HIERARCHICAL PROBABILISTIC ESTIMATION OF ROBOT REACHABLE WORKSPACE

Jing Yang, Patrick Dymond and Michael Jenkin
Department of Computer Science and Engineering, York University, 4700 Keele Street, Toronto, Canada

Keywords: Reachable workspace, Probabilistic motion planning, Mobile robotics.

Abstract: Estimating a robot's reachable workspace is a fundamental problem in robotics. For simple kinematic chains within an empty environment this computation can be relatively straightforward. For mobile kinematic structures and cluttered environments, the problem becomes more challenging. An efficient probabilistic method for workspace estimation is developed by applying a hierarchical strategy and developing extensions to a probabilistic motion planner. Rather than treating each of the degrees of freedom (DOFs) 'equally', a hierarchical representation is used to maximize the volume of the robot's workspace that is identified as reachable for each probe of the environment. Experiments with a simulated mobile manipulator demonstrate that the hierarchical approach is an effective alternative to the use of an estimation process based on the use of a traditional probabilistic planner.

1 INTRODUCTION

The reachable workspace $\mathcal{W}_{reach}$ is defined as the volume or space within which all the points can be reached by a reference point of the mechanism, for example, the centre of the end-effector (Kumar, 1980). Reachable workspace estimation is a fundamental problem in robotics as workspace properties can represent important criteria in the evaluation and design of mechanical manipulators (Lenarcic and Umek, 1994; Badescu and Mavroidis, 2004), robots (Zacharias et al., 2007) and environmental layouts (Yang et al., 2008). The determination of $\mathcal{W}_{reach}$ involves a considerable amount of numerical calculations, which increases with the number of degrees of freedom of the mechanism and the complexity of the environment.

For kinematic chains within an empty environment the computation of $\mathcal{W}_{reach}$ can be relatively straightforward. For mobile kinematic structures in the presence of obstacles, the problem becomes more challenging. Notably, estimating the $\mathcal{W}_{reach}$ for a mobile robot can be expressed in terms of the ability of the device to plan motions within its environment. Motion planning emerged as a crucial and productive research area in robotics in the late 1960's (Latombe, 1991) and its applications in real world problems continue to attract researchers from all over the world.

In basic motion planning (Latombe, 1991), given a robot $\mathcal{A}$ and a static workspace $\mathcal{W}$ containing a set of obstacles, the objective is to determine a collision-free motion between the specified start and goal for $\mathcal{A}$. A configuration c of $\mathcal{A}$ is a specification of the position and orientation of $\mathcal{A}$ in $\mathcal{W}$. A configuration c is said to be free if $\mathcal{A}$ positioned at c does not collide with any obstacles in $\mathcal{W}$. The free configuration space $\mathcal{C}_{free}$ is defined as the set of all free configurations of $\mathcal{A}$. The motion planning problem is therefore formulated as computing a path in $\mathcal{C}_{free}$ between two given configurations.

A complete solution to the motion planning problem is known to be exponential to the robot's degree of freedom (DOF) (Canny, 1988). As a consequence a number of heuristic approaches to path planning have been developed. The main difference between the probabilistic approaches and earlier complete approaches is that the probabilistic approaches do not attempt to construct an exact representation of $\mathcal{C}_{free}$. Rather they create a simplified graph that approximately "covers" $\mathcal{C}_{free}$ and captures its connectivity in reasonable time. The Probabilistic Roadmap Method (PRM) (Horsch et al., 1994; Kavraki et al., 1996) is a popular heuristic motion planner. The algorithm first constructs a roadmap by connecting randomly sampled collision free configurations and then answers multiple queries by attempting to connect them to the roadmap.

