# Reusable Sampling-Based Techniques for Manipulation via Pushing

Christopher Vo        Jyh-Ming Lien

***Abstract*—In this work, we consider the problem of manipulating a polygonal object through an obstacle-filled environment using only push interactions, or nudges. For this problem, we propose a manipulation planner that could handle multiple queries and offer solutions with a higher probability of success than traditional planners, especially in the face of uncertainty. We have implemented our approach in simulation, and offer some preliminary results to demonstrate the technique.**

## I. Introduction

One of the goals in robotics is to make it possible for robots to interact with objects in the real world. Such tasks usually require the solution to a manipulation planning problem. In our work, we seek ways to perform this kind of manipulation on objects without relying on grasping, using special purpose manipulators, or requiring the design of rigorous analytical models for each type of object. Instead, we seek to perform this manipulation just using push interactions, and through sampling-based methods rather than of analytical models.

Currently, the most popular planners for solving problems with kinodynamic constraints are tree-based planners such as RRT [1], EST [2], PDST [3], and DSLX [4]. The general strategy behind these planners is to grow an exploring tree rooted at an initial state until one of the goal states is (approximately) reached. However, if the initial or an intermediate state is shifted even by a small amount (e.g., due to uncertainty in the pose of the object relative to the robot, and inaccuracies in the simulation and model), then the plan usually becomes invalid, and a new tree must be built from scratch. As a result, many tree-based planners use a limited planning horizon in practice.

Some planners such as in [5] deal with uncertainty through analysis of theoretical models of uncertainty and control. One interesting way to model planning problems is to use Partially Observable Markov Decision Processes (POMDPs). POMDP models are notoriously difficult to solve, but there are several approximation and point-based algorithms such as HSVI2 [6] and SARSOP [7] that have been shown to handle relatively large problems in a reasonable time [8]. Another general approach to handle uncertainty is to use a hybrid (or "hierarchical") motion planning approach, such as in [9], [10]. One interesting method that uses this technique is *temporal logic motion planning* [11]. Along similar lines is the use of *uncertainty roadmaps* as in [12], which explicitly capture the probability of successfully transitioning from one valid configuration (or set of configurations) to another.

All authors are with Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030 USA, {cvo1, jmlien}@gmu.edu

In our work, we propose a new planner that can support multiple queries and can provide solutions with high probability of reaching a goal state despite uncertainties. To accomplish these goals, we propose to use a *graph-based planner*. This generates a reusable roadmap that can be used to improve the efficiency of replanning. The graph generated by the planner also encodes the configuration of the target object in a fuzzy manner. That is, by considering regions of the configuration space instead of single points, we reduce the sensitivity of the resulting plans to uncertainty in position. Each edge of the roadmap also encodes the probability of sucessfully transitioning from one metanode to the next, similar to the concept of uncertainty roadmaps. In our recent work [13], a simplified version of this proposed strategy has already been applied to solve *group motion control problems* involving two groups of interacting agents with complex dynamics and behaviors. Here, we propose a new method that may work well in the scenario of pushing polygonal models, but should be readily modified to handle other problems.

## II. Description of Our Method

Our instance of the polygon pushing problem poses the question of how to plan the motion for a cylindrical robot $R$ to move a polygonal target object $P$ from a given initial pose to a given goal pose using only push interactions. The workspace may be filled with known obstacles that the robot and the object are permitted to touch, but not penetrate. We assume that $R$ is a holonomic cylinder, and that the $P$ that is an extruded polygonal shape with known geometry. We also assume that a black-box kinodynamic simulator is provided, containing the obstacles, the robot, and the object. The simulation provides the resulting trajectories of $P$ and $R$ given control inputs for $R$.

Our method consists of several phases. First, we want to create a local planning function $LP$ that given a starting configuation for $P$ ($p_{start}$), and a desired destination configuration for $P$ ($p_{dest}$), outputs a push manipulation that supports moving $P$ from $P_{start}$ to $P_{dest}$. To do so, we sample information about how $P$ will move when it is pushed by $R$ from various angles. We store this information as points in a KD-tree so that when we are given the inputs $p_{start}$ and $p_{dest}$, we can quickly look up the closest sampled push manipulation that supports pushing $P$ to $p_{dest}$. Furthermore, for each manipulation, we also sample many times with random perturbations, to simulate uncertainty that may arise from imperfectly executing pushes. This information will be used later in roadmap generation to compute probabilities.

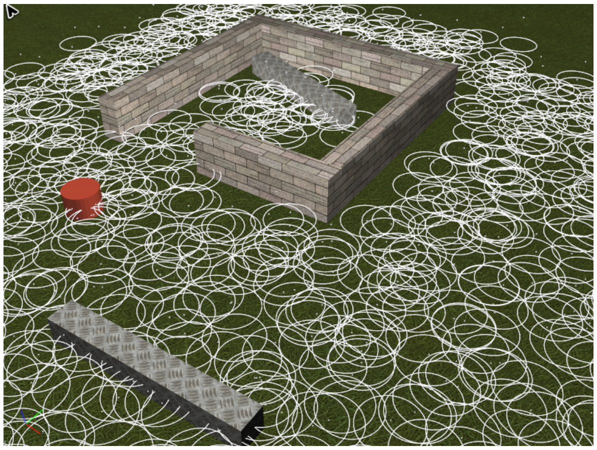

Fig. 1. An example scene. The robot is depicted as a red cylinder, and the target object is the gray bar. White circles depict the meta-nodes sampled from the configuration space after overlap reduction is performed.

