANT, J., BAILEY, T., NIETO, J., AND NEBOT, E. Integrated sensing framework for 3d mapping in outdoor navigation. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (2006), IEEE, pp. 2264–2269.

[19] KAVRAKI, L. E., ŠVESTKA, P., LATOMBE, J. C., AND OVERMARS, M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat. 12*, 4 (August 1996), 566–580.

[20] KONOLIGE, K. Projected texture stereo. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (2010), IEEE, pp. 148–155.

[21] LOZANO-PÉREZ, T., AND WESLEY, M. A. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM 22*, 10 (October 1979), 560–570.

[22] MISSIURO, P. E., AND ROY, N. Adapting probabilistic roadmaps to handle uncertain maps. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (2006), IEEE, pp. 1261–1267.

[23] PEPY, R., AND LAMBERT, A. Safe path planning in an uncertain-configuration space using rrt. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* (2006), IEEE, pp. 5376–5381.

[24] PRENTICE, S., AND ROY, N. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research 28*, 11-12 (2009), 1448–1465.

[25] QUINLAN, S., AND KHATIB, O. Elastic bands: Connecting path planning and control. In *In Proc. of the International Conference on Robotics and Automation* (1993), pp. 802–807.

[26] RODRIGUEZ, S., LIEN, J.-M., AND AMATO, N. M. A framework for planning motion in environments with moving obstacles. In *Proc. IEEE Int. Conf. Intel. Rob. Syst. (IROS)* (2007).

[27] SONG, G., MILLER, S. L., AND AMATO, N. M. Customizing PRM roadmaps at query time. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (2001), pp. 1500–1505.

[28] STURM, J., MAGNENAT, S., ENGELHARD, N., POMERLEAU, F., COLAS, F., BURGARD, W., CREMERS, D., AND SIEGWART, R. Towards a benchmark for rgb-d slam evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf.(RSS), Los Angeles, USA* (2011), vol. 2, p. 3.

[29] THRUN, S. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium 1* (2003), 1–35.

[30] THRUN, S., BURGARD, W., AND FOX, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (2000), vol. 1, IEEE, pp. 321–328.

[31] WILMARTH, S. A., AMATO, N. M., AND STILLER, P. F. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)* (1999), vol. 2, pp. 1024–1031.

[32] YOSHIDA, E., AND KANEHIRO, F. Reactive robot motion using path replanning and deformation. In *ICRA* (2011), pp. 5456–5462.