In estimating a robot's workspace it is important to

Yang J., Dymond P. and Jenkin M. (2009).
HIERARCHICAL PROBABILISTIC ESTIMATION OF ROBOT REACHABLE WORKSPACE.
In *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 60-66
DOI: 10.5220/0002205600600066

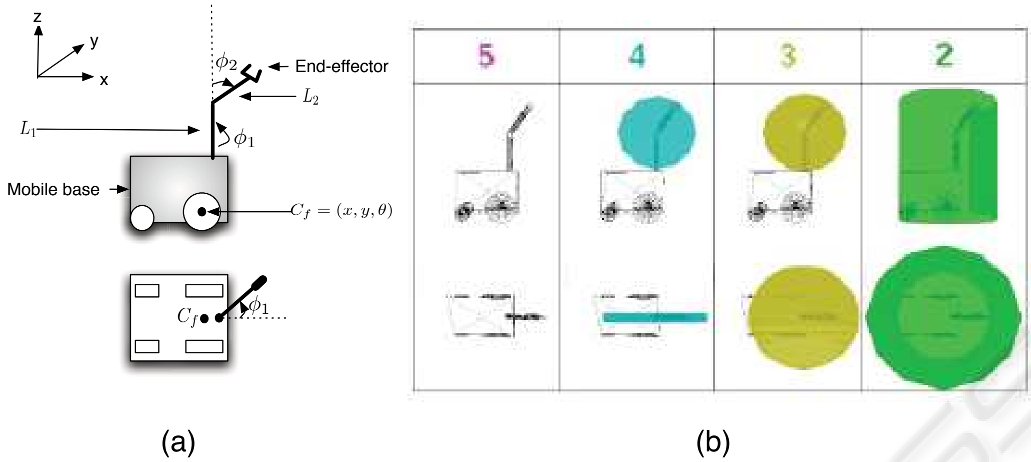


Figure 1: (a) $\mathcal{A}$ consists of $\mathcal{A}_{base}$ and $\mathcal{A}_{arm}$ that has two links L_1 and L_2 connected by revolute joints. The configuration of $\mathcal{A}$ is written as $(x, y, \theta, \phi_1, \phi_2)$. (b) Representations of the hierarchical occupancy of $\mathcal{A}$ are shown in different colors. The body itself also has occupancy constraints but these are not shown.

observe that some DOFs are likely to be more important than others in terms of their effect on how much of the workspace is reachable. In addition, it may be possible to construct subspaces of the workspace defined by arbitrary configurations of certain combinations of joints. Motivated by these observations we explore the use of a hierarchical structure of the DOFs of the kinematic device to establish the entire $\mathcal{W}_{reach}$. In this hierarchy we order the DOFs of the robot in terms of their predefined "importance". Then we consider corresponding sub-versions of the kinematic structure in which sub-versional joints are considered over their range of motion. Each of these sub-versions defines a reachability subspace that can be established as being reachable with a single probe into the robot's workspace. This hierarchical search mechanism can be used to enhance the probabilistic algorithms for reachable workspace estimation.

This paper is structured as follows. Section 2 reviews existing algorithms for reachable workspace estimation. Section 3 outlines our hierarchical approach. Section 4 includes comparison results from applying the hierarchical approach and basic PRM to a simulated mobile manipulator. Finally Section 5 summarizes the work and provides possible directions for future research.

2 RELATED WORK

Although a range of techniques exist for reachable workspace estimation (Badescu and Mavroidis, 2004; Hsu and Kohli, 1987; Zacharias et al., 2007) most existing approaches consider the problem for robotic manipulators and do not consider arbitrary obstacles in the environment. Robotic manipulators are fixed at one end and this assumption provides certain efficiencies for reachable workspace estimation. For example, one straightforward method to compute $\mathcal{W}_{reach}$ is to take planer sections of the workspace defined by the joint angles that make up the kinematic structure and determine the contour of the section in the plane. Rotating and translating this plane based on other joints in the chain yields the three-dimensional workspace (Morecki and Knapczyk, 1999).