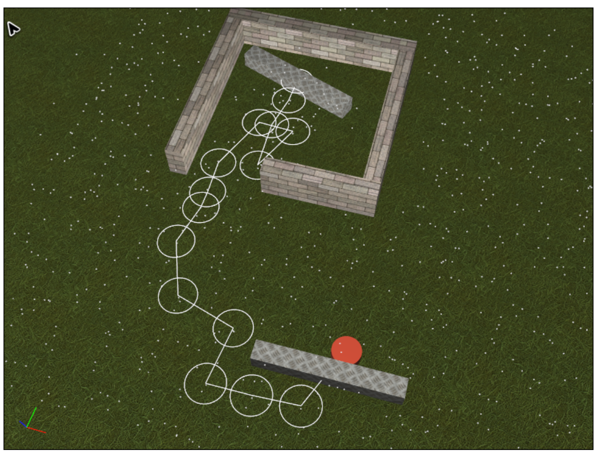

Fig. 2. A safe corridor is computed for the target object to move inside the room. The corridor is depicted by a series of connected circles (metanodes).

Next, we build the roadmap using this data. In general, we first seek to generate *metanodes* which capture sets of states rather than single states, and then we connect them with edges whose weights reflect the probability of successful transit from one metanode to the next. We begin by sampling from the free space. Many sampling techniques can be used; we use the sampling technique from Gaussian PRM [14], mixed with uniform sampling. We convert each configuration $s$ from the set of samples into a metanode by building a sphere, centered at $s$, with pre-defined radius. Configurations that lie within the sphere defined for a configuration $s$ are thus considered to be *conforming* to the metanode for $s$. An example environment with sampled metanodes is shown in Figure 1. This sampling process may lead to many overlapping metanodes, so we also attempt reduce the number of metanodes in the roadmap by removing metanodes that have significant overlap.

Finally, to complete the roadmap, we connect meta-nodes that are in close proximity. The edge between two nodes, $s_a$ and $s_b$ is given a weight that is computed based on sampling the number of feasible, successful pushes between uniformly random configrations conforming to $s_a$, and *any* configuration conforming to $s_b$. The number of successful pushes versus the number of attempted pushes at an edge provides us with a metric that can be used to estimate the probability of successfully pushing an object from some state conforming to node $s_a$ to some state conforming to $s_b$, in the presence of the obstacles. Other metrics can be used, such as the average probability of success.

Given the weighted roadmap, we compute an all-pairs shortest path. This information is stored so that upon any query, a sequence of meta-nodes can be quickly obtained that has the greatest probability of successful manipulation. We call this sequence of meta-nodes a *corridor*, representing many possible paths for pushing $P$. Figure 2 shows an example corridor extracted from a query in the example scene. This corridor represents a *robust path* and can be used online quickly and efficiently to generate a final trajectory.

## References


[1] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," in *Proceeedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999, pp. 473–479.

[2] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," pp. 2719–2726, 1997.

[3] A. M. Ladd and L. E. Kavraki, *Fast Tree-Based Exploration of State Space for Robots with Dynamics*. Springer, STAR 17, 2005, pp. 297–312.

[4] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Discrete search leading continuous exploration for kinodynamic motion planning." in *Robotics: Science and Systems*, W. Burgard, O. Brock, and C. Stachniss, Eds. The MIT Press, 2007.

[5] J. V. D. Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," in *Proceedings of Robotics: Science and Systems (RSS)*, Zaragoza, Spain, June 2010.

[6] T. Smith and R. G. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," in *Proc. Int. Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2005.

[7] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proceedings of Robotics: Science and Systems (RSS)*, 2008.

[8] H. Kurniawati, Y. Du, D. Hsu, and W. S. Lee, "Motion planning under uncertainty for robotic tasks with long time horizons," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 308–323, 2011. [Online]. Available: http://ijr.sagepub.com/content/30/3/308.abstract

[9] L. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *Proceeedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Mobile Manipulation*, 2010.

[10] J. Choi and E. Amir, "Combining planning and motion planning," in *Proceeedings of the IEEE International Conference on Robotics and Automation (ICRA)*, may 2009, pp. 238 –244.

[11] G. Fainekos, H. Kress-Gazit, and G. Pappas, "Temporal logic motion planning for mobile robots," in *Proceeedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 2020 – 2025.

[12] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *Proceeedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3313–3318.

[13] C. Vo, J. Harrison, and J.-M. Lien, "Behavior-based motion planning for group control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, N. Papanikolopoulos, S. Sugano, S. Chiavernini, and M. Meng, Eds. IEEE, 2009.

[14] Y.-T. Lin, "The gaussian prm sampling for dynamic configuration spaces," in *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, dec. 2006, pp. 1 –5.