A numerical approach calculates the exact $\mathcal{W}_{reach}$ by tracing boundary surfaces of a workspace (Kumar, 1980). In this approach, an imaginary force is applied to the reference point at the end-effector in order to achieve the maximum extension in the direction of the applied force. The manipulator reaches its maximum extension when the force's line of action intersects all joint axes of rotational joints and is perpendicular to all joint axes of prismatic joints. The drawback of this algorithm is its exponential time complexity and that it only deals with manipulators that have ideal joints (without limits). A more efficient system was later developed for computing $\mathcal{W}_{reach}$ for manipulators with joint limits (Alameldin et al., 1990). The system decomposed the problem into two subproblems: workspace point generation by direct kinematic based techniques and surface computation by extracting the workspace contours utilizing a subset of the points generated in the first module.

Much work has been done on capturing workspace properties for interesting kinematic structures such as human arms (Lenarcic and Umek, 1994; Zacharias et al., 2007) and reconfigurable robotic arms (Bade-

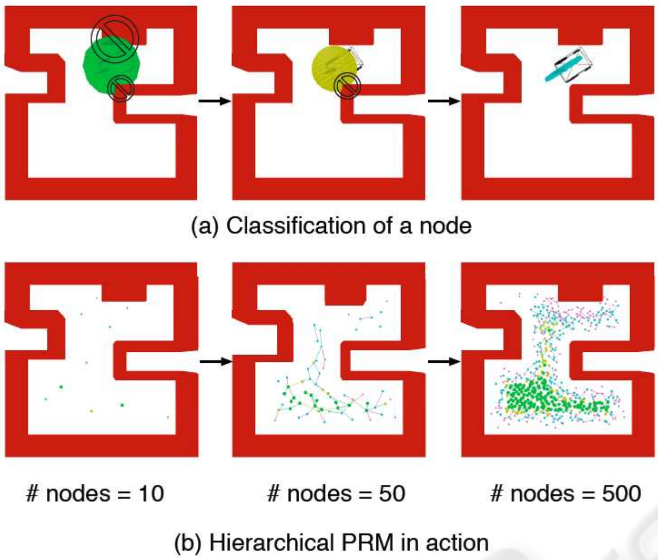


Figure 2: Construction of $\mathcal{R}$. Upper row shows the generation and classification of one node. The rank of a node c^r is calculated by checking the hierarchical occupancy representation of $\mathcal{A}$. The lower row shows the hierarchical PRM in operation. The coverage and connectivity of $\mathcal{R}$ increases as more nodes are added.

scu and Mavroidis, 2004). Lenaric and Umek developed a simplified kinematic model to represent human arm motion. $\mathcal{W}_{reach}$ was determined by calculating the reference point on the wrist for all combinations of values of joint coordinates inside the given ranges. Another approach presented by Zacharias et al. discretizes the workspace into equally sized small cubes. Into each cube a sphere is inscribed and sample points on the sphere are examined using inverse kinematics. The percentage of the points on the sphere that are reachable is used to represent its level of reachability.

The structure of $\mathcal{W}_{reach}$ for a given device can be very complex. Existing methods aim at capturing the exact shape and volume of $\mathcal{W}_{reach}$ and they involve a considerable amount of numerical, time-consuming calculations. It can be expensive to compute exact $\mathcal{W}_{reach}$ for a mobile robot in a cluttered environment. In this paper we develop a probabilistic algorithm to give a proper estimation for $\mathcal{W}_{reach}$ for an arbitrary mobile device operating in a known and cluttered environment.

3 HIERARCHICAL REACHABLE WORKSPACE ESTIMATION

For a robot $\mathcal{A}$ moving in the workspace $\mathcal{W}$, the robot's degrees of freedom (DOFs) are the minimum set of independent displacements/orientations that specify $\mathcal{A}$'s complete position and orientation in $\mathcal{W}$. Thus a configuration c of $\mathcal{A}$ with n DOFs can be specified as a set of n parameters, say $j_1, ..., j_n$, and theoretically there can be $O(n^2)$ different orderings of the joints. First we define an ordering of the DOFs such that more important DOFs have a lower index. Although there can be many definitions according to the nature of the problem, here we define the importance of a DOF by its effect on the volume of $\mathcal{A}$'s occupation in $\mathcal{W}$. This importance weight is expressed more formally below.

For a point a in $\mathcal{A}$, let $P_a : \mathcal{C} \rightarrow \mathcal{W}$ be the mapping that calculates the position of point a in $\mathcal{W}$ when $\mathcal{A}$ is placed at configuration c. Depending on the position of a in $\mathcal{A}$, P_a is a function of $c' \subseteq c$. We say that a is determined by a DOF j_i if $j_i \in c'$. Now we can define a weight function of a DOF j_i in terms of the volume of $\mathcal{A}$ in $\mathcal{W}$ determined by it:

Defn 1: $w(j_i) = \mid \{a \in A \mid a \text{ is determined by } j_i\} \mid$.

A DOF j_x is more important than j_y if $w(j_x) > w(j_y)$. Therefore we can write the configuration c as a vector of length n in a decreasing order of their importance, say $c = (j_1, j_2, ... j_n)$, i.e. for $i = 1, 2, ... n-1, w(j_i) \geq w(j_{i+1})$.

Take the 5-DOF mobile manipulator shown in Figure 1 as an example. x and y determines the entire robot, and θ determines all the portions except the rotation center of the mobile base (assume the mobile robot can rotate around its center). ϕ_1 determines links L_1 and L_2. ϕ_2 determines link L_2. Therefore, $w(x) = w(y) > w(\theta) > w(\phi_1) > w(\phi_2)$, and the ordered configuration can be written as $(x, y, \theta, \phi_1, \phi_2)$ (or $(y, x, \theta, \phi_1, \phi_2)$).

Given an ordering of the DOFs, we seek a hierarchical representation within which certain joint angles are ‘free’ and can assume arbitrary values within some previously defined domain. Let the domain of j_i be $\mathcal{D}_i$, $c^r = (j_1, j_2, ... j_r)$ is a subset of $\mathcal{D}_1 \times \mathcal{D}_2 \times ... \mathcal{D}_n$, given by $\{\forall x_{r+1} \in \mathcal{D}_{r+1}, x_{r+2} \in \mathcal{D}_{r+2}, ... x_n \in \mathcal{D}_n \mid (j_1, j_2, .. j_r, x_{r+1}, x_{r+2}, ... x_n)\}$. That is c^r is the set of possible configurations with joints $1...r$ having specific values but joints $r+1...n$ being free. This hierarchical concept applies to general kinematic structures in the domain of motion planning.

Algorithm 1: Node selection.

$nodeFound \leftarrow false$
while $\neg nodeFound$ **do**
  $c \leftarrow$ a randomly chosen configuration in $\mathcal{C}$
  for $k \leftarrow 1$ to n **do**
    if $V(c^k)$ **then**
      $nodeFound \leftarrow true$
      break
    end if
  end for
end while
$N \leftarrow N \cup \{c^k\}$

In order to integrate the above hierarchy into the workspace estimation process, we first perform two types of analysis to the hierarchical representations: occupation analysis and reachability analysis. Define the occupied area $OA_c(c^r)$ of c^r as the union of the occupied volume in $\mathcal{W}$ of every element in c^r, and the reachable area $RA_c(c^r)$ of c^r as the union of the reachable points of every element in c^r. Under the hierarchy nodes with lower r occupy and reach larger workspace than those with higher r. To be precise, we have these two lemmas:

Lemma 1. $\forall i, j \in [0, n], i < j \implies OA_c(c^i) \supseteq OA_c(c^j)$. For some configuration of $\mathcal{A}$, the occupied workspace of the lower hierarchy is the superset of that of the higher hierarchy.

Lemma 2. $\forall i, j \in [0, n], i < j \implies RA_c(c^i) \supseteq RA_c(c^j)$. For some configuration of $\mathcal{A}$, the reachable workspace of the lower hierarchy is the superset of that of the higher hierarchy.

In addition, let $V(c)$ be the function that returns true iff $\mathcal{A}$ is collision free when it is at configuration c. Similarly, $V(c^r)$ returns true iff every element in c^r is valid, i.e. $OA_c(c^r)$ does not collide with any obstacle in the environment. Therefore, we have the following lemma:

Lemma 3. $\forall i, j \in [0, n], i < j \wedge V(c^i) \implies V(c^j)$. For some configuration of $\mathcal{A}$, that its lower hierarchical representation is free implies the higher hierarchical representation is free, too.

Hierarchical representations of both the reachability and occupancy can be very complex shapes depending on the kinematics of the robot. In practice the computation of the exact hierarchical representations is unnecessary. Conservative representations of these complex shapes can provide significant computational savings and this computation can be done prior to the execution of the motion planner. This needs to be done only once for each DOF of the robot, independent of the robot’s configuration.

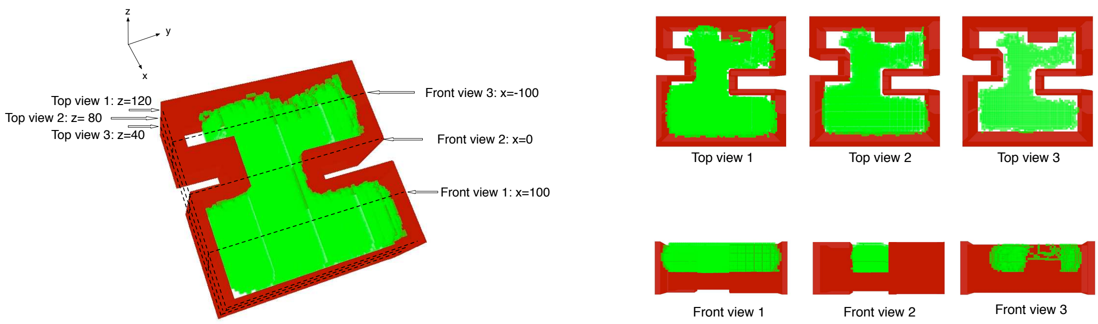
Algorithm 2: Connect(a^{r_1}, b^{r_2}).

$\tau \leftarrow$ the edge candidate returned by the local path locator
Discretize τ into a list of configurations $\tau' = (c_1, c_2, ..., c_m)$
$r_{current} \leftarrow MAX(r_1, r_2)$
for all $c_i \in \tau'$ **do**
  for $k \leftarrow r_{current}$ to n **do**
    if $V(c_i^k)$ **then**
      $r_{current} \leftarrow k$
      break
    else
      exit and report failure
    end if
  end for
end for
$E \leftarrow E \cup \{(a, b)^{r_{current}}\}$

To establish the general representation of $OA(c^r)$ and $RA(c^r)$, we fix the first r DOFs and take all possible values of the remaining DOFs to construct the hierarchical body. Note that $RA(c^r)$ defines a subspace in the robot’s workspace that is reachable by some unique configuration within c^r, and that for any two configurations within c^r there exists a continuous path between them.

As an example, consider the mobile manipulator $\mathcal{A}$ shown in Fig. 1(a), let $\mathcal{D}_1 = [x_{min}, x_{max}]$, $\mathcal{D}_2 = [y_{min}, y_{max}]$, $\mathcal{D}_3 = [-\pi, \pi)$, $\mathcal{D}_4 = [-\pi, \pi)$, and $\mathcal{D}_5 = [-\pi, \pi)$. Fig 1(b) shows the hierarchical representation of $\mathcal{A}$ from level 5 down to 2, indicated by color. For simplicity we used ideal ranges $[-\pi, \pi)$ for the rotational joints of $\mathcal{A}$.

The hierarchy can be integrated into workspace estimation using probabilistic planners such as PRM. We describe below the main steps involved in the construction of a hierarchical roadmap $\mathcal{R}$ for efficient reachable workspace estimation. Nodes with large reachable areas are preferred (they establish more of the environment as being reachable for each calculation). So for each configuration c, we look for the minimum value r_{min} such that $V(c^{r_{min}})$ is true, and we call r_{min} the rank of c. The procedure described in



(a) 3D workspace view (b) Slices through the workspace

Figure 3: The estimation of the 3D reachable workspace $\mathcal{W}_{reach}$ (green area) shown in different vertical and horizontal layers of top and front views.

the pseudocode below finds a random configuration and establishes its most general representation in the hierarchy.

In the "for" loop from Line 4 to Line 9, the algorithm computes the rank of the node by checking collisions of the hierarchical representations of the robot's occupancy. Once the minimal valid hierarchical node is established the configuration together with the computed rank is added to the set of nodes N (Line 11).

Whenever a new hierarchical node is found, we select a number of candidate nodes from the current set N and try to connect the new node to each of them. In addition to the connection computation performed by the traditional local planner, the rank of the edge should be established. For an edge e we look for the minimum hierarchy r_{min} such that $V(e^{r_{min}})$ is true along the edge.

The hierarchical node interconnection is built upon a local path locator and a hierarchy establisher. The local path locator returns an edge candidate, i.e. a local path that $\mathcal{A}$ can follow from one configuration to another. Then the hierarchy establisher checks if the edge candidate is collision free and meanwhile establishes the edge's most general representation in the hierarchy. The process of establishing the hierarchical node interconnection is outlined in Algorithm 2.

In line 3, the hierarchy r is initialized to be the maximum value of the ends of the edge. There is an obvious lemma according to the definition of the hierarchical edge connecting two nodes a^{r_1} and b^{r_2}:

Lemma 4: $V(e^r) = true \implies r \geq r_1 \wedge r \geq r_2$, i.e. the rank of an edge is not less than the rank of either end node of the edge. Algorithm 2 searches over the sequence of configurations on the edge for verification and hierarchy establishment. This general approach is straightforward to implement. Note that the hierarchy is established through the validation of the configurations. Similarly, to apply this strategy to other probabilistic motion planners it can be done when samples are checked for collision.

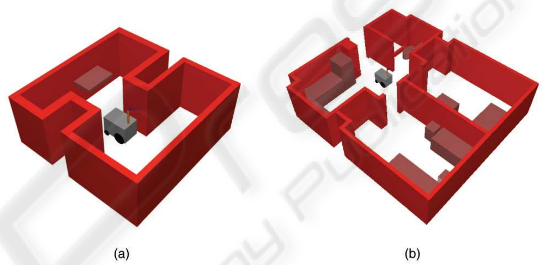


Figure 4: Experimental setup. (a) The mobile manipulator is placed in a simple 3D environment; (b) The mobile manipulator is placed in an apartment-like environment.

After the hierarchical roadmap $\mathcal{R}$ is built, $\mathcal{W}_{reach}$ can be computed from the connected component of $\mathcal{R}$ that contains the initial configuration of $\mathcal{A}$ through mapping function RA.

4 EXPERIMENTAL VALIDATION

We conducted experiments of our algorithm on the simulated 5-DOF mobile manipulator $\mathcal{A}$ shown in Fig. 1. First we provide an example that illustrates the hierarchical strategy described in the previous section. Fig. 2 provides details of the execution of the hierarchical PRM on this example. Initially $\mathcal{R}$ contains only one node that represents the initial configuration of $\mathcal{A}$. The rank of each randomly generated node is determined by looking for the most general occupancy representation of $\mathcal{A}$ that does not collide with any obstacles. Similarly the rank of each edge is determined by looking for the most general occupancy representation of $\mathcal{A}$ along the edge that does not collide with any obstacles. The top row of Fig. 2 shows tests for a randomly generated node. c^2 and c^3 generate collisions while c^4 does not, so this specific

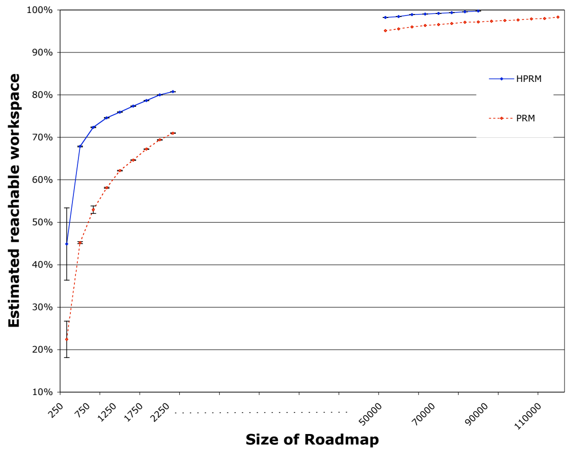


Figure 5: Experimental result for the environment shown in 4(a). The graph shows a comparison of workspace volume computed from hierarchical PRM and repetitive PRM sampling. Each data point plots the average of ten experiments. Standard deviations are plotted.

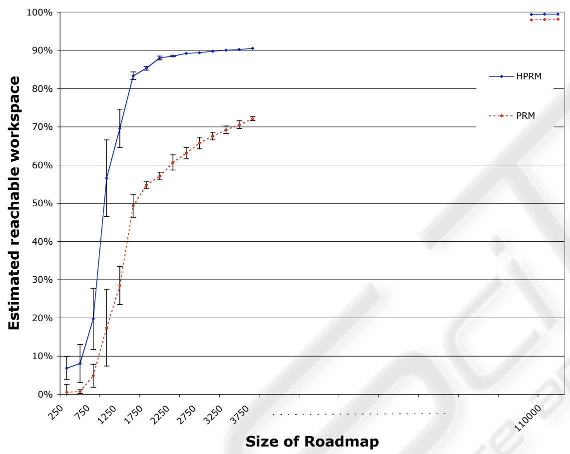


Figure 6: Experimental result for the environment shown in 4(b). The graph shows a comparison of workspace volume computed from hierarchical PRM and repetitive PRM sampling. Each data point plots the average of ten experiments. Standard deviations are plotted.

node is classified as c^4. The construction of $\mathcal{R}$ is incremental. The lower row shows incremental changes in $\mathcal{R}$. As more nodes are added both coverage and connectivity of $\mathcal{R}$ increases.

The connected component of the constructed $\mathcal{R}$ can be mapped to $\mathcal{W}$ such that $\mathcal{W}_{reach}$ is obtained. We use uniform cell decomposition to represent $\mathcal{W}$. Fig. 3 shows the estimated $\mathcal{W}_{reach}$ in different layers for the environment given in Fig. 4(a).

Fig. 5 and Fig. 6 show comparisons of the effectiveness of the hierarchical PRM and random sampling using repetitive PRM for reachable workspace

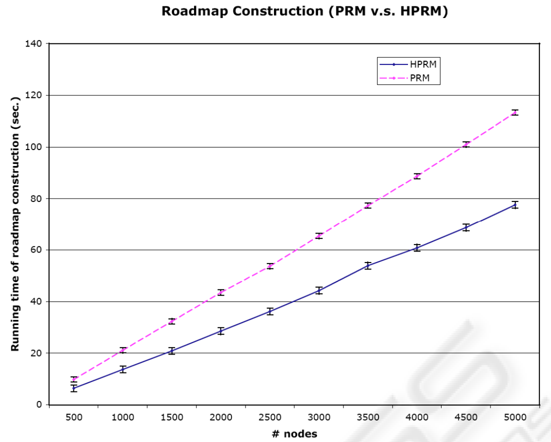


Figure 7: Comparison of the average running time of the construction of traditional roadmaps and the hierarchical ones. Averages are for 20 trials. Standard deviations are shown.

estimation on the environments shown in Fig. 4 (a) and (b) respectively. The x-axis represents the number of samples used in the estimation. The y-axis represents the percentage of the estimated $\mathcal{W}_{reach}$ (the true $\mathcal{W}_{reach}$ of the robot was computed by brute force search). Because randomness is involved, the workspace volume for ten independent runs for each case were averaged. Standard deviations are also plotted. Both graphs show that the hierarchical approach is more effective than repetitive PRM sampling for reachable workspace estimation.

Finally we evaluate the time efficiency of our hierarchical approach. Figure 7 shows the comparison of the running time of the roadmap construction in the basic PRM and the hierarchical approach for the model given in 4(b). Because randomness is involved, running times for 20 independent runs for each case were averaged. The hierarchical PRM performed reasonably well in these experiments. As can be seen from the results, creating a hierarchical roadmap takes less time than creating a traditional roadmap. This is because the hierarchical PRM saves time in collision checking in easy regions.

5 SUMMARY AND FUTURE WORK

A hierarchical approach was presented for adapting probabilistic motion planners for reachable workspace estimation. Unlike traditional probabilistic motion planners that treat each DOF equally, we order the DOF's of the kinematic structure and consider a hierarchical approach to the planning task. Considering the characteristic of the reachable workspace estimation problem, this hierarchy exploration improves the planning process through two critical computations: occupational analysis and reachability analysis. Validation of configurations begins by doing fast tests on simple occupational representations and only progresses to more accurate (and more expensive) evaluations as necessary. Because randomness is involved it is hardly possible to estimate the entire reachable workspace from the probabilistic roadmap within reasonable time. However, by iteratively computing the maximal reachable workspace from each node and edge our hierarchical motion planner can be more effective in the computation process than the traditional ones.

The hierarchical workspace estimation algorithm is especially useful for mobile robots in environments with obstacles. Experiments were conducted on a simulated 5-DOF mobile manipulator in two 3D environments. Experiments show that the hierarchical approach can be an effective and efficient alternative to the repetitive PRM for reachable workspace estimation.

Our current hierarchical algorithm uses the coarse-to-fine hierarchical nature in the process of estimating the workspace. The hierarchical characteristic might also be employed in other aspects of motion planners. For example, one heuristic would be to let the established hierarchy lead the sampling process toward the boundaries of obstacles, i.e. to sample more densely near nodes with higher hierarchy labels than those with lower hierarchy labels.

We can also imagine a more sophisticated definition of reachable workspace which might involve establishing the number of configurations from which the kinematic structure can reach a given location. This might provide insights into different levels of reachability. A space for where there exists many reachable configurations should probably be considered more reachable than one with just a few.

ACKNOWLEDGEMENTS

The financial support of NSERC Canada is gratefully acknowledged.

REFERENCES

Alameldin, T., Badler, N. I., and Sobh, T. (1990). An adaptive and efficient system for computing the 3-d reachable workspace. In *IEEE International Conference on Systems Engineering*, pages 503–506.

Badescu, M. and Mavroidis, C. (2004). New performance indices and workspace analysis of reconfigurable hyper-redundant robotic arms. *The International Journal of Robotics Research*, 23:643–659.

Canny, J. F. (1988). *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA.

Horsch, T., Schwarz, F., and Tolle, H. (1994). Motion planning for many degrees of freedom – random reflections at c-space obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '94)*, pages 3318–3323.

Hsu, M.-S. and Kohli, D. (1987). Boundary surfaces and accessibility regions for regional structures of manipulators. *Mechanism and Machine Theory*, 22:277–289.

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

Kumar, A. (1980). *Characterization of Manipulator Geometry*. PhD thesis, University of Houston.

Latombe, J.-C. (1991). *Robot Motion Planning*. Cluwer.

Lenarcic, J. and Umek, A. (1994). Simple model of human arm reachable workspace. *IEEE Transactions on Systems, Man and Cybernetics*, 24(8):1239–1246.

Morecki, A. and Knapczyk, J. (1999). *Basics of Robotics: Theory and Components of Manipulators and Robots*. SpringerWienNewYork.

Yang, J., Dymond, P., and Jenkin, M. (2008). Accessibility assessment via workspace estimation. *International Journal of Smart Home*, 3:73–90.

Zacharias, F., Borst, C., and Hirzinger, G. (2007). Capturing robot workspace structure: representing robot capabilities. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236.